# PP-Module for Email Clients

Version: 1.0
2021-06-18
**National Information Assurance Partnership**

## Revision History

| Version | Date | Comment |
|---|---|---|
| 1.0 | 2021-06-18 | Initial release as PP-Module |

## Contents


1 Introduction
1.1 Overview
1.2 Terms
1.2.1 Common Criteria Terms
1.2.2 Technical Terms
1.3 Compliant Targets of Evaluation
1.3.1 TOE Boundary
1.4 Use Cases
2 Conformance Claims
3 Security Problem Description
3.1 Threats
3.2 Assumptions
3.3 Organizational Security Policies
4 Security Objectives
4.1 Security Objectives for the TOE
4.2 Security Objectives for the Operational Environment
4.3 Security Objectives Rationale
5 Security Requirements
5.1 PP Security Functional Requirements Direction
5.1.1 Modified SFRs
5.1.1.1 Cryptographic Support (FCS)
5.1.1.2 Identification and Authentication (FIA)
5.1.1.3 Trusted Path/Channels (FTP)
5.2 TOE Security Functional Requirements
5.2.1 Cryptographic Support (FCS)
5.2.2 User Data Protection (FDP)
5.2.3 Identification and Authentication (FIA)
5.2.4 Security Management (FMT)
5.2.5 Protection of the TSF (FPT)
5.2.6 Trusted Path/Channels (FTP)
5.3 TOE Security Functional Requirements Rationale
6 Consistency Rationale
6.1 Protection Profile for
6.1.1 Consistency of TOE Type
6.1.2 Consistency of Security Problem Definition
6.1.3 Consistency of Objectives
6.1.4 Consistency of Requirements
Appendix A - Optional SFRs
A.1 Strictly Optional Requirements
A.1.1 Cryptographic Support (FCS)
A.1.2 User Data Protection (FDP)
A.2 Objective Requirements
A.3 Implementation-based Requirements
Appendix B - Selection-based Requirements
B.1 Cryptographic Support (FCS)
B.2 Identification and Authentication (FIA)
B.3 Protection of the TSF (FPT)
Appendix C - Extended Component Definitions
C.1 Extended Components Table
C.2 Extended Component Definitions
Appendix D - Implicitly Satisfied Requirements
Appendix E - Entropy Documentation and Assessment
Appendix F - Acronyms
Appendix G - Bibliography

# 1 Introduction

## 1.1 Overview

The scope of the Email Client PP-Module is to describe the security functionality of email client applications in terms of [CC] and to define functional and assurance requirements for the specific email-related capabilities of email client applications. Email clients are user applications that provide functionality to send, receive, access, and manage email. This PP-Module is intended for use with the following Base-PP:

- Protection Profile for Application Software, Version 1.3

This Base-PP is valid because email clients are a specific type of software application.

## 1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

### 1.2.1 Common Criteria Terms

| | |
|---|---|
| Assurance | Grounds for confidence that a TOE meets the SFRs [CC]. |
| Base Protection Profile (Base-PP) | Protection Profile used as a basis to build a PP-Configuration. |
| Common Criteria (CC) | Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408). |
| Common Criteria Testing Laboratory | Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility, accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations. |
| Common Evaluation Methodology (CEM) | Common Evaluation Methodology for Information Technology Security Evaluation. |
| Distributed TOE | A TOE composed of multiple components operating as a logical whole. |
| Operational Environment (OE) | Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy. |
| Protection Profile (PP) | An implementation-independent set of security requirements for a category of products. |
| Protection Profile Configuration (PP-Configuration) | A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module. |
| Protection Profile Module (PP-Module) | An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles. |
| Security Assurance Requirement (SAR) | A requirement to assure the security of the TOE. |
| 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. |
| 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 an ST. |
| Target of Evaluation (TOE) | The product under evaluation. |

### 1.2.2 Technical Terms

| | |
|---|---|
| ActiveSync | Microsoft protocol for synchronizing messaging and calendar data between mobile clients and email servers. |
| Add-on | Capability or functionality added to an application including plug-ins, extensions or other controls. |
| Email Client | Application used to send, receive, access and manage email provided by an email server. The terms email client and TOE are interchangeable in this document. |
| Internet Message Access Protocol (IMAP) | Protocol for an email client to retrieve email from an email server over TCP/IP; IMAP4 defined in RFC 3501. |

| | |
|---|---|
| Messaging Application Programming Interface (MAPI) | Open specification used by email clients such as Microsoft Outlook and Thunderbird; defined in [MS-OXCMAPIHTTP]. |
| Post Office Protocol (POP) | Protocol for an email client to retrieve email from an email server over TCP/IP; POP3 defined in RFC 1939. |
| Remote Procedure Call (RPC) | Protocol used by Microsoft Exchange to send/receive MAPI commands; defined in [MS-OXCRPC]. |
| Secure/Multipurpose Internet Mail Extensions (S/MIME) | Used to sign or encrypt messages at the request of the user upon sending email and to verify digital signature on a signed message upon receipt. |
| Simple Mail Transfer Protocol (SMTP) | Protocol for an email client to send email to an email server over TCP/IP; SMTP defined in RFC 5321. |

## 1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this PP-Module is an email client application running on a desktop or mobile operating system.

The complexity of email content and email clients has grown over time. Modern email clients can render HTML as well as plaintext, and may include functionality to display common attachment formats, such as Adobe PDF and Microsoft Word documents. Some email clients allow their functionality to be modified by users through the addition of add-ons. Protocols have also been defined for communicating between email clients and servers. Some clients support multiple protocols for doing the same task, allowing them to be configured according to email server specifications.

The complexity and rich feature set of modern email clients make them a target for attackers, introducing security concerns. This document is intended to facilitate the improvement of email client security by requiring use of operating system security services, cryptographic standards, and environmental mitigations. Additionally, the requirements in this document define acceptable behavior for email clients regardless of the security features provided by the operating system.

This Module along with the Protection Profile for Application Software [App PP] provide a baseline set of Security Functional Requirements (SFRs) for email clients running on any operating system regardless of the composition of the underlying platform.

### 1.3.1 TOE Boundary

The physical boundary of the email client is a software application running on a general-purpose operating system. The TOE boundary may include third-party add-ons, but these are non-interfering with respect to security; add-ons provide features that are outside the TOE's logical boundary but must be implemented in such a manner that their inclusion does not compromise the security of the TSF. The figure below shows the TOE's interaction with remote external interfaces that are used to transfer mail between clients. Two separate email clients are shown to show how the TOE can function as both a sender and a receiver using different protocols.

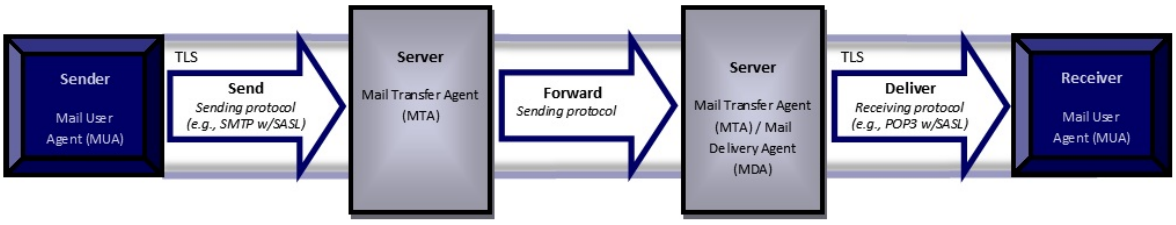



**Figure 1: Sending and Delivering Email over TLS**

## 1.4 Use Cases

Email clients perform tasks associated primarily with the following use case.

**[USE CASE 1] Sending, receiving, accessing, managing and displaying email**

Email clients are used for sending, receiving, viewing, accessing, managing email in coordination with a mail server. Email clients can render HTML as well as plaintext, and can display common attachment formats.

# 2 Conformance Claims

**Conformance Statement**

This PP-Module inherits exact conformance as required from the specified Base-PP and as defined in the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

No additional PPs or PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module aside from the Base-PP.

**CC Conformance Claims**

This PP-Module is conformant to Parts 2 (extended) and 3 (extended) of Common Criteria Version 3.1, Release 5 [CC].

**Package Claims**

This PP-Module is TLS Pacakge conformant.

# 3 Security Problem Description

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

This PP-Module does not repeat the threats, assumptions, and organizational security policies identified in the App PP, though they all apply given the conformance and hence dependence of this PP-Module on it. Together the threats, assumptions and organizational security policies of the App PP and those defined in this PP-Module describe those addressed by an email client as the Target of Evaluation.

Notably, email clients are particularly at risk from the Network Attack threat identified in the App PP. Attackers can send malicious email messages directly to users, and the email client will render or otherwise process this untrusted content.

## 3.1 Threats

The following threat is specific to email clients, and represents an addition to those identified in the Base-PP.

**T.FLAWED_ADDON**
: Email client functionality can be extended with integration of third-party utilities and tools. This expanded set of capabilities is made possible via the use of add-ons. The tight integration between the basic email client code and the new capabilities that add-ons provide increases the risk that malefactors could inject serious flaws into the email client application, either maliciously by an attacker, or accidentally by a developer. These flaws enable undesirable behaviors including, but not limited to, allowing unauthorized access to sensitive information in the email client, unauthorized access to the device's file system, or even privilege escalation that enables unauthorized access to other applications or the operating system.

**T.NETWORK_ATTACK**
: See App PP, Section 3.1.

**T.NETWORK_EAVESDROP**
: See App PP, Section 3.1.

**T.PHYSICAL_ACCESS**
: See App PP, Section 3.1.

## 3.2 Assumptions

This document does not define any additional assumptions.

## 3.3 Organizational Security Policies

An organization deploying the TOE is expected to satisfy the organizational security policy listed below in addition to all organizational security policies defined by the claimed base PP.

This document does not define any additional OSPs.

# 4 Security Objectives

This PP-Module adds SFRs to objectives identified in the Base-PP and describes an additional objective specific to this PP-Module.

## 4.1 Security Objectives for the TOE

**O.MANAGEMENT**
This objective is defined in the Base-PP. This PP-Module maps additional SFRs to it to address the management functionality that is specific to email client applications.

**O.PROTECTED_STORAGE**
This objective is defined in the Base-PP. This PP-Module maps additional SFRs to it to address the data at rest protection functionality that is specific to email client applications.

**O.PROTECTED_COMMS**
This objective is defined in the Base-PP. This PP-Module maps additional SFRs to it to address the data in transit protection functionality that is specific to email client applications.

**O.ADDON_INTEGRITY**
To address issues associated with malicious or flawed plug-ins or extensions, conformant email clients implement mechanisms to ensure their integrity. This includes verification at installation time and update.

## 4.2 Security Objectives for the Operational Environment

This PP-Module does not define any objectives for the Operational Environment.

No environmental security objectives have been identified that are specific to email clients. However, any environmental security objectives defined in the Base-PP will also apply to the portion of the TOE that implements email client functionality.

## 4.3 Security Objectives Rationale

This section describes how the assumptions, threats, and organization security policies map to the security objectives.

**Table 1: Security Objectives Rationale**

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.FLAWED_ADDON | O.MANAGEMENT | The ability to manage the TOE allows for only authorized users to install add-ons, to enable/disable the ability to install add-ons, or to not have any support for add-ons at all. |
| T.NETWORK_ATTACK | O.PROTECTED_COMMS | The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. |
| | O.MANAGEMENT | The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the application to defend against network attack. |
| T.NETWORK_EAVESDROP | O.PROTECTED_COMMS | The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. |
| | O.MANAGEMENT | The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the application to protect the confidentiality of its transmitted data. |
| T.PHYSICAL_ACCESS | O.PROTECTED_STORAGE | The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE. |

# 5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): is used to add details to a requirement (including replacing an assignment with a more restrictive selection) or to remove part of the requirement that is made irrelevant through the completion of another operation, 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: is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

## 5.1 PP Security Functional Requirements Direction

In a PP-Configuration that includes PP, the TOE is expected to rely on some of the security functions implemented by the as a whole and evaluated against the PP. The following sections describe any modifications that the ST author must make to the SFRs defined in the PP in addition to what is mandated by Section 5.2 TOE Security Functional Requirements.

