National Cyber Security Centre
a part of GCHQ

CPA SECURITY CHARACTERISTIC

CPA-SC DESKTOP EMAIL ENCRYPTION 1.1 DOC

Version 1.1

Document History

Version	Date	Description
0.0	6th March 2012	Preparation for industry review
1.0	17th April 2012	Updates following industry review
1.1	25th October 2018	Amended to reflect formation of NCSC

This Security Characteristic is derived from the following files

File Name	Version
Desktop Email Encryption – v1.0.cxl	1.0
Common Email Encryption – v1.4.cxl	1.4
Common Libraries – v1.6.cxl	1.6
Crypt Libraries – v1.4.cxl	1.4
Hardware Libraries – v1.3.cxl	1.3

Soft copy location: NCSC-1844117881- 471

This document is authorised by:

Deputy Technical Director (Assurance), NCSC

This document is issued by NCSC

For queries about this document please contact:

CPA Administration Team
NCSC, A2i,
Hubble Road
Cheltenham
Gloucestershire
GL51 0EX
United Kingdom

Tel: +44 (0)1242 221 491
Email: cpa@ncsc.gov.uk

The CPA Authority may review, amend, update, replace or issue new Scheme Documents as may be required from time to time.

CONTENTS

REFERENCES

[a] The Process for Performing Foundation Grade CPA Evaluations, v1.3, NCSC [August 2011]

[b] NIST SP 800-90 – Recommendation for Random Number Generation Using Deterministic Random Bit Generators [2007]

[c] RFC 4880 – OpenPGP Message Format [November 2007]

[d] RFC 5751 – Secure Multipurpose Internet Mail Extensions, version 3.2 [January 2010]

[e] RFC 5321 – Simple Mail Transfer Protocol [October 2008]

[f] RFC 1939 – Post Office Protocol – version 3 [May 1996]

[g] RFC 3501 – Internet Messaging Access Protocol version 4, revision 1 [March 2003]

[h] RFC 3207 - SMTP Service Extension for Secure SMTP over TLS [February 2002]

[i] RFC 2595 - Using TLS with IMAP, POP3 & ACAP [June 1999] *(see also RFC 4616)*

[j] RFC 4616 - The PLAIN Simple Authentication and Security Layer (SASL) [August 2006] *(updates RFC 2595)*

[k] RFC 5408 – IBE Architecture and Supporting Data Structures [January 2009]

[l] HMG IA Standard No. 5 - Secure Sanitisation, issue 4.0 [April 2011]

[m] HMG IA Standard No. 7 - Authentication of Internal Users of ICT Systems Handling Government Information, issue 4.0 [October 2010]

[n] NCSC Good Practice Guide No. 35 – Protecting an Internal ICT Network, issue 2.0 [August 2011]

[o] CPA Security Characteristic - Network Authentication: Protected Endpoint, version 0.4 [October 2011]

[p] CPA Security Characteristic – Gateway Email Encryption, version 0.9f [October 2011]

I. OVERVIEW

1. This document is a CPA Security Characteristic – it describes requirements for a particular type of assured product for evaluation and certification under NCSC's Commercial Product Assurance (CPA) scheme.

A. Product Aims

2. Email encryption products are intended to protect the confidentiality and integrity of emails in addition to providing the recipient with authentication of the sender.

3. "Desktop Email Encryption" in the context of this Security Characteristic refers to an encrypted email client deployed within a corporate desktop environment, which applies and removes cryptographically protection for email messages sent to and received by a user endpoint.

4. This protection is:

a) Applied to an email through encryption and digital signing.

b) Removed from an email through decryption and digital signature validation. The recipient is made aware if an email fails authentication during removal of protection.

B. Typical Use Case(s)

5. A desktop email encryption product extends the usage of a general email client as follows:

a) Allows a user endpoint to optionally encrypt and digitally sign an email before sending it to its intended destination(s).

b) Transparently decrypts and validates digital signatures for incoming cryptographically-protected emails that are destined for the local user endpoint, warning the user of any failures/anomalies encountered during this process.

