PP-Module for Email Client

Version: 1.0
2025-06-16
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2025-06-10	Initial release as CC:2022 PP-Module

Contents

1 Introduction

1.1 Overview

The scope of the PP-Module for Email Clients, Version 1.0 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 2.0

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.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes.
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.
Direct Rationale	A type of Protection Profile, PP-Module, or Security Target in which the security problem definition (SPD) elements are mapped directly to the SFRs and possibly to the security objectives for the operational environment. There are no security objectives for the TOE.
Extended Package (EP)	A deprecated document form for collecting SFRs that implement a particular protocol, technology, or functionality. See Functional Packages.
Functional Package (FP)	A document that collects SFRs for a particular protocol, technology, or functionality.
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-PPs.
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.
Target of Evaluation (TOE)	The product under evaluation.
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.

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, which introduces 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] provides 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. Figure 1 shows the TOE's interaction with remote external interfaces that are used to transfer mail between clients. Two separate email clients are shown to illustrate how the TOE can function as both a sender and a receiver using different protocols.

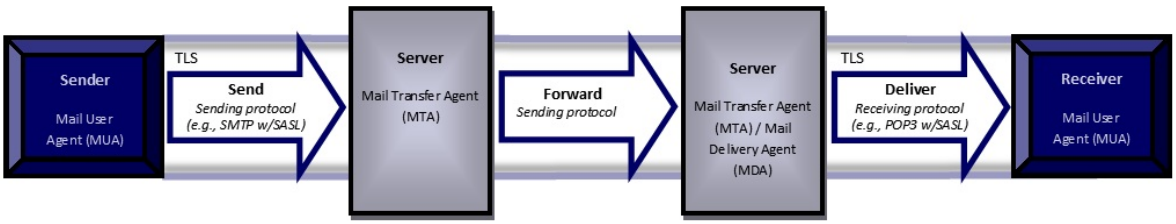


Figure 1: Sending and Delivering Email over TLS

1.4 Use Cases

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

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

Email clients are used for sending, receiving, viewing, accessing, and 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

An ST must claim exact conformance to this PP-Module.

The evaluation methods used for evaluating the TOE are a combination of the workunits defined in [CEM] as well as the Evaluation Activities for ensuring that individual SFRs and SARs have a sufficient level of supporting evidence in the Security Target and guidance documentation and have been sufficiently tested by the laboratory as part of completing ATE_IND.1. Any functional packages this PP claims similarly contain their own Evaluation Activities that are used in this same manner.

CC Conformance Claims

This PP-Module is conformant to Part 2 (extended) and Part 3 (extended) of Common Criteria CC:2022, Revision 1.

PP Claim

This PP-Module does not claim conformance to any Protection Profile.

The following PPs and PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module:

- Protection Profile for Application Software, Version 2.0

Package Claim

This PP-Module is not conformant to any Functional or Assurance Packages.

3 Security Problem Definition

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.

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-partyutilities and tools. This expanded set of capabilities is made possible via the use ofadd-ons. The tight integration between the basic email client code and the newcapabilities that add-ons provide increases the risk that malefactors could injectserious flaws into the email client application, either maliciously by an attacker, oraccidentally by a developer. These flaws enable undesirable behaviors including, but notlimited to, allowing unauthorized access to sensitive information in the email client,unauthorized access to the device's file system, or privilege escalation thatenables unauthorized access to other applications or the operating system.

T.NETWORK_ATTACK (from App PP)
This threat from the above Base PP also applies to the functionality defined in this PP-Module.

T.NETWORK_EAVESDROP (from App PP)
This threat from the above Base PP also applies to the functionality defined in this PP-Module.

T.PHYSICAL_ACCESS (from App PP)
This threat from the above Base PP also applies to the functionality defined in this PP-Module.

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

4.1 Security Objectives for the Operational Environment

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

4.2 Security Objectives Rationale

This section describes how the assumptions and organizational security policies map to operational environment security objectives.

Table 1: Security Objectives Rationale

Assumption or OSP	Security Objectives	Rationale

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 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 Protection Profile for Application Software Security Functional Requirements Direction

5.1.1 Modified SFRs

This PP-Module does not modify any SFRs defined by the App PP.