### 5.1.1 Modified SFRs

The SFRs listed in this section are defined in the PP and relevant to the secure operation of the TOE.

#### 5.1.1.1 Cryptographic Support (FCS)

##### FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [**selection**:

- *invoke platform-provided functionality for asymmetric key generation,*
- *implement asymmetric key generation*

].

**Application Note:** This SFR is modified from its Base-PP definition to remove the selection for the TOE not requiring asymmetric key generation.

**Evaluation Activities**

*FCS_CKM_EXT.1*
*There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed, aside from the fact that the materials for the selections that have been refined out of this SFR are not applicable.*

##### FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [**selection**:

- *invoke platform-provided DRBG functionality,*
- *implement DRBG functionality*

] for its cryptographic operations.

**Application Note:** This SFR is modified from its Base-PP definition to remove the selection for the TOE using no DRBG functionality.

**Evaluation Activities**

*FCS_RBG_EXT.1*
*There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed, aside from the fact that the materials for the selections that have been refined out of this SFR are not applicable.*

#### 5.1.1.2 Identification and Authentication (FIA)

##### FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

This SFR is selection-based in the App PP. When the TOE conforms to this PP-Module, it is mandatory because of the modifications that this PP-Module makes to FTP_DIT_EXT.1.

**Evaluation Activities**

*FIA_X509_EXT.1*
*There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.*

##### FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

This SFR is selection-based in the App PP. When the TOE conforms to this PP-Module, it is mandatory because of the modifications that this PP-Module makes to FTP_DIT_EXT.1.

**Evaluation Activities**

*FIA_X509_EXT.2*
*There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed.*

#### 5.1.1.3 Trusted Path/Channels (FTP)

##### FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [**selection**:

- *encrypt all transmitted [sensitive data] with **TLS as defined in the TLS Package and [selection**: HTTPS in accordance with FCS_HTTPS_EXT.1, DTLS as defined in the TLS Package, SSH as conforming to the Extended Package for Secure Shell, IPsec as defined in the PP-Module for VPN Client, **no other protocols**] ,*
- *invoke platform-provided functionality to encrypt all transmitted sensitive data with **TLS and [selection**: HTTPS, DTLS, SSH, **no other protocols**]*

] between itself and another trusted IT product.

**Application Note:** This SFR is modified from its definition in the Base-PP to require that the TOE supports TLS and that its use of TLS is only limited to sensitive data. A conformant TOE must support the use of TLS for email encryption but is permitted to send and receive non-sensitive email messages over an untrusted channel.

**Evaluation Activities**

*FTP_DIT_EXT.1*
*There is no change to the Base-PP EAs for this SFR when this PP-Module is claimed, aside from the fact that the materials for the selections that have been refined out of this SFR are not applicable.*

## 5.2 TOE Security Functional Requirements

The following section describes the SFRs that must be satisfied by any TOE that claims conformance to this PP-Module. These SFRs must be claimed regardless of which PP-Configuration is used to define the TOE.

### 5.2.1 Cryptographic Support (FCS)

##### FCS_CKM_EXT.3 Protection of Key and Key Material

FCS_CKM_EXT.3.1

The TSF shall [**selection**:

- *not store keys in non-volatile memory,*
- *only store keys in non-volatile memory when wrapped as specified in FCS_COP_EXT.2 unless the key meets any one of following criteria: [**selection**:*
  - *The plaintext key is not part of the key chain as specified in FCS_KYC_EXT.1,*
  - *The plaintext key will no longer provide access to the encrypted data after initial provisioning,*
  - *The plaintext key is a key split that is combined as specified in FCS_SMC_EXT.1, and the other half of the key split is either [**selection**: wrapped as specified in FCS_COP_EXT.2, derived and not stored in non-volatile memory],*
  - *The plaintext key is stored on an external storage device for use as an authorization factor,*
  - *The plaintext key is used to wrap a key as specified in FCS_COP_EXT.2 that is already wrapped as specified in FCS_COP_EXT.2,*
  - *The plaintext key is the public portion of the key pair*

  *]*

].

**Application Note:** The plaintext key storage in non-volatile memory is allowed for several reasons. If the keys exist within protected memory that is not user accessible on the email client or operational environment, the only methods that allow it to play a security relevant role is if it is a key split or providing additional layers of wrapping or encryption on keys that have already been protected.

**Evaluation Activities**

*FCS_CKM_EXT.3*
***TSS***
*The evaluator shall verify the TSS for a high-level description of method used to protect keys stored in nonvolatile memory.*

*The evaluator shall verify the TSS to ensure it describes the storage location of all keys and the protection of all keys stored in nonvolatile memory. The description of the key chain shall be reviewed to ensure FCS_COP_EXT.2 is followed for the storage of wrapped or encrypted keys in nonvolatile memory and plaintext keys in nonvolatile memory meet one of the criteria for storage.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

##### FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1

The TSF shall [**selection**:

- *invoke platform-provided key destruction,*
- *implement key destruction using [**selection**:*
  - *For volatile memory, the erasure shall be executed by a [**selection**:*
    - *single direct overwrite [**selection**:*
      - *consisting of a pseudo-random pattern using the email client's RBG,*
      - *consisting of a pseudo-random pattern using the host platform's RBG,*
      - *consisting of zeroes*

      *],*
    - *destruction of reference to the key directly followed by a request for garbage collection*

    *],*
  - *For non-volatile storage, the erasure shall be executed by [**selection**:*
    - *single,*
    - *three or more times*

    *] overwrite of key data storage location consisting of [**selection**:*
    - *a pseudo-random pattern using the email client's RBG (as specified in FCS_RBG_EXT.2 of [App PP],*
    - *a pseudo-random pattern using the host platform's RBG,*
    - *a static pattern*

    *]*

*]*

] that meets the following:[**selection**:

- *NIST SP 800-88,*
- *no standard*

] for destroying all keying material and cryptographic security parameters when no longer needed.

**Application Note:** For the purposes of this requirement, keying material refers to authentication data, passwords, symmetric keys, data used to derive keys, etc. The destruction indicated above applies to each intermediate storage area for key/cryptographic critical security parameters (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another memory location.

**Evaluation Activities**

*FCS_CKM_EXT.4*
***TSS***
*If the platform provides the key destruction, then the evaluator examines the TSS to verify that it describes how the key destruction functionality is invoked.*

*If "destruction of reference..." (for volatile memory) is selected, then the evaluator shall examine the relevant interface definition to ensure that the interface supports the selection and description in the TSS.*

*If the application invokes key destruction, the evaluator checks to ensure the TSS describes each of the secret keys (keys used for symmetric encryption or data authentication), private keys, and CSPs used to generate keys; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeroes, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator checks to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on a drive are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*If the TSF performs its own key destruction, the evaluator shall perform the following test:*

- ***Test 1:** For each type of authorization service, encryption mode, and encryption operation, a known authorization factor, and chain of keys must be provided to the evaluator with an associated ciphertext data set (e.g. if a passphrase is used to create an intermediate key, then the ciphertext containing the encrypted key as well as the intermediate key itself must be provided to the evaluator.) The evaluator shall use the email client in conjunction with a debugging or forensics utility to attempt to authorize themselves, resulting in the generation of a key or decryption of a key. The evaluator shall ascertain from the TSS what the vendor defines as "no longer needed" and execute the sequence of actions via the email client to invoke this state. At this point, the evaluator should take a dump of volatile memory and search the retrieved dump for the provided authorization credentials or keys (e.g. if the password was "PaSSw0rd", perform a string search of the forensics dump for "PaSSw0rd"). The evaluator must document each command, program or action taken during this process, and must confirm that no plaintext keying material resides in volatile memory. The evaluator must perform this test three times to ensure repeatability. If during the course of this testing the evaluator finds that keying material remains in volatile memory, they should be able to identify the cause (i.e. execution of the grep command for "PaSSw0rd" caused a false positive) and document the reason for failure to comply with this requirement. The evaluator shall repeat this same test, but looking for keying material in nonvolatile memory.*

### FCS_KYC_EXT.1 Key Chaining

FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of: [**selection**:

- *one,*
- *a key stored in platform key storage,*
- *intermediate keys originating from: [**selection**:*
  - *a password as specified in FCS_CKM_EXT.5,*
  - *one or more other authorization factor(s),*
  - *credentials stored in platform key storage*

*]*

] to the data encryption/decryption key(s) using the following method(s): [**selection**:

- *use of the platform key storage*,
- *use of platform key storage that performs key wrap with a TSF provided key*,
- *implement key wrapping as specified in FCS_COP_EXT.2*,
- *implement key combining as specified in FCS_SMC_EXT.1*

] while maintaining an effective strength of [**selection**:

- *128 bits*,
- *256 bits*

]

**Application Note:** Key Chaining is the method of using multiple layers of encryption keys to ultimately secure the data encryption key. The number of intermediate keys will vary. This applies to all keys that contribute to the ultimate wrapping or derivation of the data encryption key; including those in protected areas. This requirement also describes how keys are stored.

**Evaluation Activities**

*FCS_KYC_EXT.1*
***TSS***
*The evaluator shall verify that the TSS* describes a high-level description of the key hierarchy for all authorizations methods that are used to protect the encryption keys. The evaluator shall examine the TSS to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap that meets FCS_COP_EXT.2.*

*The evaluator shall then verify that the TSS* describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. A high-level description should include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or knowledge of the key within the chain and the effective strength of the data encryption key is maintained throughout the key chain.*

**If necessary, this information may be presented in a proprietary document rather than the TSS.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

### FCS_SMIME_EXT.1 Secure/Multipurpose Internet Mail Extensions (S/MIME)

FCS_SMIME_EXT.1.1

The TSF shall implement both a sending and receiving S/MIME v4.0 Agent as defined in RFC 8551, using CMS as defined in RFCs 5652, 5754, and 3565.

**Application Note:** The RFCs allow for an agent to be either sending or receiving, or to include both capabilities. The intent of this requirement is to ensure that the email client is capable of both sending and receiving S/MIME v4.0 messages.

FCS_SMIME_EXT.1.2

The TSF shall transmit the ContentEncryptionAlgorithmIdentifier for AES-128 CBC, AES-256 CBC, and [**selection**: *AES-128 GCM*, *AES-256 GCM*, *no other*] as part of the S/MIME protocol.

**Application Note:** AES was added to CMS as defined in RFC 3565.

FCS_SMIME_EXT.1.3

The TSF shall present the digestAlgorithm field with the following Message Digest Algorithm identifiers [**selection**: *id-sha256*, *id-sha384*, *id-sha512*] and no others as part of the S/MIME protocol.

FCS_SMIME_EXT.1.4

The TSF shall present the signatureAlgorithm field with the following: sha256withRSAEncryption and [**selection**:

- *sha384WithRSAEncryption*,
- *sha512WithRSAEncryption*,
- *ecdsawithsha256*,
- *ecdsawithsha384*,
- *ecdsawithsha512*

] and no other algorithms as part of the S/MIME protocol.

**Application Note:** RFC 8551 mandates that receiving and sending agents support RSA with SHA256. The algorithms to be tested in the evaluated configuration are limited to the algorithms specified in the FCS_SMIME_EXT.1.4 selection. Any other algorithms implemented that do not comply with these requirements should not be included in an evaluated email client.

FCS_SMIME_EXT.1.5

The TSF shall support use of different private keys (and associated certificates) for signature and for encryption as part of the S/MIME protocol.

FCS_SMIME_EXT.1.6

The TSF shall only accept a signature from a certificate with the digitalSignature bit set as part of the S/MIME protocol.

**Application Note:** It is acceptable to assume that the digitalSignature bit is set in cases where there is no keyUsage extension.

FCS_SMIME_EXT.1.7

The TSF shall implement mechanisms to retrieve certificates and certificate revocation information [**selection**: *for each signed/encrypted message sent/received*, *[**assignment**: frequency]*] as part of the S/MIME protocol.