6. The encrypted emails are sent to and received from other user endpoints that may be located in the local trusted domain and remote domains. Remote user endpoints may require transit of the encrypted email through a remote encrypted email gateway.

a) **Outbound protection** – the product encrypts the email for each recipient and signs the email to ensure integrity of delivery. The email is then sent to the destination address.

b) **Inbound email receipt** – the product attempts to cryptographically validate the signature and decrypt any protected emails it receives for the local recipient. Any warnings/errors encountered are displayed to the user before making the raw email content accessible (if decryption succeeded).

C. Expected Operating Environment

7. A desktop email encryption product comprises software that a user endpoint operates on a desktop machine or corporate-managed laptop as part of an enterprise email deployment, within a security domain. It is expected to present the same user interface as a standard email client, but additionally allow the user endpoint to specify cryptographic protection for outgoing emails and warn the user of any problems encountered when removing cryptographic protection from incoming emails.

8. In the envisaged deployment architecture (see Figure 1), the desktop email encryption software is expected to process cryptographic protection for emails sent to and received from other user endpoints located both within the local security domain and also within external security domains via less trusted networks (such as the Internet).

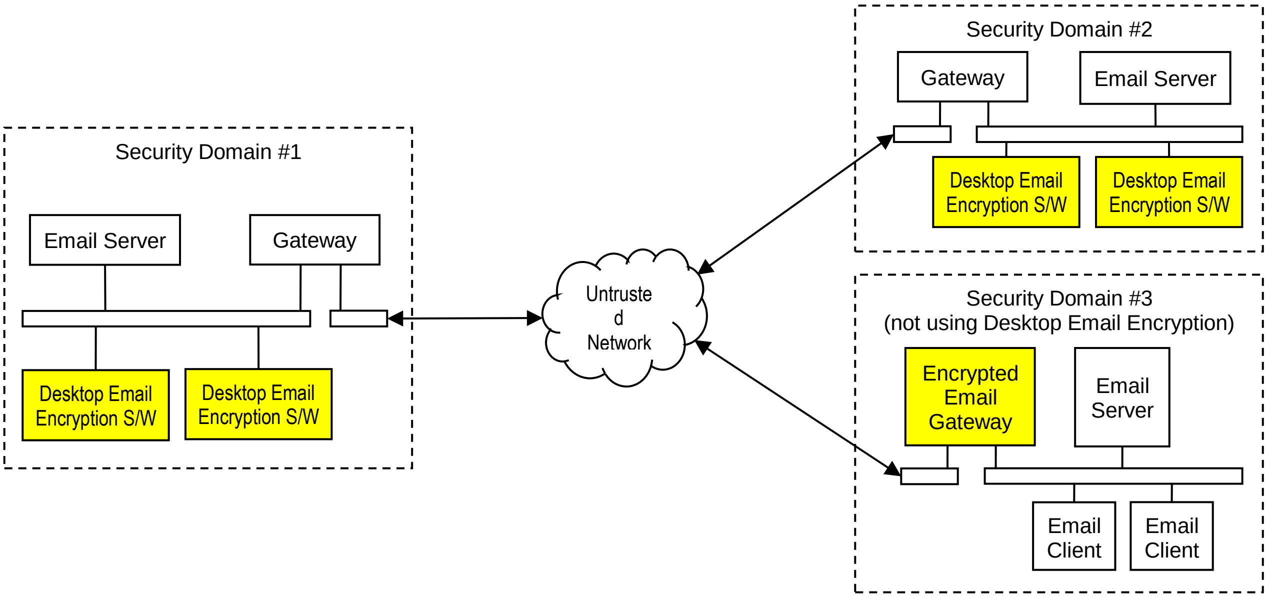

Figure 1: Expected Operating Environment for Desktop Email Encryption

9. In Figure 1, in order for the security domains to interact with each other, each domain needs to be able to access/retrieve public keys of remote endpoints for encryption and digital signature validation. The authenticity of these public keys is protected by a key management solution, in which one or more trusted entities cryptographically bind the key with the endpoint’s identity, allowing the product to digitally verify the binding. Other details about the key management solution are beyond the scope of this document.