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.6 Cryptographic Key Destruction

FCS_CKM.6.1

The TSF shall destroy [**assignment**: *list of cryptographic keys (including keying material)*] when [**selection**: *no longer needed*, *[**assignment**: other circumstances for key or keying material destructions.]*]

FCS_CKM.6.2

The TSF shall [**selection**: ***destroy***, ***invoke platform-provided functionality to destroy***] cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [**selection**:

- *for volatile memory, a single direct overwrite of [**selection**: zeroes, a pseudo-random pattern generated by the TOE's DRBG, a pseudo-random pattern generated by the host platform's DRBG]*
- *for non-volatile memory, a [**selection**: single, three or more times] overwrite of key data consisting of [**selection**: a static pattern, a pseudo-random pattern generated by the TOE's DRBG, a pseudo-random pattern generated by the host platform's DRBG]*

] that meets the following: [**selection**: *NIST SP 800-88, no standard*]

Application Note: For the purpose 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 keys and 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 keys and cryptographic critical security parameter to another memory location.

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: This SFR references the selection-based SFRs FCS_COP_EXT.2 and FCS_SMC_EXT.1. If any selection that references these SFRs is chosen, the ST must also claim that selection-based SFR.

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.

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 factors*
  - *credentials stored in platform key storage*

  ]

] to the data encryption and decryption keys using the following methods: [**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 256 bits.

Application Note: This SFR references the selection-based SFRs FCS_CKM_EXT.5, FCS_COP_EXT.2, and FCS_SMC_EXT.1. If any selection that references one of these SFRs is chosen, the ST must also claim that selection-based SFR.

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.

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-256 CBC and [**selection**: *AES-256 GCM*, *no other*] as part of the S/MIME protocol.

Application Note: Advanced Encryption Standard (AES) was added to Cryptographic Message Syntax (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-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: [**selection**:

- *sha384WithRSAEncryption*
- *sha512WithRSAEncryption*
- *ecdsawithsha384*
- *ecdsawithsha512*
- *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.

Additional algorithms supported by RFC 8551 will be reviewed and considered by the TC in a future version of this PP-Module.

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 in accordance with FIA_X509_EXT.1.1 [**selection**: *for each signed and encrypted message sent and 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 and receiving messages. Frequency is configurable in FMT_MOF.1/EmailClient. 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_EXT.3. The import of certificates and certificate chains is not included in this requirement, but is covered in FMT_SMF.1/EmailClient.

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.

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 and outgoing messages, or that the email client automatically encrypt or sign and verify all sent or received messages. This requirement only specifies that the mechanism for digital signature and encryption must be S/MIME.

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.2/SMIME X.509 Certificate Support for Functions (S/MIME)

FIA_X509_EXT.2.1/SMIME

The TSF shall [**selection**: *invoke platform-provided functionality to validate, validate*] X.509v3 certificates in accordance with **in the Functional Package for X.509, version 1.0** to support [*authentication and encryption for S/MIME, installation of code*] using [*[code signing for system software updates, [encryption of email messages, digitally signing of email messages]]*].

Application Note: The intent of this requirement is to require the use of X.509 certificates to determine authentication and integrity for email messages using S/MIME, and to prevent the installation of code in the event of an invalid code signing certificate.

FIA_X509_EXT.2.2/SMIME

For each function indicated in FIA_X509_EXT.2.1**/SMIME**, the TSF shall [**selection**: *invoke the TOE platform to determine, determine*] whether the [*certificate is not accpted*] when valid certificate revocation status information cannot be obtained from a source indicated in FIA_X509_EXT.1.3.

Application Note: The intent of this requirement is to enforce that the TSF presents the installation of code or the signing and encryption of email if the associated code signing or email protection certificate is deemed invalid.