**Application Note:** In accordance with FIA_X509_EXT.1.1 in [App PP], certificate revocation may use a Certificate Revocation List (CRL) or Online Certificate Status Protocol (OCSP). The email client can define how this mechanism behaves, including whether it uses the underlying OS, but it is required that a mechanism exists such that revocation status is supported and so that certificates can be retrieved for sending/receiving messages. Frequency is configurable in FMT_MOF_EXT.1.1. In this requirement, frequency can be interpreted as a one-time function with local storage, as a regularly scheduled retrieval, or as a mechanism that requires manual intervention. If the retrieval mechanism is periodic in nature, then the ST author will need to include an iteration of FCS for storage of revocation information; storage of certificates is covered in FCS_CKM. The import of certificates and certificate chains is not included in this requirement, but is covered in FIA_X509 and FMT_MOF.

**Evaluation Activities**

*FCS_SMIME_EXT.1*

***TSS***

*The evaluator shall verify that the version of S/MIME implemented by the email client is present in the TSS. The evaluator shall also verify that the algorithms supported are specified, and that the algorithms specified are those listed for this component.*

*The evaluator shall verify that the TSS describes the ContentEncryptionAlgorithmIdentifier and whether the required behavior is performed by default or may be configured.*

*The evaluator shall verify that the TSS describes the digestAlgorithm and whether the required behavior is performed by default or may be configured.*

*The evaluator shall verify that the TSS describes the AlgorithmIdentifier and whether the required behavior is performed by default or may be configured.*

*The evaluator shall verify that the TSS describes the retrieval mechanisms for both certificates and certificate revocation as well as the frequency at which these mechanisms are implemented.*

***Guidance***

*The evaluator shall review the operational guidance to ensure that it contains instructions on configuring the email client such that it complies with the description in the TSS.*

*If the TSS indicates that the algorithms in FCS_SMIME_EXT.1.2 must be configured to meet the requirement, the evaluator shall verify that the operational guidance includes the configuration of this ID.*

*If the TSS indicates that the algorithms in FCS_SMIME_EXT.1.3 must be configured to meet the requirement, the evaluator shall verify that the operational guidance includes the configuration.*

*If the TSS indicates that the algorithms in FCS_SMIME_EXT.1.4 must be configured to meet the requirement, the evaluator shall verify that the operational guidance includes the configuration of this ID.*

*If the TSS indicates that the mechanisms in FCS_SMIME_EXT.1.7 are configurable, the evaluator shall verify that the operational guidance includes the configuration of these mechanisms.*

***Tests***

*The evaluator shall perform the tests listed below. These tests can be performed in conjunction with the tests specified in FIA_X509_EXT.1 (defined in the Base-PP) for certificate/certificate chain verification and in FDP_NOT_EXT.1.*

- ***Test 1:*** *The evaluator shall both send and receive a message with no protection (no signature or encryption) and verify that the message is transmitted properly and can be viewed at the receiving agent. This transmission can be performed as part of a number of mechanisms; it is sufficient to observe that the message arrives at the intended recipient with the same content as when sent.*
- ***Test 2:*** *The evaluator shall both send and receive a signed message using each of the algorithms specified in the ST corresponding to the requirement and verify that the signature is valid for both received and sent messages. After verifying the signatures are valid, the evaluator shall send a signed message using each of the algorithms specified in the ST and use a man-in-the-middle tool to modify at least one byte of the message such that the signature is no longer valid. This can be done by modifying the content of the message over which the signature is calculated or by modifying the signature itself. The evaluator shall then verify that the received message fails the signature validation check.*
- ***Test 3:*** *The evaluator shall send an encrypted message from the TOE to an OE receiver using each of the algorithms specified in the ST. The evaluator shall verify that each message is encrypted and the OE receiver can successfully decrypt each message. The evaluator shall then use the OE receiver to send an encrypted reply back to the TOE for each message sent at the start of this test. The evaluator shall verify that each reply is encrypted and the TOE can successfully decrypt each reply.*
- ***Test 4:*** *The evaluator shall verify that the contents are encrypted in transit and that the received message decrypts.*
- ***Test 5:*** *After verifying the message decrypts, the evaluator shall send an encrypted message using each of the algorithms specified in the ST and use a man-in-the-middle tool to modify at least one byte of the message such that the encryption is no longer valid. The evaluator shall then verify that the received message fails to decrypt.*
- ***Test 6:*** *The evaluator shall send an encrypted message to the email client using an encryption algorithm not supported according to the signatureAlgorithm field. The evaluator shall verify that the email client does not display/decrypt the contents of the message.*
- ***Test 7:*** *The evaluator shall send a signed message to the email client using a signature algorithm not supported according to the digestAlgorithm ID (e.g., SHA1). The evaluator shall then verify that the email client provides a notification that the contents cannot be verified because the signature algorithm is not supported.*
- ***Test 8:*** *The evaluator shall send an encrypted message to the email client using an encryption algorithm not supported according to the AlgorithmIdentifier field. The evaluator shall then verify that the email client does not display/decrypt the contents of the message.*
- ***Test 9:*** *The evaluator shall send the email client a message signed by a certificate without the digitalSignature bit set. The evaluator shall then verify that the email client notifies the user that the signature is invalid.*
- ***Test 10:*** *The evaluator shall send the email client a message signed by a certificate without the Email Protection purpose in the extendedKeyUsage. The evaluator shall then verify that the email client notifies the user that the signature is invalid.*
- ***Test 11:*** *The evaluator shall verify that the email client uses OCSP or downloads the CRL at the assigned frequency.*

### 5.2.2 User Data Protection (FDP)

#### FDP_NOT_EXT.1 Notification of S/MIME Status

FDP_NOT_EXT.1.1

The TSF shall display a notification of the S/MIME status of received emails upon viewing.

**Application Note:** S/MIME status is whether the email has been signed or encrypted and whether the signature can be verified and the associated certificate can be validated. This notification must at least display when the email content is viewed. Many implementations also display the S/MIME status of each email when all emails are viewed as a list.

**Evaluation Activities**

*FDP_NOT_EXT.1*
***TSS***
*The evaluator shall ensure that the TSS describes notifications of S/MIME status, including whether S/MIME status is also indicated upon viewing a list of emails.*

***Guidance***
*The evaluator shall verify that the operational guidance provides a description (with appropriate visual figures) of the S/MIME status notification(s), including how each of the following are indicated: encryption, verified and validated signature, and unverified and unvalidated signature.*

***Tests***
*The evaluator shall perform the following tests and may perform them in conjunction with the tests for FCS_SMIME_EXT.1:*

- ***Test 1:*** *The evaluator shall send the client an unencrypted and unsigned email and verify that no notifications are present upon viewing.*
- ***Test 2:*** *The evaluator shall send the client an encrypted email and verify that the encrypted notification is present upon viewing.*
- ***Test 3:*** *The evaluator shall send the client a valid signed email and verify that the signed notification is present upon viewing.*
- ***Test 4:*** *The evaluator shall send the client an invalid signed email (for example, using a certificate that does not contain the correct email address or a certificate that does not chain to the root store) and verify that the invalid signature notification is present upon viewing.*

#### FDP_SMIME_EXT.1 S/MIME

FDP_SMIME_EXT.1.1

The TSF shall use S/MIME to sign, verify, encrypt, and decrypt mail.

**Application Note:** Note that this requirement does not mandate that S/MIME be used for all incoming/outgoing messages, or that the email client automatically encrypt or sign/verify all sent or received messages. This requirement only specifies that the mechanism for digital signature and encryption must be S/MIME.

**Evaluation Activities**

*FDP_SMIME_EXT.1*
***TSS***
*The evaluator shall verify that the TSS contains a description of the S/MIME implementation and its use to protect mail from undetected modification using digital signatures and unauthorized disclosure using encryption. The evaluator shall also verify that the TSS describes whether signature verification and decryption occur at receipt or viewing of the message contents, and whether messages are stored with their S/MIME envelopes.*

***Guidance***
*The evaluator shall ensure that the operational guidance includes instructions for configuring a certificate for S/MIME use and instructions for signing and encrypting email.*

***Tests***
*Tests for this component are performed in conjunction with tests for FCS_SMIME_EXT.1 and FDP_NOT_EXT.1.*

### 5.2.3 Identification and Authentication (FIA)

#### FIA_X509_EXT.3 X.509 Authentication and Encryption

FIA_X509_EXT.3.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support encryption and authentication for S/MIME.

FIA_X509_EXT.3.2

The TSF shall prevent the establishment of a trusted communication channel when the peer certificate is deemed invalid.

**Application Note:** Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FIA_X509_EXT.3.3

The TSF shall prevent the installation of code if the code signing certificate is deemed invalid.

FIA_X509_EXT.3.4

The TSF shall prevent the encryption of email if the email protection certificate is deemed invalid.

FIA_X509_EXT.3.5

The TSF shall prevent the signing of email if the email protection certificate is deemed invalid.

**Evaluation Activities**

*FIA_X509_EXT.3*
***TSS***
*The evaluator shall check the TSS to ensure that it describes how the email client chooses which certificates to use so that the email client can use the certificates.*

*The evaluator shall examine the TSS to confirm that it describes the behavior of the email client when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel and protecting email.*

***Guidance***
*The evaluator shall verify that the administrative guidance contains any necessary instructions for configuring the operating environment so that the email client can use the certificates.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall perform Test 1 for each function listed in FIA_X509_EXT.2.1 (from the Base-PP) that requires the use of certificates. The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. The evaluator shall then load into the platform's root store any certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds.*
- ***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 email client is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 (from the Base-PP) 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.*

### 5.2.4 Security Management (FMT)

#### FMT_MOF_EXT.1 Management of Functions Behavior

FMT_MOF_EXT.1.1

The TSF shall be capable of performing the following management functions, controlled by the user or administrator as shown:

- X: Mandatory
- O: Optional

| # | Management Function | Administrator | User |
|---|---|---|---|
| 1 | Enable/disable downloading embedded objects globally and by [**selection**: *domain, sender, no other method*] | O | O |
| 2 | Enable/disable plaintext-only mode globally and by [**selection**: *domain, sender, no other method*] | O | O |
| 3 | Enable/disable rendering and execution of attachments globally and by [**selection**: *domain, sender, no other method*] | O | O |
| 4 | Enable/disable email notifications | O | O |
| 5 | Configure a certificate repository for encryption | O | O |
| 6 | Configure whether to establish a trusted channel or disallow establishment if the email client cannot establish a connection to determine the validity of a certificate | O | O |
| 7 | Configure message sending/receiving to only use cryptographic algorithms defined in FCS_SMIME_EXT.1 | O | O |
| 8 | Configure CRL retrieval frequency | O | O |
| 9 | Enable/disable support for add-ons | O | O |
| 10 | Change password/passphrase authentication credential | O | O |
| 11 | Disable key recovery functionality | O | O |
| 12 | Configure cryptographic functionality | O | O |
| 13 | [**assignment**: *Other management functions*] | O | O |

**Application Note:** For these management functions, the term "Administrator" refers to the administrator of a non-mobile device or the device owner of a mobile device. The Administrator is responsible for management activities, including setting the policy that is applied by the enterprise on the email client. The Administrator could be acting remotely and could be the MTA administrator acting through a centralized management console or dashboard. Applications used to configure enterprise policy should have their own identification and authorization and additional security requirements to ensure that the remote administration is trusted.

The intent of this requirement is to allow the Administrator to configure the email client with a policy that may not be over-ridden by the user. If the Administrator has not set a policy for a particular function, the user may still perform that function. Enforcement of the policy is done by the email client itself, or the email client and the email client platform in coordination with each

other.

The function to configure whether to establish a trusted channel corresponds to the functionality described in FIA_X509_EXT.2.2 (from the Base-PP). The Administrator has the option of accepting or rejecting all certificates that cannot be validated, accepting a given certificate that cannot be validated, or not accepting a given certificate that cannot be validated. Depending on the choice that the Administrator has made in FIA_X509_EXT.2.2 (from the Base-PP), the trusted connection will either be allowed for all certificates that cannot be validated, disallowed for all certificates that cannot be validated, allowed for a given certificate that cannot be validated, or disallowed for a given certificate that cannot be validated.

If password or passphrase authorization factors are implemented by the email client, then the appropriate "change" selection must be included.

If the email client provides configurability of the cryptographic functions (for example, key size), then "configure cryptographic functionality" will be included, and the specifics of the functionality offered can either be written in this requirement as bullet points, or included in the TSS. This applies even if the configuration is in the form of parameters that may be passed to cryptographic functionality implemented on the TOE platform.