D. Compatibility

10. A desktop email encryption product conforming to this Security Characteristic is expected to comprise one or more software modules, deployed on a general purpose desktop client platform (such as a Windows workstation, Unix workstation, etc) within a corporate domain environment. Other than the product operating correctly for the desktop client platform's O/S, this Security Characteristic places no specific requirement on the operating system.

11. This Security Characteristic therefore does not place any specific hardware or requirements upon the product beyond its normal technical requirements. For example some products may have specific CPU or memory requirements in order to function effectively. This Security Characteristic does not define minimum hardware requirements.

E. Interoperability

12. An encrypted email gateway product must support the following algorithms when applying and removing cryptographic protection:

Algorithm Type	Approved Algorithm/s
Symmetric Encryption	AES-128-CBC, and/or AES-128-CFB
Key Wrap (Key Encryption)	AES-128 Key Wrap
Session Key Agreement	ECDH-256, and/or DH-based 1536/192
Hash Function	SHA-256
Digital Signing	ECDSA-256, and/or DSA 1536/192

13. The product is likely, but not necessarily required, to interoperate with one or more of the following:

a) Widely-used open-standard encrypted email protocols, such as OpenPGP and S/MIME (references [c] and [d])

b) General email messaging protocols, such as SMTP, POP3 and IMAP (references [e], [f] and [g])

c) PKI management nodes, such as certification authorities, public key servers, public key repositories, etc

d) Email-aware anti-malware and content filtering products

e) The user endpoint's desktop profile – for instance, to access the user's long term private key data

F. Variants

14. This Security Characteristic has the following variants:

a) Encrypted Email Type:

i) OpenPGP - Email cryptographically protected using OpenPGP security mechanisms (Reference [c])

ii) S/MIME - Email cryptographically protected using S/MIME security mechanisms (Reference [d])

iii) Bespoke - Email cryptographically protected using a bespoke security mechanism (i.e. not OpenPGP or S/MIME)

G. High Level Functional Components

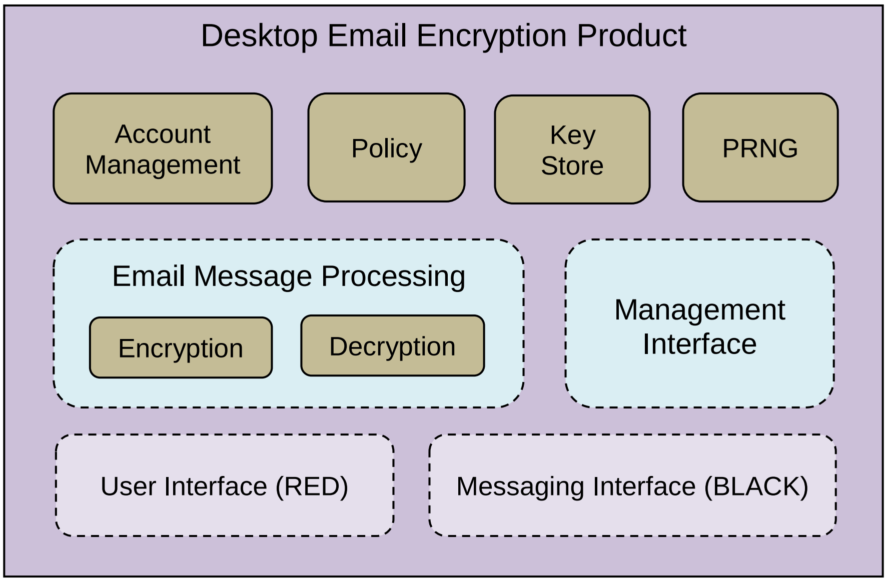

Figure 2: Desktop Email Encryption Product Components

15. The functionality of the desktop email encryption product can be broken down into the key components shown in Figure 2, described as follows:

- **Account Management**

Handles the management of email accounts in terms of identities and associated keys.

- **Policy**

A set of rules, enforced by the desktop email encryption product, which determine the cryptography applied to incoming and outgoing email messages. It is expected to incorporate default rules that cover requirements specific to the local security domain in which the desktop email encryption product is deployed.