5.2.4 Security Management (FMT)

FMT_SMF.1/EmailClient Specification of Management Functions

FMT_SMF.1.1/EmailClient

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

- *X: Mandatory*
- *O: Optional*

Table 2: Management Functions

Status Markers:
O - Indicates that this function is optional for this role

#	Management Function	Administrator	User
1	Enable or disable downloading embedded objects globally and by [**selection**: *domain, sender, no other method*]	O	O
2	Enable or disable plaintext-only mode globally and by [**selection**: *domain, sender, no other method*]	O	O
3	Enable or disable rendering and execution of attachments globally and by [**selection**: *domain, sender, no other method*]	O	O
4	Enable or 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 and receiving to only use cryptographic algorithms defined in FCS_SMIME_EXT.1	O	O
8	Configure CRL retrieval frequency	O	O
9	Enable or disable support for add-ons	O	O
10	Change password or 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 mail transfer agent (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.

For each management function specified in this SFR, it is optional for a conformant TOE to implement this function because not all conformant email clients may need to implement each function. Additionally, within each function, if the TSF does implement the function, it is optional as to whether a user is authorized to perform that function or if it is restricted to an administrator.

The intent of this requirement is to allow the Administrator to configure the email client with a policy that may not be overridden 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 Functional Package for X.509, version 1.0). 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 Functional Package for X.509, version 1.0), 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.

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_SMF.1/EmailClient.

5.2.6 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

The TSF shall enforce the [**assignment**: *access control SFP(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the TOE.

FTP_ITC.1.2

The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FTP_ITC.1.3

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**assignment**: *additional importation control rules*].

5.3 TOE Security Functional Requirements Rationale

The following rationale provides justification for each SFR for the TOE, showing that the SFRs are suitable to address the specified threats:

Table 3: SFR Rationale

Threat	Addressed by	Rationale
T.FLAWED_ADDON	FDP_NOT_EXT.1	FDP_NOT_EXT.1 mitigates the threat by defining a mechanism for users to determine whether a given email has been signed or encrypted.
	FDP_NOT_EXT.2	FDP_NOT_EXT.2 mitigates the threat by optionally requiring

	(optional)	the TSF to enumerate the uniform resource identifier (URI) of embedded links in emails so that a user can determine the source of the link.
	FDP_REN_EXT.1 (optional)	FDP_REN_EXT.1 mitigates the threat by optionally defining a plaintext-only operational mode that does not allow a user to interact with embedded content in an email message.
	FMT_SMF.1/EmailClient	FMT_SMF_EXT.1/EmailClient mitigates the threat by defining the technology-specific management functions that may exist for email client applications.
	FPT_AON_EXT.1	FPT_AON_EXT.1 mitigates the threat by specifying whether or not the TSF has the ability to load add-ons.
	FPT_AON_EXT.2 (selection-based)	FPT_AON_EXT.2 mitigates the threat by defining a cryptographic method for the TSF to validate the integrity of add-ons if the TOE supports their use.
T.NETWORK_ATTACK (from App PP)	FCS_SMIME_EXT.1	FCS_SMIME_EXT.1 mitigates the threat by defining the TOE's cryptographic implementation of S/MIME to both assert and validate the confidentiality and integrity of secure email messages.
	FDP_NOT_EXT.1	FDP_NOT_EXT.1 mitigates the threat by defining a mechanism for users to determine whether a given email has been signed or encrypted.
	FDP_NOT_EXT.2 (optional)	FDP_NOT_EXT.2 mitigates the threat by optionally requiring the TSF to enumerate the uniform resource identifier (URI) of embedded links in emails so that a user can determine the source of the link.
	FDP_REN_EXT.1 (optional)	FDP_REN_EXT.1 mitigates the threat by optionally defining a plaintext-only operational mode that does not allow a user to interact with embedded content in an email message.
	FDP_SMIME_EXT.1	FDP_SMIME_EXT.1 mitigates the threat by requiring the TSF to use S/MIME to protect email message data in transit.
	FIA_SASL_EXT.1 (selection-based)	FIA_SASL_EXT.1 mitigates the threat by specifying how SASL is implemented in the case where the TOE claims to support it.
	FIA_X509_EXT.2/SMIME	FIA_X509_EXT.2/SMIME mitigates the threat by requiring the TSF to support the use of X.509 certificates for S/MIME.
	FMT_SMF.1/EmailClient	FMT_SMF.1/EmailClient mitigates the threat by defining the technology-specific management functions that may exist for email client applications.
	FTP_ITC.1	FTP_ITC.1 mitigates the threat by specifying the trusted communications the TSF must implement that are specific to email communications.
T.NETWORK_EAVESDROP (from App PP)	FCS_SMIME_EXT.1	FCS_SMIME_EXT.1 mitigates the threat by defining the TOE's cryptographic implementation of S/MIME to both assert and validate the confidentiality and integrity of secure email messages.
	FDP_NOT_EXT.1	FDP_NOT_EXT.1 mitigates the threat by defining a mechanism for users to determine whether a given email has been signed or encrypted.
	FDP_NOT_EXT.2 (optional)	FDP_NOT_EXT.2 mitigates the threat by optionally requiring the TSF to enumerate the uniform resource identifier (URI) of embedded links in emails so that a user can determine the source of the link.
	FDP_REN_EXT.1 (optional)	FDP_REN_EXT.1 mitigates the threat by optionally defining a plaintext-only operational mode that does not allow a user to interact with embedded content in an email message.
	FDP_SMIME_EXT.1	FDP_SMIME_EXT.1 mitigates the threat by requiring the TSF to use S/MIME to protect email message data in transit.
	FIA_SASL_EXT.1 (selection-based)	FIA_SASL_EXT.1 mitigates the threat by specifying how SASL is implemented in the case where the TOE claims to support it.
	FIA_X509_EXT.2/SMIME	FIA_X509_EXT.2/SMIME mitigates the threat by requiring the TSF to support the use of X.509 certificates for S/MIME.
	FMT_SMF.1/EmailClient	FMT_SMF.1/EmailCLient mitigates the threat by defining the technology-specific management functions that may exist for email client applications.
	FTP_ITC.1	FTP_ITC.1 mitigates the threat by specifying the trusted communications the TSF must implement that are specific to email communications.
T.PHYSICAL_ACCESS (from App PP)	FCS_CKM_EXT.3	FCS_CKM_EXT.3 mitigates the threat by defining the mechanism by which the TSF protects stored key data from unauthorized disclosure.
	FCS_CKM_EXT.5 (selection-based)	FCS_CKM_EXT.5 mitigates the threat by optionally defining the mechanism by which the TSF can derive key material using a user-supplied password credential.
	FCS_CKM.6	FCS_CKM.6 mitigates the threat by defining the mechanism by which the TSF securely destroys stored key data.
	FCS_COP_EXT.2 (selection-based)	FCS_COP_EXT.2 mitigates the threat by defining the supported key wrap mechanisms if the TSF uses key wrapping as part of maintaining a key chain.