If the email client does include a key recovery function, the email client must provide the capability for the user to turn this functionality off so that no recovery key is generated and no keys are permitted to be exported.

**Evaluation Activities**

*FMT_MOF_EXT.1*
*The evaluation activities for this component will be driven by the selections made by the ST author. If a capability is not selected in the ST, the noted evaluation activity does not need to be performed.*
***TSS***
*The evaluator shall verify that the TSS describes those management functions which may only be configured by the email client platform administrator and cannot be overridden by the user when set according to policy.*

*Change Password: The evaluator shall examine the Operational Guidance to ensure that it describes how the password/passphrase-based authorization factor is to be changed.*

*Disable Key Recovery: If the email client supports key recovery, this must be stated in the TSS. The TSS shall also describe how to disable this functionality. This includes a description of how the recovery material is provided to the recovery holder.*

*Cryptographic Configuration: The evaluator shall determine from the TSS for other requirements (FCS_*) what portions of the cryptographic functionality are configurable.*

***Guidance***
*The evaluator shall examine the operational guidance to verify that it includes instructions for an email client platform administrator to configure the functions listed in FMT_MOF_EXT.1.1.*

*Disable Key Recovery: If the email client supports key recovery, the guidance for disabling this capability shall be described in the operational guidance.*

*Cryptographic Configuration: The evaluator shall review the operational guidance to determine that there are instructions for manipulating all of the claimed mechanisms.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall verify that all functions perform as intended by enabling, disabling, and configuring the functions.*
- ***Test 2:*** *The evaluator shall set management functions which are controlled by the (enterprise) administrator and cannot be overridden by the user. The evaluator shall apply these functions to the client, attempt to override each setting as the user, and ensure that the email client does not permit it.*
- ***Test 3:*** *[Conditional: the TSF has a key recovery capability] The evaluator shall devise a test that ensures that the key recovery capability has been or can be disabled following the guidance provided by the vendor*

### 5.2.5 Protection of the TSF (FPT)

**FPT_AON_EXT.1 Support for Only Trusted Add-ons**

FPT_AON_EXT.1.1

The TSF shall include the capability to load [**selection**: *trusted add-ons*, *no add-ons*].

**Application Note:** If "trusted add-ons" is selected in FPT_AON_EXT.1.1, the TOE must also claim the selection-based SFR FPT_AON_EXT.2.

If the email client does not include support for installing only trusted add-ons, this requirement can be met by demonstrating the ability to disable all support for add-ons as specified in FMT_MOF_EXT.1.

**Evaluation Activities**

*FPT_AON_EXT.1*
***TSS***
*The evaluator shall verify that the TSS describes whether the email client is capable of loading trusted add-ons.*

***Guidance***
*The evaluator shall examine the operational guidance to verify that it includes instructions on loading trusted add-on sources.*

***Tests***
*The evaluator shall create or obtain an untrusted add-on and attempt to load it. The evaluator shall then verify that the untrusted add-on is rejected and cannot be loaded.*

### 5.2.6 Trusted Path/Channels (FTP)

**FTP_ITC_EXT.1 Inter-TSF Trusted Channel**

FTP_ITC_EXT.1.1

The TSF shall initiate or receive communication via the trusted channel.

FTP_ITC_EXT.1.2

The TSF shall communicate via the trusted channel for [**selection**:

- *IMAP*,
- *SMTP*,
- *POP*,
- *MAPI Extensions for HTTP*,
- *MAPI/RPC*,
- *ActiveSync*,
- *[**assignment**: other protocol (reference RFC or specification)]*

].

**Application Note:** FIA_SASL_EXT.1 depends upon the selection(s) made here. For example, if *POP* is chosen, then FIA_SASL_EXT.1 must be included in the ST. Selections must include at least one sending and one receiving protocol. If the assignment is used, the ST author must also include a reference for the protocol (e.g., an RFC number).

**Evaluation Activities**

*FTP_ITC_EXT.1*
***TSS***
*The evaluator shall examine the TSS to determine that it describes the details of the email client connecting to a Mail Transfer Agent in terms of the trusted connection (i.e., TLS) according to FTP_DIT_EXT.1 in the Base-PP, along with email client-specific options or procedures that might not be reflected in the specification.*

***Guidance***
*The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to the Mail Transfer Agent.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:** The evaluator shall ensure that the email client is able to initiate or receive communications using any selected or assigned protocols specified in the requirement over TLS, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- ***Test 2:** The evaluator shall ensure that the email client is able to initiate or receive communications with a Mail Transfer Agent using any assigned protocols specified in the requirement over TLS, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- ***Test 3:** The evaluator shall ensure, for each communication channel with an authorized IT entity in tests 1 and 2, the channel data is not sent in plaintext. To perform this test, The evaluator shall use a sniffer and a packet analyzer. The packet analyzer must indicate that the protocol in use is TLS.*

## 5.3 TOE Security Functional Requirements Rationale

The following rationale provides justification for each security objective for the TOE, showing that the SFRs are suitable to meet and achieve the security objectives:

**Table 2: SFR Rationale**

| OBJECTIVE | ADDRESSED BY | RATIONALE |
|---|---|---|
| O.MANAGEMENT | FDP_NOT_EXT.1, FMT_MOF_EXT.1, FDP_NOT_EXT.2 (optional), FDP_REN_EXT.1 (optional) | FDP_NOT_EXT.1 supports the objective by defining a mechanism for users to determine whether a given email has been signed or encrypted.<br>FMT_MOF_EXT.1 supports the objective by defining the technology-specific management functions that may exist for email client applications.<br>FDP_NOT_EXT.2 supports the objective by optionally requiring the TSF to enumerate the URI of embedded links in emails so that a user can determine the source of the link.<br>FDP_REN_EXT.1 supports the objective by optionally defining a plaintext-only operational mode that does not allow a user to interact with embedded content in an email message. |
| O.PROTECTED_STORAGE | FCS_CKM_EXT.3, FCS_CKM_EXT.4, FCS_KYC_EXT.1, FCS_IVG_EXT.1 (optional), FCS_NOG_EXT.1 (optional), FCS_SAG_EXT.1 (optional), FDP_PST_EXT.1 (optional), | FCS_CKM_EXT.3 supports the objective by defining the mechanism by which the TSF |

| | | |
|---|---|---|
| | FCS_CKM_EXT.5 (selection-based), FCS_COP_EXT.2 (selection-based), FCS_SMC_EXT.1 (selection-based) | protects sotred key data from unauthorized disclosure.<br>FCS_CKM_EXT.4 supports the objective by defining the mechanism by which the TSF securely destroys stored key data.<br>FCS_KYC_EXT.1 supports the objective by defining any key chain that the TSF implements to protect a root encryption key.<br>FCS_IVG_EXT.1 supports the objective by optionally specifying the initialization vectors used for various cryptographic modes if the TOE supports any of these modes.<br>FCS_NOG_EXT.1 supports the objective by optionally defining the minimum nonce size if the TSF uses any cryptographic algorithms that require the use of nonces.<br>FCS_SAG_EXT.1 supports the objective by optionally defining the supported methods for salt generation if the TSF uses any cryptographic algorithms that require the use of salts.<br>FDP_PST_EXT.1 supports the objective by optionally defining the ability of the TOE to operate without persistently storing certain types of data at all.<br>FCS_CKM_EXT.5 supports the objective by optionally defining the mechanism by which the TSF can derive key material using a user-supplied password credential.<br>FCS_COP_EXT.2 supports the objective by defining the supported key wrap mechanisms if the TSF uses key wrapping as part of maintaining the a key chain.<br>FCS_SMC_EXT.1 supports the objective by defining the supported key combination mechanisms if the TSF uses key combining as part of maintaining the a key chain. |
| O.PROTECTED_COMMS | FCS_CKM_EXT.1 (modified from Base-PP), FCS_RBG_EXT.1 (modified from Base-PP), FCS_TLS_EXT.1 (modified from TLS Package), FCS_TLSC_EXT.1 (from TLS package), FIA_X509_EXT.1 (from Base-PP), FIA_X509_EXT.2 (from Base-PP), FTP_DIT_EXT.1 (modified from Base-PP), FCS_SMIME_EXT.1, FDP_SMIME_EXT.1, FIA_X509_EXT.3, FTP_ITC_EXT.1, FIA_SASL_EXT.1 (selection-based) | FCS_CKM_EXT.1 supports the objective by requiring that the TSF provide or invoke a cryptographic function for asymmetric key generation.<br>FCS_RBG_EXT.1 supporst the objective by requiring that the TSF provide or invoke a DRBG for secure key generation.<br>FCS_TLS_EXT.1 supports the objective by requiring the TSF to implement TLS as a client.<br>FCS_TLSC_EXT.1 supports the objective by |

| | | |
|---|---|---|
| | | requiring the TSF to implement TLS client functionality in a specified manner.<br><br>FIA_X509_EXT.1 supports the objective by requiring the TSF to implement or invoke an X.509 certificate validation service.<br><br>FIA_X509_EXT.2 supports the objective by defining the TOE's use of X.509 certificates and what behavior the TOE takes when the revocation status of a certificate cannot be determined.<br><br>FTP_DIT_EXT.1 supports the objective by specifying the trusted communications channels used by the TOE to protect data in transit.<br><br>FCS_SMIME_EXT.1 supports the objective by defining the TOE's cryptographic implementation of S/MIME to both assert and validate the confidentiality and integrity of secure email messages.<br><br>FDP_SMIME_EXT.1 supports the objective by requiring the TSF to use S/MIME to protect email message data in transit.<br><br>FIA_X509_EXT.3 supports the objective by requiring the TSF to support the use of X.509 certificates for S/MIME.<br><br>FTP_ITC_EXT.1 supports the objective by specifying the trusted communications the TSF must implement that are specific to email communications.<br><br>FIA_SASL_EXT.1 supports the objective by specifying how SASL is implemented in the case where the TOE claims to support it. |
| O.ADDON_INTEGRITY | FPT_AON_EXT.1, FPT_AON_EXT.2 (selection-based) | FPT_AON_EXT.1 supports the objective by specifying whether or not the TSF has the ability to load add-ons.<br><br>FPT_AON_EXT.2 supports the objective by defining a cryptographic method for the TSF to validate the integrity of add-ons if the TOE supports their use. |

# 6 Consistency Rationale

## 6.1 Protection Profile for

### 6.1.1 Consistency of TOE Type

If this PP-Module is used to extend the App PP, the TOE type for the overall TOE is still a software application. The TOE boundary is simply extended to include the email client functionality that is built into the application so that additional security functionality is claimed within the scope of the TOE.

### 6.1.2 Consistency of Security Problem Definition

| PP-Module Threat, Assumption, OSP | Consistency Rationale |
|---|---|
| T.FLAWED_ADDON | The threat of a user installing a flawed add-on is consistent with the T.LOCAL_ATTACK threat from the Base-PP. A flawed addon, whether crafted deliberately or unintentionally, could cause the product to operate in a manner where it or its platform can be compromised. |
| T.NETWORK_ATTACK | This threat comes directly from the Base-PP. |
| T.NETWORK_EAVESDROP | This threat comes directly from the Base-PP. |
| T.PHYSICAL_ACCESS | This threat comes directly from the Base-PP. |

### 6.1.3 Consistency of Objectives

The objectives for the TOEs are consistent with the PP based on the following rationale:

| PP-Module TOE Objective | Consistency Rationale |
|---|---|
| O.MANAGEMENT | This objective is an enhancement to the O.MANAGEMENT objective defined in the Base-PP, specifically in regards to the secure administration of functions that are specific to email client applications. |
| O.PROTECTED_STORAGE | This objective is an enhancement to the O.PROTECTED_STORAGE objective defined in the Base-PP, specifically in regards to the data at rest that is specified to email client applications. |
| O.PROTECTED_COMMS | This objective is an enhancement to the O.PROTECTED_COMMS objective defined in the Base-PP, specifically in regards to the data in transit that is specified to email client applications. |
| O.ADDON_INTEGRITY | This objective is an enhancement to the O.INTEGRITY objective defined in the Base-PP. Where O.INTEGRITY is concerned with the integrity of the TOE application, O.ADDON_INTEGRITY is concerned with the integrity of third-party addons that can be loaded into the TOE. |

This PP-Module does not define any objectives for the TOE's operational environment.