- **Key Store**

A local storage mechanism for the local user endpoint’s private long term keys (and any pre-distributed symmetric keys). It will also need to have access to public keys of other user endpoints within the secure email deployment (covering the local domain and any remote trusted domains).

- **PRNG**

The Pseudo-Random Number Generator generates random data primarily for symmetric content encryption keys and IVs. It may also generate ephemeral key-agreement keys and random components required in digital signatures.

- **Email Message Processing**

Handles the encoding and decoding of email messages including the application of cryptographic mechanisms such as encryption, decryption, digital signing/verification. Options for the processing are expected to be constrained according to policy.

- **User Interface (RED)**

This interface handles the presentation of non-encrypted email data either during the editing of a message to be sent or the viewing of a received message.

- **Messaging Interface (BLACK)**

This interface handles the transfer of email data, which may or may not be encrypted, between the local network interface agent.

- **Management Interface**

This logical interface enables the management of product (policy, accounts, keys, etc). Only the system administrator/s for the security domain will be able to access this interface.

H. Future Enhancements

16. NCSC welcomes feedback and suggestions on possible enhancements to this Security Characteristic.

17. At the time of writing, there exist Identity Based Encryption (IBE) standards (e.g. RFC 5408, reference[k]) and IBE email products. Future revisions of this document may additionally cover IBE implementations of gateway email encryption.

18. Other future enhancements to this Security Characteristic may include the following:

a) Integration with desktop encryption software (e.g. general PGP encryption software to integrate with an OpenPGP secure email deployment).

b) Use of Trusted Platform Module (TPM) hardware devices to provide key management for the encrypted email deployment.

II. SECURITY CHARACTERISTIC FORMAT

19. All CPA Security Characteristics contain a list of mitigations which are split into three requirement categories: development, verification and deployment requirements. Within each of these sets the mitigations can be grouped based on areas of the product (as illustrated in the High Level Functional Component Diagram above), such as bulk encryption or authentication, or they may be overarching requirements which apply to the whole product. Reference [a] describes how evaluation teams should interpret Security Characteristics.

20. The three types of mitigations are denominated as follows:

- **DEV** – These are mitigations that are included by the developer during the design or implementation of the product. These are validated via a review of the product's design or implementation during a CPA evaluation.

- **VER** – Verification mitigations are specific mitigations that the evaluator must test during the assessment of the product.

- **DEP** – Deployment mitigations are points that must be considered by users or administrators during the deployment of the product. These mitigations are incorporated into the security procedures for the product.

21. Each mitigation includes informational text in italics, describing the threat that it is expected to mitigate. It also lists at least one specific mitigation, which describes what must actually be done to achieve that requirement. In some cases there is additional explanatory text which expands upon these requirements.

22. In the requirements listed below, the following terminology can be used:

- 'Must', 'Mandatory' and "Required" are used to express a mitigation that is essential. All mitigations and detailed mitigations are mandatory unless there is an explicit caveat, such as 'if supported by the product'.

- 'Should' and 'Strongly Recommended' are used whenever a requirement is highly desirable, but is not essential. These are likely to become mandatory in future iterations of the Security Characteristic.

- 'Could' and 'Recommended' are used to express a non-mandatory requirement that may enhance security or functionality.

23. For example:

DEV.M1: [A mitigation]

*This mitigation is required to counter [**a threat**]*

At Foundation the product must [**do something**].

This can be achieved by [**explanatory comment**].

III. REQUIREMENTS

A. Design Mitigations

DEV.M41: Crash reporting

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product is required to ensure crashes are logged

Where it is possible that sensitive data may end up in the crash data, this must be handled as red data and must only be available to an administrator. Crash data from both the product and the underlying operating system must be considered.

DEV.M42: Heap hardening

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product is required to use the memory management provided by the operating system. Products should not implement their own heap

DEV.M43: Stack protection

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product is required to be compiled with support for stack protection in all libraries, where the tool chain supports it

If more recent versions of the tool chain support it for the target platform then they should be used in preference to a legacy tool chain.

