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

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	Common Evaluation Methodology for Information Technology Security Evaluation.

(CEM)	
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.

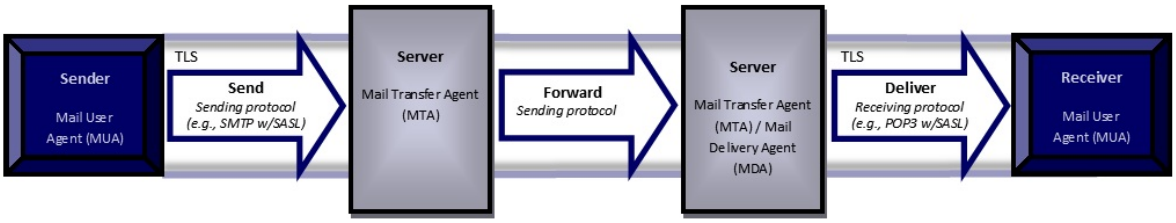

Figure 1: Sending and Delivering Email over TLS

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 Application Software PP Security Functional Requirements Direction

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

5.1.1 Modified SFRs

This PP-Module does not modify any SFRs defined by the Application Software 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_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.

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.1 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.

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.5.1*,
  - *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.

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.

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/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.

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.3 X509 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.

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	O	O

	FCS_SMIME_EXT.1		
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.

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: FPT_AON_EXT.2 depends upon the selection made here. 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. Cryptographic verification (i.e., trust) of add-ons is tested in FPT_AON_EXT.2.1.

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).

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. FDP_MOF_EXT.1 supports the objective by defining the technology-specific management functions that may exist for email client applications. 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. 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_CKM_EXT.5 (optional), FCS_IVG_EXT.1 (optional), FCS_NOG_EXT.1 (optional), FCS_SAG_EXT.1 (optional), FDP_PST_EXT.1 (optional), FCS_COP_EXT.2 (selection-based), FCS_SMC_EXT.1 (selection-based)	FCS_CKM_EXT.3 supports the objective by defining the mechanism by which the TSF protects sotred key data from unauthorized disclosure. FCS_CKM_EXT.4 supports the objective by defining the mechanism by which the TSF securely destroys stored key data. FCS_KYC_EXT.1 supports the objective by defining any key chain that the TSF implements to protect a root encryption key. 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. 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. 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. 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. 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. 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. 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_TLS_EXT.1 (modified from TLS Package), FCS_TLSC_EXT.1 (from TLS package),	FCS_TLS_EXT.1 supports the objective by requiring the

	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)	TSF to implement TLS as a client. FCS_TLSC_EXT.1 supports the objective by requiring the TSF to implement TLS client functionality in a specified manner. FIA_X509_EXT.1 supports the objective by requiring the TSF to implement or invoke an X.509 certificate validation service. 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. FTP_DIT_EXT.1 supports the objective by specifying the trusted communications channels used by the TOE to protect data in transit. 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. FDP_SMIME_EXT.1 supports the objective by requiring the TSF to use S/MIME to protect email message data in transit. FIA_X509_EXT.3 supports the objective by requiring the TSF to support the use of X.509 certificates for S/MIME. FTP_ITC_EXT.1 supports the objective by specifying the trusted communications the TSF must implement that are specific to email communications. 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. 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 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.

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

PP-Module Requirement	Consistency Rationale
Modified SFRs	
This PP-Module does not modify any requirements when the Application Software PP is the base.	
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_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_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 how clients display URIs in embedded links 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_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_CKM_EXT.5 Cryptographic Key Derivation (Password/Passphrase Conditioning)

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: 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_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.

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.1,*
- *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_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.

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_COP_EXT.2 Key Wrapping

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: 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).

FCS_SMC_EXT.1 Key Combining

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 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: 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.

B.3 Protection of the TSF (FPT)

FPT_AON_EXT.2 Trusted Installation and Update for 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.

Appendix C - 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
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 D - 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 E - 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 F - Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017. • Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017. • 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