### 6.1.4 Consistency of Requirements

This PP-Module identifies several SFRs from the PP that are needed to support Email Clients functionality. This is considered to be consistent because the functionality provided by the PP is being used for its intended purpose. The PP-Module also identifies a number of modified SFRs from the PP that are used entirely to provide functionality for Email Clients. The rationale for why this does not conflict with the claims defined by the PP are as follows:

| PP-Module Requirement | Consistency Rationale |
|---|---|
| **Modified SFRs** | |
| FCS_CKM_EXT.1 | This SFR is changed from its definition in the Base-PP to remove one of the available selection options because it will never apply in the case where the TOE conforms to this PP-Module. |
| FCS_RBG_EXT.1 | This SFR is changed from its definition in the Base-PP to remove one of the available selection options because it will never apply in the case where the TOE conforms to this PP-Module. |
| FIA_X509_EXT.1 | This SFR is unchanged from its definition in the Base-PP; the SFR is recategorized from selection-based to mandatory when the TOE conforms to this PP-Module. |
| FIA_X509_EXT.2 | This SFR is unchanged from its definition in the Base-PP; the SFR is recategorized from selection-based to mandatory when the TOE conforms to this PP-Module. |
| FTP_DIT_EXT.1 | This SFR is changed from its definition in the Base-PP to modify the selection options such that some options are mandated if another selection is chosen and some are removed entirely, due to the specific cryptographic needs of email client applications. |
| **Mandatory SFRs** | |
| FCS_CKM_EXT.3 | This SFR defines how keys and key material are saved by the email client. It does not impact the App PP functionality. |
| FCS_CKM_EXT.4 | This SFR defines how email messages are formatted when sent and received by the client. It does not impact the App PP functionality. |
| FCS_KYC_EXT.1 | This SFR defines how email clients maintain key chains. It does not impact the App PP functionality. |
| FCS_SMIME_EXT.1 | This SFR defines how email messages are formatted when sent and received by the client. It does not impact the App PP functionality. |
| FDP_NOT_EXT.1 | This SFR defines the behavior an email client exhibits when a message is received. It does not impact the App PP functionality. |

| | |
|---|---|
| FDP_SMIME_EXT.1 | This SFR defines the format an email client shall use as output for cryptographic operations. It does not impact the App PP functionality. |
| FIA_X509_EXT.3 | This SFR defines the format an email client shall use for certificates to perform encryption and authentication. It does not impact the App PP functionality. |
| FMT_MOF_EXT.1 | This SFR defines a specific set of management functions for an email client. It does not impact the App PP functionality. |
| FPT_AON_EXT.1 | This SFR defines what types of plugins an email client may use. It does not impact the App PP functionality. |
| FTP_ITC_EXT.1 | This SFR defines which channels for an email client must be considered trusted. It does not impact the App PP functionality. |
| **Optional SFRs** | |
| FCS_IVG_EXT.1 | This SFR defines how clients generate IVs for cryptographic operations. It does not impact functionality described by the Base-PP. |
| FCS_NOG_EXT.1 | This SFR defines how clients generate nonces for cryptographic operations. It does not impact functionality described by the Base-PP. |
| FCS_SAG_EXT.1 | This SFR defines how clients generate salts for cryptographic operations. It does not impact functionality described by the Base-PP. |
| FDP_NOT_EXT.2 | This SFR defines how clients display URIs in embedded links. It does not impact functionality described by the Base-PP. |
| FDP_PST_EXT.1 | This SFR defines the persistent information that must be stored for email client functionality to work as intended. It does not impact functionality described by the Base-PP. |
| FDP_REN_EXT.1 | This SFR defines functionality to display message content. It does not impact functionality described by the Base-PP. |
| **Selection-based SFRs** | |
| FCS_CKM_EXT.5 | This SFR defines restrictions on password composition and key derivation mechanisms. It defines functionality similar to FCS_CKM.1(3) in the Base-PP but has additional details specific to the composition of the actual password authentication factor, rather than just defining a method for key derivation. |
| FCS_COP_EXT.2 | This SFR defines how clients wrap keys. It does not impact functionality described by the Base-PP. |
| FCS_SMC_EXT.1 | This SFR defines how clients combine keys. It does not impact functionality described by the Base-PP. |
| FIA_SASL_EXT.1 | This SFR defines an alternate method of transmitting messages. It does not impact functionality described by the Base-PP. |
| FPT_AON_EXT.2 | This SFR defines how email clients verify Add-Ons. It does not impact functionality described by the Base-PP. |
| **Objective SFRs** | |
| This PP-Module does not define any Objective requirements. | |
| **Implementation-Dependent SFRs** | |
| This PP-Module does not define any Implementation-Dependent requirements. | |

# Appendix A - Optional SFRs

## A.1 Strictly Optional Requirements

### A.1.1 Cryptographic Support (FCS)

#### FCS_IVG_EXT.1 Initialization Vector Generation

FCS_IVG_EXT.1.1

The TSF shall create IVs in the following manner: [**selection**:

- *CBC: IVs shall be non-repeating,*
- *CCM: IV shall be non-repeating,*
- *XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,*
- *GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key.*

]

**Application Note:** FCS_IVG_EXT.1.1 specifies how the IV should be handled for each encryption mode. CBC, XTS, and GCM are allowed for AES encryption of the data. AES-CCM is an allowed mode for Key Wrapping.

**Evaluation Activities**

*FCS_IVG_EXT.1*
***TSS***
*The evaluator shall ensure the TSS describes how IVs and tweaks are handled (based on the AES mode). The evaluator shall confirm that the IVs and tweaks meet the stated requirements.*

*If the platform provides the IV generation, then the evaluator shall examine the TSS to verify that it describes how the IV generation is invoked.*
***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

#### FCS_NOG_EXT.1 Cryptographic Nonce Generation

FCS_NOG_EXT.1.1

The TSF shall only use unique nonces with a minimum size of [*64*] bits.

**Evaluation Activities**

*FCS_NOG_EXT.1*
***TSS***
*The evaluator shall verify that the TSS describes how unique nonces are created.*
***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

#### FCS_SAG_EXT.1 Cryptographic Salt Generation

FCS_SAG_EXT.1.1

The TSF shall only use salts that are generated by a [**selection**:

- *DRBG as specified in [FCS_RBG_EXT.2 (as defined in the Base-PP)],*
- *DRBG provided by the host platform*

]

**Evaluation Activities**

*FCS_SAG_EXT.1*
***TSS***
*The evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generated using a DRBG as described in FCS_RBG_EXT.1 in [App PP] or by the Operational Environment. If an external function is used for this purpose, the evaluator shall ensure tha the TSS references the specific API that is called with inputs.*

*If the email client is relying on random bit generation from the host platform, the evaluator shall verify that the TSS includes the name/manufacturer of the external DRBG and describes the function call and parameters used when calling the external DRBG function. If different external DRBGs are used for different platforms, the evaluator shall ensure that the TSS identifies each RBG for each platform.*

*For all cases where the TSF relies on an external DRBG, the evaluator shall ensure that the TSS includes a short description of the TOE developer's assumption for the amount of entropy that is used to seed the external DRBG.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

### A.1.2 User Data Protection (FDP)

#### FDP_NOT_EXT.2 Notification of URI

FDP_NOT_EXT.2.1

The TSF shall display the full Uniform Resource Identifier (URI) of any

embedded links.

**Application Note:** Embedded links are HTML URI objects which may have a tag (such as a word, phrase, icon, or picture) that obfuscates the URI of the link. The intent of this requirement is to de-obfuscate the link. The URI may be displayed as a "mouse-over" event or may be rendered next to the tag.

**Evaluation Activities**

*FDP_NOT_EXT.2*
***TSS***
*The evaluator shall verify that the TSS includes a description of how embedded links are rendered and the method by which the URI of the link is displayed.*

***Guidance***
*The evaluator shall ensure that the operational guidance includes instructions (with any appropriate visual figures) for viewing the URI of an embedded link.*

***Tests***
*The evaluator shall send the client an HTML message with an embedded link whose tag is not the URI itself (for example, "click here"). The evaluator shall view the message and, following the instructions in the operational guidance, verify that the full URI of the embedded link is displayed.*

### FDP_PST_EXT.1 Storage of Persistent Information

FDP_PST_EXT.1.1

The TSF shall be capable of operating without storing persistent information to the client platform with the following exceptions: [**selection**: *credential information, administrator provided configuration information, certificate revocation information, no exceptions*].

**Application Note:** Any data that persists after the email client closes, including temporary files, is considered to be persistent data. Satisfying this requirement would require the use of a protocol such as IMAP or MAPI. It is not compatible with POP.

**Evaluation Activities**

*FDP_PST_EXT.1*
***TSS***
*The evaluator shall examine the TSS to determine that it describes all persistent information stored on the platform, and the locations on the platform where these data are stored. The evaluator shall confirm that the persistent data described is limited to the data identified in the selection.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*The evaluator shall operate the email client so that several messages, signed, encrypted, and unsigned, are processed. The evaluator shall also exercise functionality such as moving messages to folders, writing unsent drafts of messages, etc., as provided by the client. The evaluator shall then examine the client platform to determine that the only persistent information stored is that which is identified in the TSS.*

### FDP_REN_EXT.1 Rendering of Message Content

FDP_REN_EXT.1.1

The TSF shall have a plaintext-only mode which disables the rendering and execution of [**selection**:

- *HTML,*
- *JavaScript,*
- *[**assignment**: other embedded content types],*
- *no embedded content types*

].

**Application Note:** Plaintext-only mode prevents the automatic downloading, rendering and execution of images, external resources and embedded objects such as HTML or JavaScript objects. FMT_MOF_EXT.1.1 addresses configuration of this mode. The ST author must identify all content types supported by the email client through selections and assignments. If the email client only supports plaintext-only mode, no embedded content types should be selected.

**Evaluation Activities**

*FDP_REN_EXT.1*
***TSS***
*The evaluator shall ensure that the TSS describes plaintext-only mode for sending and receiving messages. The evaluator shall verify that the TSS describes whether the email client is capable of rendering and executing HTML or JavaScript. If the email client can render or execute HTML or JavaScript, this description shall indicate how the email client handles received messages that contain HTML or JavaScript while in plaintext-only mode, and the evaluator shall ensure that the description indicates that embedded objects of these types are not rendered or executed and images/external resources are not automatically downloaded.*

***Guidance***
*The evaluator shall examine the operational guidance and verify that it contains instructions for enabling plaintext-only mode.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:** [Conditional: HTML is selected in FDP_REN_EXT.1.1] The evaluator shall send a message to the client containing HTML embedded objects and shall verify that the HTML renders. The evaluator shall then enable plaintext-only mode and verify that the HTML does*

*not render.*

- ***Test 2:*** *[Conditional: JavaScript is selected in FDP_REN_EXT.1.1] The evaluator shall send a message to the client containing JavaScript embedded objects and shall verify that the JavaScript renders and executes. The evaluator shall then enable plaintext-only mode and verify that the JavaScript does not render or execute.*

## A.2 Objective Requirements

This PP-Module does not define any Objective SFRs.

## A.3 Implementation-based Requirements

This PP-Module does not define any Implementation-based SFRs.

# Appendix B - Selection-based Requirements

## B.1 Cryptographic Support (FCS)

### FCS_CKM_EXT.5 Cryptographic Key Derivation (Password/Passphrase Conditioning)

***The inclusion of this selection-based component depends upon a selection in .***

FCS_CKM_EXT.5.1

The TSF shall support a password/passphrase of up to [**assignment**: *maximum password size, positive integer of 64 or more*] characters used to generate a password authorization factor.

**Application Note:** The password/passphrase is represented on the host machine as a sequence of characters whose encoding depends on the TOE and the underlying OS. The ST author assigns the maximum size of the password/passphrase it supports; it must support at least 64 characters.

FCS_CKM_EXT.5.2

The TSF shall allow passwords to be composed of any combination of upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [**selection**: [***assignment**: other supported special characters], no other characters*]

FCS_CKM_EXT.5.3

The TSF shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [*HMAC-[**selection**: SHA-256, SHA-384, SHA-512]*], with [**assignment**: *positive integer of 4096 or more*] iterations, and output cryptographic key sizes [**selection**: *128, 256*] bits that meet the following: [NIST SP 800-132].