DEV.M46: User least privilege

This mitigation is required to counter taking advantage of existing user privilege

At Foundation Grade the product is required to operate correctly from a standard account without elevated privileges

DEV.M159: Update product

This mitigation is required to counter exploitation of a software logic error
This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product should support the use of software updates

DEV.M267: Provide an automated configuration tool to enforce required settings

This mitigation is required to counter exploitation of an accidental misconfiguration

At Foundation Grade the product is required to be provided with a configuration tool, or other method, for an administrator to initially set it up into a suitable configuration

If the product requires more than 12 options to be changed or set by an administrator to comply with these Security Characteristics, the developer must supply a tool or policy template which helps the administrator to achieve this in fewer steps

DEV.M321: Data Execution Protection

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product is required to support Data Execution Protection (DEP) when enabled on its hosting platform and must not opt out of DEP

If the product is to be specifically deployed on a platform that does not support either Software DEP or Hardware-enforced DEP, there is no requirement for DEP compatibility.

DEV.M340: Address Space Layout Randomisation

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the product is required to be compiled with full support for ASLR, including all libraries used

ASLR may be disabled for specific aspects of the product, provided there is justification of why this is required.

DEV.M353: Ensure product security configuration can only be altered by an authenticated system administrator

This mitigation is required to counter unauthorised alteration of product's configuration

At Foundation Grade the product is required to ensure that only administrators are able to change the product's security enforcing settings

The only security enforcing setting a user should be able to change is their passphrase.

DEV.M355: Secure software delivery

This mitigation is required to counter installation of malware on host

This mitigation is required to counter installing compromised software using the update process

At Foundation Grade the product should be distributed via a cryptographically protected mechanism, such that the authenticity of software can be ensured.

Initial code for the product, and any subsequent updates, must be distributed in such a way that tampering is cryptographically detectable. The recipient of the software must be able to ensure the identity of the originator (i.e. vendor).

DEV.1 - Design >> Decryption

DEV.1.M66: Ephemeral keys protected from high risk processes

This mitigation is required to counter compromised device exfiltrating keys

At Foundation Grade the product is required to use operating system mechanisms (process separation, etc) to protect ephemeral secrets

DEV.1.M349: Sanitise temporary variables

This mitigation is required to counter reading remnant volatile memory

At Foundation Grade the product is required to sanitise temporary variables containing sensitive information as soon as no longer required

A secure erase must consist of at least one complete overwrite with a fixed or random pattern and subsequent verification.

DEV.1.M589: Warn user about errors when removing cryptographic protection from emails

This mitigation is required to counter MITM redirecting message to a different destination

This mitigation is required to counter MITM changing recipient key identifier to one denoting a different recipient

This mitigation is required to counter MITM corrupting ciphertext

This mitigation is required to counter MITM corrupting decryption parameters

This mitigation is required to counter MITM changing recipient key identifier to one that has expired or been revoked

At Foundation Grade the product is required to alert user endpoint about errors encountered when decrypting and/or digitally verifying an incoming encrypted email message

Additionally, the product must allow the user to attempt to decrypt and/or verify the message using known cryptographic keys for the sender and/or recipient (sender public keys to be identified by the sender's email address). If the decryption succeeds, the user must be notified that the message was potentially tampered with. Otherwise the user must be notified of the decryption failure.

DEV.1.M590: Warn user about public key mismatches

This mitigation is required to counter MITM redirecting message to a different destination

This mitigation is required to counter MITM changing recipient key identifier to one denoting a different recipient

At Foundation Grade the product is required to alert user endpoint if public key data (or identifier) in the message's cryptographic headers conflicts with known public key data for the recipient or message sender.

Note: The encrypted email client software is expected to attempt to decrypt the message using known key data for the given sender and recipient as identified by the email addresses. If the decryption succeeds, the recipient is notified that the message was potentially tampered with. Otherwise the recipient is notified of the decryption failure.