FCS_IVG_EXT.1 (optional)	FCS_IVG_EXT.1 mitigates the threat by optionally specifying the initialization vectors used for various cryptographic modes if the TOE supports any of these modes.
FCS_KYC_EXT.1	FCS_KYC_EXT.1 mitigates the threat by defining any key chain that the TSF implements to protect a root encryption key.
FCS_NOG_EXT.1 (optional)	FCS_NOG_EXT.1 mitigates the threat by optionally defining the minimum nonce size if the TSF uses any cryptographic algorithms that require the use of nonces.
FCS_SAG_EXT.1 (optional)	FCS_SAG_EXT.1 mitigates the threat by optionally defining the supported methods for salt generation if the TSF uses any cryptographic algorithms that require the use of salts.
FCS_SMC_EXT.1 (selection-based)	FCS_SMC_EXT.1 mitigates the threat by defining the supported key combination mechanisms if the TSF uses key combining as part of maintaining a key chain.
FDP_PST_EXT.1 (optional)	FDP_PST_EXT.1 mitigates the threat by optionally defining the ability of the TOE to operate without persistently storing certain types of data at all.

6 Consistency Rationale

6.1 Protection Profile for Application Software

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.

The only asset for the TOE is the software executable and sensitive data that comprises the TOE. The entire TOE as defined by the combination of the Base-PP and this PP-Module is a single asset. The only difference to the threat model is that the PP-Module introduces the concept of add-ons, which introduces the threat of an add-on being flawed in some way.

6.1.2 Consistency of Security Problem Definition

Listed below are the threats, objectives, and OSPs defined in this PP-Module with rationale for their consistency with the App PP. The PP-Module shares the executable application asset with the App PP but defines an additional threat because the PP-Module defines a specific type of software application with potential exploits that are common to the application type.