**Application Note:** The ST author selects the parameters based on the PBKDF used by the TSF. The password/passphrase must be conditioned into a string of bits that forms the submask to be used as input into a key. 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. 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.

Appendix A of SP 800-132 recommends setting the iteration count in order to increase the computation needed to derive a key from a password and, therefore, increase the workload of performing a password recovery attack. However, for this PP-Module, a minimum iteration count of 4096 is required in order to ensure that twelve bits of security is added to the password/passphrase value. A significantly higher value is recommended to ensure optimal security.

FCS_CKM_EXT.5.4

The TSF shall not accept passwords less than [**selection**: *a value settable by the administrator*, [***assignment**: minimum password length accepted by the TOE, must be >= 1*]] and greater than the maximum password length defined in FCS_CKM_EXT.5.1.

**Application Note:** This selection-based SFR is claimed when "a password as specified in FCS_CKM_EXT.5" is selected in FCS_KYC_EXT.1.1.

If the minimum password length is settable, then ST author chooses "a value settable by the administrator for this component," as well as the "configure password/passphrase complexity setting" item for FMT_SMF.1.1. If the minimum length is not settable, the ST author fills in the assignment with the minimum length the password must be (zero-length passwords are not allowed for compliant TOEs).

**Evaluation Activities**

*FCS_CKM_EXT.5*
***TSS***
*For FCS_CKM_EXT.5.1, there are two aspects of this component that require evaluation: passwords/passphrases of the length specified in the requirement (at least 64 characters) are supported, and that the characters that are input are subject to the selected conditioning function. These activities are separately addressed in the text below.*

*Support for Password/Passphrase length: The evaluator shall check to ensure that the TSS describes the allowable ranges for password/passphrase lengths, and that at least 64 characters may be specified by the user.*

*Support for PBKDF: The evaluator shall examine the password hierarchy TSS to ensure that the formation of all keys is described and that the key sizes match that described by the ST author.*

*The evaluator shall check that the TSS describes the method by which the password/passphrase 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 KEK as specified in FCS_CKM_EXT.4.*

*For the NIST SP 800-132-based conditioning of the password/passphrase, the required assurance activities will be performed when doing the assurance activities for the appropriate requirements (FCS_COP.1.1(4) from the [AppPP]). If any manipulation of the key is performed in forming the submask that will be used to form the FEK or KEK, that process shall be described in the TSS. For the claimed iteration count, the evaluator shall verify that the iteration count for PBKDFs performed by the TOE comply with NIST SP 800-132 by ensuring that the TSS contains a description of the estimated time required to derive key material from passwords and how the TOE increases the computation time for password-based key derivation (including but not limited to increasing the iteration count).*

***Guidance***
*The evaluator shall check the operational guidance to determine that there are instructions on*

*how to generate large passwords/passphrases, and instructions on how to configure the password/passphrase length (and optional complexity settings) to provide entropy commensurate with the keys that the authorization factor is protecting. This is important because many default settings for passwords/passphrases will not meet the necessary entropy needed as specified in this PP-Module*

***Tests***
*The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, and shall verify that the TOE's behavior is consistent with the requirements. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and minimum and maximum lengths listed in the requirement, are supported, and justify the subset of those characters chosen for testing.*

*Support for Password/Passphrase characteristics: In addition to the analysis above, the evaluator shall also perform the following tests on a TOE configured according to the Operational Guidance*

- ***Test 1:** The evaluator shall ensure that the TOE supports passwords/passphrases of exactly 64 characters.*
- ***Test 2:** The evaluator shall ensure that the TOE does not accept more than the maximum number of characters specified in FCS_CKM_EXT.5.1.*
- ***Test 3:** The evaluator shall ensure that the TOE does not accept less than the minimum number of characters specified in FCS_CKM_EXT.5.4. If the minimum length is settable by the administrator, the evaluator determines the minimum length or lengths to test.*
- ***Test 4:** The evaluator shall ensure that the TOE supports passwords consisting of all characters listed in FCS_CKM_EXT.5.2 and of varying lengths within the range specified in FCS_CKM_EXT.5.4.*

*No explicit testing of the formation of the submask from the input password is required.*

*For password conditioning, no explicit testing of the formation of the authorization factor from the input password/passphrase is required.*

### FCS_COP_EXT.2 Key Wrapping

***The inclusion of this selection-based component depends upon a selection in .***

FCS_COP_EXT.2.1

The TSF shall [**selection**:

- *use platform-provided functionality to perform Key Wrapping,*
- *implement functionality to perform Key Wrapping*

] in accordance with a specified cryptographic algorithm [**selection**:

- *AES Key Wrap,*
- *AES Key Wrap with Padding,*
- *RSA using the KTS-OAEP-basic scheme,*
- *RSA using the KTS-OAEP-receiver-confirmation scheme,*
- *ECC CDH*

] and the cryptographic key size [**selection**:

- *128 bits (AES),*
- *256 bits (AES),*
- *2048 (RSA),*
- *4096 (RSA),*
- *256-bit prime,*
- *modulus (ECC CDH),*
- *384-bit prime modulus (ECC CDH)*

] that meet the following: [**selection**:

- *"NIST SP 800-38F" for Key Wrap (section 6.2) and Key Wrap with Padding (section 6.3),*
- *"NIST SP 800-56B" for RSA using the KTS-OAEP-basic (section 9.2.3) and KTS-OAEP-receiver-confirmation (section 9.2.4) scheme, "NIST SP 800-56A rev 2" for ECC CDH (sections 5.6.1.2 and 6.2.2.2)*

].

**Application Note:** This selection-based SFR is claimed when any of the selections that explicitly reference FCS_COP_EXT.2 are selected in FCS_CKM_EXT.3.1 or FCS_KYC_EXT.1.1.

In the first selection, the ST author chooses the entity that performs the decryption/encryption. In the second selection, the ST author chooses the method used for encryption:

- Using one of the two AES-based Key Wrap methods specified in NIST SP 800-38F
- Using one of the two the KTS-OAEP schemes for RSA as described in NIST SP 800-56B (KTSOAEP-basic described in section 9.2.3)
- Using ECC CDH as described in NIST SP 800-56A section 6.2.2.2.

The third selection should be made to reflect the key size. 2048/4096 is used for the RSA-based schemes, while the size of the prime modulus is used for ECC-based schemes. Support for 256-bit AES key sizes will be required for products entering evaluation after Quarter 3, 2015. Based on the method(s) selected, the last selection should be used to select the appropriate reference(s).

### Evaluation Activities

*FCS_COP_EXT.2*
***TSS***
*The evaluator shall examine the TSS to ensure that it has a high-level description of how the key is protected and meets the appropriate specification.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

### FCS_SMC_EXT.1 Key Combining

***The inclusion of this selection-based component depends upon a selection in .***

FCS_SMC_EXT.1.1

The TSF shall combine submasks using the following method [**selection**:

- *exclusive OR (XOR)*,
- *SHA-256*,
- *SHA-512*

] to generate another key.

**Application Note:** This selection-based SFR is claimed when any of the selections that explicitly reference FCS_SMC_EXT.1 are selected in FCS_CKM_EXT.3.1 or FCS_KYC_EXT.1.1.

This requirement specifies the way that a product may combine the various submasks by using either an XOR or an approved SHA-hash.

**Evaluation Activities**

*FCS_SMC_EXT.1*
***TSS***
*If keys are XORed together to form an intermediate key, the evaluator shall verify that the TSS describes how this is performed (e.g., if there are ordering requirements, checks performed, etc.).*

*The evaluator shall also confirm that the TSS describes how the length of the output produced is at least the same as that of the data encryption key.*

***Guidance***
*There are no guidance EAs for this component.*

***Tests***
*There are no test EAs for this component.*

## B.2 Identification and Authentication (FIA)

### FIA_SASL_EXT.1 Simple Authentication and Security Layer (SASL)

***The inclusion of this selection-based component depends upon a selection in .***

FIA_SASL_EXT.1.1

The TSF shall implement support for Simple Authentication and Security Layer (SASL) that complies with RFC 4422.

**Application Note:** SASL is needed if the email implements SMTP to send messages. Clients that do not use SMTP (e.g., ActiveSync or MAPI) would not need to implement support for SASL.

FIA_SASL_EXT.1.2

The TSF shall support the POP3 CAPA and AUTH extensions for the SASL mechanism.

FIA_SASL_EXT.1.3

The TSF shall support the IMAP CAPABILITY and AUTHENTICATE extensions for the SASL mechanism.

FIA_SASL_EXT.1.4

The TSF shall support the SMTP AUTH extension for the SASL mechanism.

**Application Note:** This selection-based SFR is claimed when QQQQ.

In order for an email client to support PKI X.509 Certificates for POP3, IMAP, and SMTP as required in this document, the client must support the Simple Authentication and Security Layer (SASL) authentication method as described in RFC 4422, the AUTH and CAPA extensions for POP3, as described in RFC 5034, the AUTHENTICATION and CAPABILITY extensions for IMAP, as described in RFC 4959 and the AUTH extension for SMTP, as described in RFC 4954.

**Evaluation Activities**

*FIA_SASL_EXT.1*
***TSS***
*The evaluator shall examine the TSS to verify that it describes the details of the email client connecting to a Mail Transfer Agent in terms of the SASL connection, along with email client-specific options or procedures that might not be reflected in the specification.*

***Guidance***
*The evaluator shall confirm that the operational guidance contains instructions for establishing the connection to the Mail Transfer Agent.*

***Tests***
*The evaluator shall also perform the following tests:*

- ***Test 1:*** *The evaluator shall ensure that the email client is able to initiate communications using POP, IMAP, and SMTP and requiring SASL, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- ***Test 2:*** *The evaluator shall ensure, for each communication channel with an authorized IT entity in Test 1, that a valid SASL handshake is performed. To perform this test, The evaluator shall use a sniffer and a packet analyzer. The sniffer and packet analyzer must allow the evaluator to view the plaintext email protocol (e.g., proxy, loading the server private key). The evaluator shall identify the SASL handshake within the email protocol.*

## B.3 Protection of the TSF (FPT)

**FPT_AON_EXT.2 Trusted Installation and Update for Add-ons**

***The inclusion of this selection-based component depends upon a selection in FPT_AON_EXT.1.1.***

FPT_AON_EXT.2.1

The TSF shall [**selection**: *provide the ability, leverage the platform*] to provide a means to cryptographically verify add-ons using a digital signature mechanism and [**selection**: *published hash, no other functions*] prior to installation and update.

FPT_AON_EXT.2.2

The TSF shall [**selection**: *provide the ability, leverage the platform*] to query the current version of the add-on.

FPT_AON_EXT.2.3

The TSF shall prevent the automatic installation of add-ons.

**Application Note:** This selection-based SFR is claimed when "trusted add-ons" is selected in FPT_AON_EXT.1.1.

**Evaluation Activities**

*FPT_AON_EXT.2*
***TSS***
*The evaluator shall examine the TSS to verify that it states that the email client will reject add-ons from untrusted sources.*

***Guidance***
*The evaluator shall examine the operational guidance to verify that it includes instructions on how to configure the email client with trusted add-on sources.*

***Tests***
*The evaluator shall perform the following tests:*

- ***Test 1:*** *The evaluator shall create or obtain an add-on signed by a trusted source and attempt to install it. The evaluator shall then verify that the signature on the addon is valid and that the add-on can be installed.*
- ***Test 2:*** *The evaluator shall create or obtain an add-on signed with an invalid certificate and attempt to install it. The evaluator shall then verify that the signed addon is rejected and cannot be installed.*
- ***Test 3:*** *The evaluator shall create or obtain an add-on signed by a trusted source, modify the addon without resigning it, and attempt to install it. The evaluator shall then verify that the signed add-on is rejected and cannot be installed.*

# Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the PP-Module.

## C.1 Extended Components Table

All extended components specified in the PP are listed in this table:

**Table 3: Extended Component Definitions**

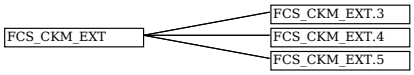
| Functional Class | Functional Components |
|---|---|
| Cryptographic Support (FCS) | FCS_CKM_EXT Cryptographic Key Management<br>FCS_COP_EXT Cryptographic Operation<br>FCS_IVG_EXT Initialization Vector Generation<br>FCS_KYC_EXT Cryptographic Key Chaining<br>FCS_NOG_EXT Cryptographic Nonce Generation<br>FCS_SAG_EXT Initialization Vector Generation<br>FCS_SMC_EXT Submask Combining<br>FCS_SMIME_EXT Secure/Multipurpose Internet Mail Extensions (S/MIME) |
| User Data Protection (FDP) | FDP_NOT_EXT Notifications<br>FDP_PST_EXT Storage of Persistent Information<br>FDP_REN_EXT Rendering of Message Conent<br>FDP_SMIME_EXT Use of Secure/Multipurpose Internet Mail Extensions (S/MIME) |
| Identification and Authentication (FIA) | FIA_SASL_EXT Simple Authentication and Security Layer (SASL)<br>FIA_X509_EXT X.509 Certificate Services |
| Security Management (FMT) | FMT_MOF_EXT Management of Functions Behavior |
| Protection of the TSF (FPT) | FPT_AON_EXT Add-Ons |
| Trusted Path/Channels (FTP) | FTP_ITC_EXT Inter-TSF Trusted Channel |

## C.2 Extended Component Definitions

### FCS_CKM_EXT Cryptographic Key Management

### Family Behavior

Components in this family define requirements for cryptographic key management beyond those which are specified in the Part 2 family FCS_CKM.




### Component Leveling

FCS_CKM_EXT.3, Protection of Key and Key Material, requires the TSF to identify the method that it uses to prevent the plaintext storage of secret key data.

### Management: FCS_CKM_EXT.3

No specific management functions are identified.

### Audit: FCS_CKM_EXT.3

There are no auditable events foreseen.

### FCS_CKM_EXT.3 Protection of Key and Key Material

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FCS_CKM_EXT.3.1

The TSF shall [**assignment**: *method of ensuring plaintext key data is not stored in non-volatile memory*].

### Component Leveling

FCS_CKM_EXT.4, Cryptographic Key Destruction, requires the TSF to identify the method that it uses to destroy key data.

### Management: FCS_CKM_EXT.4

No specific management functions are identified.

### Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

### FCS_CKM_EXT.4 Cryptographic Key Destruction

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FCS_CKM_EXT.4.1

The TSF shall [**assignment**: *key destruction method*] that meets the following:[**selection**:

- *NIST SP 800-88*,
- *no standard*

] for destroying all keying material and cryptographic security parameters when no longer needed.

### Component Leveling

FCS_CKM_EXT.5, Cryptographic Key Derivation (Password/Passphrase Conditioning), requires the TSF to support password/passphrase credentials with certain strength of secret characteristics and to support the

use of such credentials as an input to a password-based key derivation function.

### Management: FCS_CKM_EXT.5

The following actions could be considered for the management functions in FMT:

- Change password/passphrase authentication credential.
- Change password/passphrase minimum length.

### Audit: FCS_CKM_EXT.5

There are no auditable events foreseen.

### FCS_CKM_EXT.5 Cryptographic Key Derivation (Password/Passphrase Conditioning)

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_CKM_EXT.5.1

The TSF shall support a password/passphrase of up to [**assignment**: *maximum password size, positive integer of 64 or more*] characters used to generate a password authorization factor.

### FCS_CKM_EXT.5.2

The TSF shall allow passwords to be composed of any combination of upper case characters, lower case characters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")", and [**selection**: ***[assignment**: other supported special characters], no other characters*]

### FCS_CKM_EXT.5.3

The TSF shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [*HMAC-[**selection**: SHA-256, SHA-384, SHA-512]*], with [**assignment**: *positive integer of 4096 or more*] iterations, and output cryptographic key sizes [**selection**: *128, 256*] bits that meet the following: [NIST SP 800-132].

### FCS_CKM_EXT.5.4

The TSF shall not accept passwords less than [**selection**: *a value settable by the administrator, [**assignment**: minimum password length accepted by the TOE, must be >= 1]*] and greater than the maximum password length defined in FCS_CKM_EXT.5.1.

## FCS_KYC_EXT Cryptographic Key Chaining

### Family Behavior

Components in this family define requirements for protection of cryptographic key data through its storage in a hierarchical key chain.

### Component Leveling

FCS_KYC_EXT.1, Key Chaining, requires the TSF to identify the method that it uses to prevent the plaintext storage of secret key data.

### Management: FCS_KYC_EXT.1

No specific management functions are identified.

### Audit: FCS_KYC_EXT.1

There are no auditable events foreseen.

### FCS_KYC_EXT.1 Key Chaining

Hierarchical to: No other components.

Dependencies to: No dependencies.

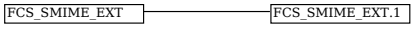
### FCS_KYC_EXT.1.1

The TSF shall maintain a key chain of: [**assignment**: *key hierarchy*] to the data encryption/decryption key(s) using the following method(s): [**assignment**: *key protection method*] while maintaining an effective strength of [**assignment**: *key strength*]

## FCS_SMIME_EXT Secure/Multipurpose Internet Mail Extensions (S/MIME)

### Family Behavior

Components in this family define requirements for the secure implementation of S/MIME.




### Component Leveling

FCS_SMIME_EXT.1, Secure/Multipurpose Internet Mail Extensions (S/MIME), requires the TSF to implement S/MIME in accordance with appropriate RFCs and using appropriate cryptographic functionality.

### Management: FCS_SMIME_EXT.1

The following actions could be considered for the management functions in FMT:

- Configure message sending/receiving to only use specified cryptographic algorithms.

### Audit: FCS_SMIME_EXT.1

There are no auditable events foreseen.

### FCS_SMIME_EXT.1 Secure/Multipurpose Internet Mail Extensions (S/MIME)

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

FIA_X509_EXT.1 X.509 Certificate Validation

### FCS_SMIME_EXT.1.1

The TSF shall implement both a sending and receiving S/MIME v4.0 Agent as defined in RFC 8551, using CMS as defined in RFCs 5652, 5754, and 3565.

### FCS_SMIME_EXT.1.2

The TSF shall transmit the ContentEncryptionAlgorithmIdentifier for AES-128 CBC, AES-256 CBC, and [**selection**: *AES-128 GCM*, *AES-256 GCM*, *no other*] as part of the S/MIME protocol.

### FCS_SMIME_EXT.1.3

The TSF shall present the digestAlgorithm field with the following Message Digest Algorithm identifiers [**assignment**: *message digest algorithm identifiers*] and no others as part of the S/MIME protocol.

### FCS_SMIME_EXT.1.4

The TSF shall present the signatureAlgorithm field with the following: sha256withRSAEncryption and [**assignment**: *signatureAlgorithm field values*] and no other algorithms as part of the S/MIME protocol.

### FCS_SMIME_EXT.1.5

The TSF shall support use of different private keys (and associated certificates) for signature and for encryption as part of the S/MIME protocol.

### FCS_SMIME_EXT.1.6

The TSF shall only accept a signature from a certificate with the digitalSignature bit set as part of the S/MIME protocol.

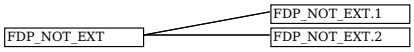
### FCS_SMIME_EXT.1.7

The TSF shall implement mechanisms to retrieve certificates and certificate revocation information [**selection**: *for each signed/encrypted message sent/received*, *[**assignment**: frequency]*] as part of the S/MIME protocol.

## FDP_NOT_EXT Notifications

### Family Behavior

Components in this family define requirements for the TSF's ability to notify users about potential insecure interactions with data.




### Component Leveling

FDP_NOT_EXT.1, Notification of S/MIME Status, requires the TSF to present the S/MIME status of received email messages.

### Management: FDP_NOT_EXT.1

No specific management functions are identified.

### Audit: FDP_NOT_EXT.1

There are no auditable events foreseen.

### FDP_NOT_EXT.1 Notification of S/MIME Status

Hierarchical to: No other components.

Dependencies to: FCS_SMIME_EXT.1 Secure/Multipurpose Internet Mail Extensions (S/MIME)

### FDP_NOT_EXT.1.1

The TSF shall display a notification of the S/MIME status of received emails upon viewing.

### Component Leveling

FDP_NOT_EXT.2, Notification of URI, requires the TSF to display the Uniform Resource Identifier (URI) of any embedded links.

### Management: FDP_NOT_EXT.2

No specific management functions are identified.

### Audit: FDP_NOT_EXT.2

There are no auditable events foreseen.

### FDP_NOT_EXT.2 Notification of URI

Hierarchical to: No other components.

Dependencies to: No dependencies.

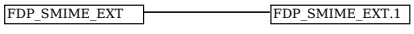
### FDP_NOT_EXT.2.1

The TSF shall display the full Uniform Resource Identifier (URI) of any embedded links.

## FDP_SMIME_EXT Use of Secure/Multipurpose Internet Mail Extensions (S/MIME)

### Family Behavior

Components in this family define requirements to implement S/MIME.




### Component Leveling

FDP_SMIME_EXT.1, S/MIME, requires the TSF to support S/MIME.

### Management: FDP_SMIME_EXT.1

No specific management functions are identified.

### Audit: FDP_SMIME_EXT.1

There are no auditable events foreseen.

### FDP_SMIME_EXT.1 S/MIME

Hierarchical to: No other components.

Dependencies to: FCS_SMIME_EXT.1 Secure/Multipurpose Internet Mail Extensions (S/MIME)

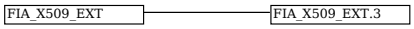
### FDP_SMIME_EXT.1.1

The TSF shall use S/MIME to sign, verify, encrypt, and decrypt mail.

### FIA_X509_EXT X.509 Certificate Services

### Family Behavior

Components in this family define requirements for the use of X.509 certifications in trusted communications.




### Component Leveling

FIA_X509_EXT.3, X.509 Authentication and Encryption, requires the TSF to use X.509 certificates for various functions.

### Management: FIA_X509_EXT.3

No specific management functions are identified.

### Audit: FIA_X509_EXT.3

There are no auditable events foreseen.

### FIA_X509_EXT.3 X.509 Authentication and Encryption

Hierarchical to: No other components.

Dependencies to: FIA_X509_EXT.1 X.509 Certificate Validation

### FIA_X509_EXT.3.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support encryption and authentication for S/MIME.

### FIA_X509_EXT.3.2

The TSF shall prevent the establishment of a trusted communication channel when the peer certificate is deemed invalid.

### FIA_X509_EXT.3.3

The TSF shall prevent the installation of code if the code signing certificate is deemed invalid.

### FIA_X509_EXT.3.4

The TSF shall prevent the encryption of email if the email protection certificate is deemed invalid.

### FIA_X509_EXT.3.5

The TSF shall prevent the signing of email if the email protection certificate is deemed invalid.

### FMT_MOF_EXT Management of Functions Behavior

### Family Behavior

Components in this family define requirements for technology-specific management functions that are not enumerated in the Part 2 family FMT_MOF.

### Component Leveling

FMT_MOF_EXT.1, Management of Functions Behavior, requires the TSF to implement management functions specified in the SFR.

### Management: FMT_MOF_EXT.1

No specific management functions are identified.

### Audit: FMT_MOF_EXT.1

There are no auditable events foreseen.

### FMT_MOF_EXT.1 Management of Functions Behavior

Hierarchical to: No other components.

Dependencies to: No dependencies.

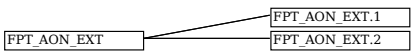
### FMT_MOF_EXT.1.1

The TSF shall be capable of performing the following management functions, controlled by the user or administrator as shown: [**assignment**: *list of management functions to be performed by role*].

### FPT_AON_EXT Add-Ons

### Family Behavior

Components in this family define requirements for the secure handling of add-ons that can be installed on top of the TOE.




### Component Leveling

FPT_AON_EXT.1, Support for Only Trusted Add-ons, requires the TSF to either support no add-ons or to only

support trusted add-ons.

### Management: FPT_AON_EXT.1

The following actions could be considered for the management functions in FMT:

- Enable/disable support for add-ons.

### Audit: FPT_AON_EXT.1

There are no auditable events foreseen.

### FPT_AON_EXT.1 Support for Only Trusted Add-ons

Hierarchical to: No other components.