Note that the PP-Module is implicitly consistent with any claimed functional packages because the applicable functional packages do not have security problem definitions of their own; per section 2, any claimed functional package is intended to support the O.PROTECTED_COMMS objective in the App PP, which helps mitigate the T.NETWORK_ATTACK and T.NETWORK_EAVESDROP threats in that PP.

Table 4: Consistency of Security Problem Definition (App PP base)

PP-Module Threat, Assumption, OSP	Consistency Rationale
T.FLAWED_ADDON	

6.1.3 Consistency of OE Objectives

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 App PP that are needed to support Email Client functionality. This is considered to be consistent because the functionality provided by the App PP is being used for its intended purpose. The rationale for why this does not conflict with the claims defined by the App PP are as follows:

Table 5: Consistency of Requirements (App PP base)

PP-Module Requirement	Consistency Rationale
Modified SFRs	
This PP-Module does not modify any requirements when the App PP is the base.	
Additional SFRs	
This PP-Module does not add any requirements when the App PP is the base.	
Mandatory SFRs	
FCS_CKM.6	This SFR defines how email messages are formatted when sent and received by the client. It does not impact the Base-PP functionality.
FCS_CKM_EXT.3	This SFR defines how keys and key material are saved by the email client. It does not impact the Base-PP functionality.
FCS_KYC_EXT.1	This SFR defines how email clients maintain key chains. It does not impact the Base-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 Base-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 Base-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 Base-PP functionality.
FIA_X509_EXT.2/SMIME	This SFR defines the format an email client shall use for certificates to perform encryption and authentication. It does not impact the Base-PP functionality.
FMT_SMF.1/EmailClient	
FPT_AON_EXT.1	This SFR defines what types of add-ons an email client may use. It does not impact the Base-PP functionality.
FTP_ITC.1	This SFR defines which channels for an email client must be considered trusted. It does not impact the Base-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.
Objective SFRs	
This PP-Module does not define any Objective requirements.	
Implementation-dependent SFRs	
This PP-Module does not define any Implementation-dependent requirements.	
Selection-based SFRs	
FCS_CKM_EXT.5	This SFR defines restrictions on password composition and key derivation mechanisms. It defines functionality similar to FCS_PBKDF_EXT.1 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.

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 and unpredictable*
- *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. Cipher Block Chaining (CBC), XTS, and Galois Counter Mode (GCM) are allowed for AES encryption of the data. AES-CCM is an allowed mode for Key Wrapping.

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.

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*

]

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.

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.

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_SMF.1/EmailClient 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.

A.2 Objective Requirements

This PP-Module does not define any Objective SFRs.

A.3 Implementation-dependent Requirements

This PP-Module does not define any Implementation-dependent 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 selection in FCS_KYC_EXT.1.1.

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 or 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*]

Application Note: *The ST author assigns any other supported characters; if there are no other supported characters, they should select "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-SHA-384***], with [**assignment**: *positive integer of 4096 or more*] iterations, and output cryptographic key sizes [***256***] bits that meet the following: [***NIST SP 800-56B***].

Application Note: The ST author selects the parameters based on the password-based key derivation function (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 pseudorandom function (PRF) consisting of HMAC with an approved hash function. The ST author must select the hash function and ensure that appropriate claims are made for FCS_COP.1/Hash and FCS_COP.1/KeyedHash in the Base-PP.

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, therefore increasing 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 12 bits of security is added to the password or 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: *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).*

FCS_COP_EXT.2 Key Wrapping

The inclusion of this selection-based component depends upon selection in FCS_CKM_EXT.3.1, FCS_KYC_EXT.1.1.

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**:
- *256 bits (AES)*
- *3072 (RSA)*
- *4096 (RSA)*
- *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 encryption or decryption. In the second selection, the ST author chooses the method used for encryption and decryption:

- Using one of the two AES-based Key Wrap methods specified in NIST SP 800-38F
- Using one of the two KTS-OAEP schemes for RSA as described in NIST SP 800-56B (KTS-OAEP-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. Support for 256-bit AES key sizes will be required for products entering evaluation after Quarter 3, 2015. Based on the methods selected, the last selection should be used to select the appropriate references.