Dependencies to: No dependencies

### FPT_AON_EXT.1.1

The TSF shall include the capability to load [**selection**: *trusted add-ons*, *no add-ons*].

### Component Leveling

FPT_AON_EXT.2, Trusted Installation and Update for Add-ons, requires the TSF to implement a method to verify the integrity of add-ons and ensure that untrusted or unknown add-ons are not loaded for use.

### Management: FPT_AON_EXT.2

No specific management functions are identified.

### Audit: FPT_AON_EXT.2

There are no auditable events foreseen.

### FPT_AON_EXT.2 Trusted Installation and Update for Add-ons

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

FPT_AON_EXT.1 Support for Only Trusted Add-Ons

### FPT_AON_EXT.2.1

The TSF shall [**selection**: *provide the ability*, *leverage the platform*] to provide a means to cryptographically verify add-ons using a digital signature mechanism and [**selection**: *published hash*, *no other functions*] prior to installation and update.

### FPT_AON_EXT.2.2

The TSF shall [**selection**: *provide the ability*, *leverage the platform*] to query the current version of the add-on.

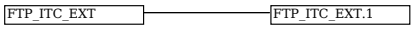
### FPT_AON_EXT.2.3

The TSF shall prevent the automatic installation of add-ons.

## FTP_ITC_EXT Inter-TSF Trusted Channel

### Family Behavior

Components in this family define technology-specific requirements for trusted communications that are not defined in the Part 2 family FTP_ITC.




### Component Leveling

FTP_ITC_EXT.1, Inter-TSF Trusted Channel, requires the TSF to identify the trusted channels it uses for communications with external entities.

### Management: FTP_ITC_EXT.1

No specific management functions are identified.

### Audit: FTP_ITC_EXT.1

There are no auditable events foreseen.

### FTP_ITC_EXT.1 Inter-TSF Trusted Channel

Hierarchical to: No other components.

Dependencies to: No dependencies.

### FTP_ITC_EXT.1.1

The TSF shall initiate or receive communication via the trusted channel.

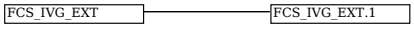
### FTP_ITC_EXT.1.2

The TSF shall communiate via the trusted channel for [**assignment**: *trusted channel protocol*].

## FCS_IVG_EXT Initialization Vector Generation

### Family Behavior

Components in this family define requirements for the secure generation of initialization vectors used in support of other cryptographic functions.




### Component Leveling

FCS_IVG_EXT.1, Initialization Vector Generation, requires the TSF to generate initialization vectors in a specified manner.

### Management: FCS_IVG_EXT.1

No specific management functions are identified.

### Audit: FCS_IVG_EXT.1

There are no auditable events foreseen.

### FCS_IVG_EXT.1 Initialization Vector Generation

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_IVG_EXT.1.1

The TSF shall create IVs in the following manner: [**assignment**: *IVs and methods of creation*].

## FCS_NOG_EXT Cryptographic Nonce Generation

### Family Behavior

Components in this family define requirements for the secure generation of nonces used in support of other cryptographic functions.

### Component Leveling

FCS_NOG_EXT.1, Cryptographic Nonce Generation, requires the TSF to generate nonces in a specified manner.

### Management: FCS_NOG_EXT.1

No specific management functions are identified.

### Audit: FCS_NOG_EXT.1

There are no auditable events foreseen.

### FCS_NOG_EXT.1 Cryptographic Nonce Generation

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

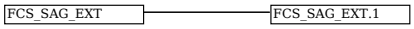
### FCS_NOG_EXT.1.1

The TSF shall only use unique nonces with a minimum size of [*64*] bits.

## FCS_SAG_EXT Initialization Vector Generation

### Family Behavior

Components in this family define requirements for the secure generation of salts used in support of other cryptographic functions.




### Component Leveling

FCS_SAG_EXT.1, Cryptographic Salt Generation, requires the TSF to generate salts in a specified manner.

### Management: FCS_SAG_EXT.1

No specific management functions are identified.

### Audit: FCS_SAG_EXT.1

There are no auditable events foreseen.

### FCS_SAG_EXT.1 Cryptographic Salt Generation

Hierarchical to: No other components.

Dependencies to: FCS_RBG_EXT.1 Random Bit Generation Services

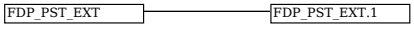
### FCS_SAG_EXT.1.1

The TSF shall only use salts that are generated by a [**assignment**: *trusted deterministic random bit generator*].

## FDP_PST_EXT Storage of Persistent Information

### Family Behavior

Components in this family define requirements for the enumeration of the minimum set of data the TSF must be able to store in order to implement its required functionality.




### Component Leveling

FDP_PST_EXT.1, Storage of Persistent Information, requires the TSF to identify the minimum set of data it can store on the TOE platform while maintaining functionality.

### Management: FDP_PST_EXT.1

No specific management functions are identified.

### Audit: FDP_PST_EXT.1

There are no auditable events foreseen.

### FDP_PST_EXT.1 Storage of Persistent Information

Hierarchical to: No other components.

Dependencies to: FDP_DAR_EXT.1 Encryption of Sensitive Application Data

### FDP_PST_EXT.1.1

The TSF shall be capable of operating without storing persistent information to the client platform with the following exceptions: [**assignment**: *data that the TSF must store persistently*].

### FDP_REN_EXT Rendering of Message Conent

### Family Behavior

Components in this family define requirements for the rendering of data presented to a user such that the risk of malicious data transmission is minimized.

### Component Leveling

FDP_REN_EXT.1, Rendering of Message Content, requires the TSF to implement a plaintext-only mode that prevents non-text content from being rendered.

### Management: FDP_REN_EXT.1

The following actions could be considered for the management functions in FMT:

- Enable/disable plaintext-only mode.

### Audit: FDP_REN_EXT.1

There are no auditable events foreseen.

### FDP_REN_EXT.1 Rendering of Message Content

Hierarchical to: No other components.

Dependencies to: No dependencies

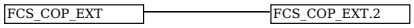
### FDP_REN_EXT.1.1

The TSF shall have a plaintext-only mode which disables the rendering and execution of [**assignment**: *embedded content types*].

### FCS_COP_EXT Cryptographic Operation

### Family Behavior

Components in this family define requirements for cryptographic operation beyond those which are specified in the Part 2 family FCS_COP.




### Component Leveling

FCS_COP_EXT.2, Key Wrapping, requires the TSF to implement key wrapping in a specified manner.

### Management: FCS_COP_EXT.2

No specific management functions are identified.

### Audit: FCS_COP_EXT.2

There are no auditable events foreseen.

### FCS_COP_EXT.2 Key Wrapping

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_COP_EXT.2.1

The TSF shall [**selection**:

- *use platform-provided functionality to perform Key Wrapping,*
- *implement functionality to perform Key Wrapping*

] in accordance with a specified cryptographic algorithm [**assignment**: *cryptographic algorithm*] and the cryptographic key size [**assignment**: *cryptographic key size*] that meet the following: [**assignment**: *list of standards*]

### FCS_SMC_EXT Submask Combining

### Family Behavior

Components in this family define requirements for key combination used in support of other cryptographic functions.

### Component Leveling

FCS_SMC_EXT.1, Key Combining, requires the TSF to implement submask combining in a specified manner.

### Management: FCS_SMC_EXT.1

No specific management functions are identified.

### Audit: FCS_SMC_EXT.1

There are no auditable events foreseen.

### FCS_SMC_EXT.1 Key Combining

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

### FCS_SMC_EXT.1.1

The TSF shall combine submasks using the following method [**selection**:

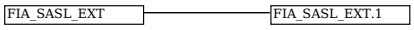
- *exclusive OR (XOR),*
- *SHA-256,*
- *SHA-512*

] to generate another key.

### FIA_SASL_EXT Simple Authentication and Security Layer (SASL)

### Family Behavior

Components in this family define requirements for the implementation of SASL.




### Component Leveling

FIA_SASL_EXT.1, Simple Authentication and Security Layer (SASL), requires the TSF to implement SASL in a manner that conforms to applicable standards.

### Management: FIA_SASL_EXT.1

No specific management functions are identified.

### Audit: FIA_SASL_EXT.1

There are no auditable events foreseen.

### FIA_SASL_EXT.1 Simple Authentication and Security Layer (SASL)

Hierarchical to: No other components.

Dependencies to: FCS_COP.1 Cryptographic Operation

#### FIA_SASL_EXT.1.1

The TSF shall implement support for Simple Authentication and Security Layer (SASL) that complies with RFC 4422.

#### FIA_SASL_EXT.1.2

The TSF shall support the POP3 CAPA and AUTH extensions for the SASL mechanism.

#### FIA_SASL_EXT.1.3

The TSF shall support the IMAP CAPABILITY and AUTHENTICATE extensions for the SASL mechanism.

#### FIA_SASL_EXT.1.4

The TSF shall support the SMTP AUTH extension for the SASL mechanism.

# Appendix D - Implicitly Satisfied Requirements

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this Protection Profile. However, these requirements are not featured explicitly as SFRs and should not be included in the ST. They are not included as standalone SFRs because it would increase the time, cost, and complexity of evaluation. This approach is permitted by [CC] Part 1, **8.2 Dependencies between components**.

This information benefits systems engineering activities which call for inclusion of particular security controls. Evaluation against the Protection Profile provides evidence that these controls are present and have been evaluated.

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this PP-Module. However, these requirements are not featured explicitly as SFRs and should not be included in the ST. They are not included as standalone SFRs because it would increase the time, cost, and complexity of evaluation. This approach is permitted by [CC] Part 1, 8.2 Dependencies between components.

This information benefits systems engineering activities which call for inclusion of particular security controls. Evaluation against the PP-Module provides evidence that these controls are present and have been evaluated.

| Requirement | Rationale for Satisfaction |
| --- | --- |
| FCS_COP.1 - Cryptographic Operation | Several SFRs in this PP-Module (e.g. FPT_AON_EXT.2) have a dependency on FCS_COP.1 because they require the existence of other cryptographic functionality to be satisfied. The Base-PP permits either the TOE or its platform to implement cryptographic functions. If the TOE platform implements these functions, FCS_COP.1 is not claimed but all SFRs that depend on it are implicitly satisfied through the TOE platform's ability to provide the required functionality. |
| FPT_STM.1 - Reliable Time Stamps | FIA_X509_EXT.3 has a dependency on FPT_STM.1 because reliable time is needed to validate whether or not an X.509 certificate is expired. This requirement is implicitly satisfied through the Base-PP assumption that the TOE platform can be assumed to be a reliable time source. |

# Appendix E - Entropy Documentation and Assessment

The TOE does not require any additional supplementary information to describe its entropy sources beyond the requirements outlined in the Base-PPs.

# Appendix F - Acronyms

| Acronym | Meaning |
|---|---|
| AES | Advanced Encryption Standard |
| Base-PP | Base Protection Profile |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |
| CMS | Cryptographic Message Syntax |
| CRL | Certificate Revocation List |
| CSP | Critical Security Parameter |
| DRBG | Deterministic Random Bit Generator |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| IETF | Internet Engineering Task Force |
| IMAP | Internet Message Access Protocol |
| IV | Initialization Vector |
| MAPI | Messaging Application Programming Interface |
| MTA | Mail Transfer Agent |
| NIST | National Institute of Standards and Technology |
| OE | Operational Environment |
| PBKDF | Password-Based Key Derivation Function |
| PDF | Portable Document Format |
| POP | Post Office Protocol |
| PP | Protection Profile |
| PP-Configuration | Protection Profile Configuration |
| PP-Module | Protection Profile Module |
| PRF | Pseudo-Random Function |
| RBG | Random Bit Generator |
| RPC | Remote Procedure Call |
| S/MIME | Secure/Multipurpose Internet Mail Extensions |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| SMTP | Simple Mail Transfer Protocol |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |

# Appendix G - Bibliography

| Identifier | Title |
| --- | --- |
| [CC] | Common Criteria for Information Technology Security Evaluation -<br>• Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.<br>• Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.<br>• Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017. |
| [App PP] | Protection Profile for Application Software, Version 1.3, March 1, 2019 |
| [MS-OXCMAPIHTTP] | Messaging Application Programming Interface (MAPI) Extensions for HTTP |
| [MS-OXCRPC] | Wire Format Protocol |