FCS_SMC_EXT.1 Key Combining

The inclusion of this selection-based component depends upon selection in FCS_CKM_EXT.3.1, FCS_KYC_EXT.1.1.

FCS_SMC_EXT.1.1

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

- *exclusive OR (XOR)*
- *SHA-384*
- *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.

B.2 Identification and Authentication (FIA)

FIA_SASL_EXT.1 Simple Authentication and Security Layer (SASL)

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 IMAP, SMTP, or POP is selected in FTP_ITC.1.2.

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.

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 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.

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-Module are listed in this table:

Table 6: Extended Component Definitions

Functional Class	Functional Components
Cryptographic Support (FCS)	FCS_CKM_EXT Cryptographic Key Management FCS_COP_EXT Cryptographic Operation FCS_IVG_EXT Initialization Vector Generation FCS_KYC_EXT Cryptographic Key Chaining FCS_NOG_EXT Cryptographic Nonce Generation FCS_SAG_EXT Cryptographic Salt Generation FCS_SMC_EXT Submask Combining FCS_SMIME_EXT Secure/Multipurpose Internet Mail Extensions (S/MIME)
Identification and Authentication (FIA)	FIA_SASL_EXT Simple Authentication and Security Layer (SASL)
Protection of the TSF (FPT)	FPT_AON_EXT Add-Ons
User Data Protection (FDP)	FDP_NOT_EXT Notifications FDP_PST_EXT Storage of Persistent Information FDP_REN_EXT Rendering of Message Content FDP_SMIME_EXT Use of Secure/Multipurpose Internet Mail Extensions (S/MIME)

C.2 Extended Component Definitions

C.2.1 Cryptographic Support (FCS)

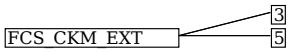
This PP-Module defines the following extended components as part of the FCS class originally defined by CC Part 2:

C.2.1.1 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.

FCS_CKM_EXT.5, Cryptographic Key Derivation (password/passphrase Conditioning), requires the TSF to support password or 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.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*].

Management: FCS_CKM_EXT.5

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

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

Audit: FCS_CKM_EXT.5

There are no audit 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-SHA-384***], with [**assignment**: *positive integer of 4096 or more*] iterations, and output cryptographic key sizes [***256***] bits that meet the following: [***NIST SP 800-56B***].

FCS_CKM_EXT.5.4

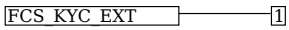
The TSF shall not accept passwords less than [**selection**: *a value settable by the administrator, [**assignment**: minimum password length accepted by the TOE, must be ≥ 1]*] and greater than the maximum password length defined in FCS_CKM_EXT.5.1.

C.2.1.2 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

There are no management functions foreseen.

Audit: FCS_KYC_EXT.1

There are no audit 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 and decryption keys using the following methods: [**assignment**: *key protection method*] while maintaining an effective strength of [**assignment**: *key strength*]

C.2.1.3 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 and receiving to only use specified cryptographic algorithms.

Audit: FCS_SMIME_EXT.1

There are no audit 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-256 CBC and [**selection**: *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

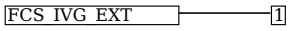
The TSF shall implement mechanisms to retrieve certificates and certificate revocation information in accordance with FIA_X509_EXT.1.1 [**selection**: *for each signed and encrypted message sent and received*, *[**assignment**: frequency]*] as part of the S/MIME protocol.

C.2.1.4 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

There are no management functions foreseen.

Audit: FCS_IVG_EXT.1

There are no audit 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

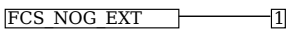
The TSF shall create IVs in the following manner: [**assignment**: *IVs and methods of creation*].

C.2.1.5 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

There are no management functions foreseen.

Audit: FCS_NOG_EXT.1

There are no audit 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

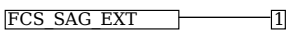
The TSF shall only use unique nonces with a minimum size of 64 bits.

C.2.1.6 FCS_SAG_EXT Cryptographic Salt 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

There are no management functions foreseen.

Audit: FCS_SAG_EXT.1

There are no audit 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

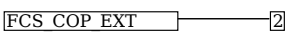
The TSF shall only use salts that are generated by a [**assignment**: *trusted deterministic random bit generator*].

C.2.1.7 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

There are no management functions foreseen.

Audit: FCS_COP_EXT.2

There are no audit 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*

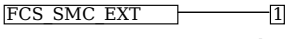
] in accordance with a specified cryptographic algorithm [**assignment**: *cryptographic algorithm*] and the cryptographic key size [**assignment**: *cryptographic key size*] that meet the following: [**assignment**: *list of standards*]

C.2.1.8 FCS_SMC_EXT Submask Combining

Family Behavior

Components in this family define requirements for the process of 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

There are no management functions foreseen.

Audit: FCS_SMC_EXT.1

There are no audit 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-384*
- *SHA-512*

] to generate another key.

C.2.2 Identification and Authentication (FIA)

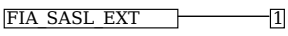
This PP-Module defines the following extended components as part of the FIA class originally defined by CC Part 2:

C.2.2.1 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

There are no management functions foreseen.

Audit: FIA_SASL_EXT.1

There are no audit events foreseen.

FIA_SASL_EXT.1 Simple Authentication and Security Layer (SASL)

Hierarchical to:	No other components.
Dependencies to:	No dependencies.

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.

C.2.3 Protection of the TSF (FPT)

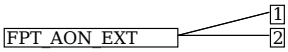
This PP-Module defines the following extended components as part of the FPT class originally defined by CC Part 2:

C.2.3.1 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.

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.1

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

- Enable or disable support for add-ons.

Audit: FPT_AON_EXT.1

There are no audit 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*].

Management: FPT_AON_EXT.2

There are no management functions foreseen.

Audit: FPT_AON_EXT.2

There are no audit 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.

C.2.4 User Data Protection (FDP)

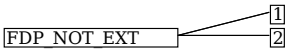
This PP-Module defines the following extended components as part of the FDP class originally defined by CC Part 2:

C.2.4.1 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.

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.1

There are no management functions foreseen.

Audit: FDP_NOT_EXT.1

There are no audit 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.

Management: FDP_NOT_EXT.2

There are no management functions foreseen.

Audit: FDP_NOT_EXT.2

There are no audit 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.

C.2.4.2 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

There are no management functions foreseen.

Audit: FDP_SMIME_EXT.1

There are no audit 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

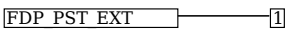
The TSF shall use S/MIME to sign, verify, encrypt, and decrypt mail.

C.2.4.3 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

There are no management functions foreseen.

Audit: FDP_PST_EXT.1

There are no audit events foreseen.

FDP_PST_EXT.1 Storage of Persistent Information

Hierarchical to: No other components.

Dependencies to: No dependencies.

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*].

C.2.4.4 FDP_REN_EXT Rendering of Message Content

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 or disable plaintext-only mode.

Audit: FDP_REN_EXT.1

There are no audit 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*].

Appendix D - Implicitly Satisfied Requirements

This appendix lists requirements that should be considered satisfied by products successfully evaluated against this PP-Module. 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, 8.3 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.2/SMIME 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-PP.

Appendix F - Acronyms

Table 7: 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
cPP	Collaborative Protection Profile
CRL	Certificate Revocation List
CSP	Critical Security Parameter
EP	Extended Package
FP	Functional Package
GCM	Galois-Counter Mode
IMAP	Internet Message Access Protocol
MAPI	Messaging Application Programming Interface
MTA	Mail Transfer Agent
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	Pseudorandom Function
RPC	Remote Procedure Call
S/MIME	Secure/Multipurpose Internet Mail Extensions
SAR	Security Assurance Requirement
SASL	Simple Authentication and Security Layer
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
URI	Uniform Resource Identifier

Appendix G - Bibliography

Table 8: Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022. • Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022. • Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022. • Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022. • Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022, Revision 1, November 2022.
[CEM]	Common Methodology for Information Technology Security Evaluation - • Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.
[App PP]	Protection Profile for Application Software, Version 2.0, June 16, 2025
[MS-OXCMAPIHTTP]	Messaging Application Programming Interface (MAPI) Extensions for HTTP
[MS-OXCRPC]	Wire Format Protocol