DEV.1.M599: Warn if expired keys are required to decrypt and/or verify data

This mitigation is required to counter key/passphrase being used enough times to significantly increase the chances of a brute-force attack succeeding

This mitigation is required to counter key/passphrase validity periods not enforced on new application of key/passphrase

At Foundation Grade the product is required to generate a warning to indicate that the protected content (a) may have not been adequately encrypted and/or (b) may not be authentic, indicating when the affected key(s) expired

The warning details shall be displayed to the user endpoint on opening the affected email.

DEV.1.M603: Warn user if non-approved cryptographic algorithm is encountered

This mitigation is required to counter exploitation of weak cryptographic algorithm

At Foundation Grade the product is required to generate a warning to indicate that the protected content may (a) have not been adequately encrypted and/or (b) not be authentic, giving the reason(s) why (e.g. use of unknown encryption algorithm)

The warning details shall be displayed to the user endpoint on opening the affected email.

DEV.2 - Design >> Email Message Processing

DEV.2.M58: (OpenPGP Protocol, SMIME Protocol ONLY) RFC compliant implementation

This mitigation is required to counter exploitation of vulnerabilities in the key exchange or digital signature protocol

At Foundation Grade the product is required to implement the latest RFC(s) for the implemented encrypted email mechanisms

For instance, use RFC 4880 for OpenPGP encrypted email, RFC 5751 for S/MIME, etc.

DEV.2.M567: Extend BCC Privacy to decryption details in encrypted email messages

This mitigation is required to counter BCC recipient being visible to other recipients due to details in decryption headers

At Foundation Grade the product is required to ensure a BCC recipient is not identifiable from any of the fields (encrypted or unencrypted) in an encrypted email message

Further, the decryption details for a given BCC recipient in an email addressed to multiple recipients should only be present in the copy of that email that is sent to that specific BCC recipient.

DEV.2.M569: (Bespoke Protocol ONLY) Bespoke Protocol implemented according to developer's Functional Specification

This mitigation is required to counter exploitation of vulnerabilities in the bespoke key exchange or digital signature protocol

This mitigation is required to counter exploitation of bespoke protocol vulnerability

At Foundation Grade the product is required to implement the bespoke email protection protocol in accordance to the developer's functional specification

The developer must have a functional specification for the protocol used to cryptographically protect the emails as they are encrypted and signed, and must provide evidence to the evaluator that the product accurately implements this specification.

DEV.2.M574: Verification of certificates immediately prior to use

This mitigation is required to counter replacement of valid recipient certificate with one associated with a compromised key

This mitigation is required to counter a private signing key of a trusted key management entity becoming compromised

At Foundation Grade the product is required to check the authenticity of a user encryption/signing key, before using it, through verification of the associated certificate's digital signature back to a trusted key management entity (including revocation checks)

Approved digital signing algorithms are listed in the Interoperability section.

DEV.2.M577: (OpenPGP Protocol ONLY) Strict OpenPGP header parsing operation and rejection of invalid fields

This mitigation is required to counter exploitation of non-standard header fields in encrypted email

At Foundation Grade the product is required to reject any non-standard OpenPGP headers

Such as marking a header field for experimental use

DEV.2.M581: (SMIME Protocol ONLY) Restrict ASN.1 encoding permutations

This mitigation is required to counter use of non-DER ASN.1 encoding options

This mitigation is required to counter use of excessively large primitive values in the ASN.1 encoding

At Foundation Grade the product is required to reject encodings that are not conformant to ASN.1 Distinguished Encoding Rules (DER)

DEV.2.M583: Use of constrained encoding format

This mitigation is required to counter modification of unencrypted headers to introduce malware

At Foundation Grade the product is required to restrict allowable encoding permutations for unencrypted headers (e.g. use DER for ASN.1, prohibit deprecated encoding options where possible, etc)

DEV.2.M585: (OpenPGP Protocol ONLY) An implementation will only generate V4 certificate packets

This mitigation is required to counter exploitation of older format of packet header in email

This mitigation is required to counter exploitation of a weak V3 certificate packet

At Foundation Grade the product is required to provide backward compatibility when decoding old V3 signature packets, but always use V4 for generating new signature packets

DEV.2.M591: Warn about use of expired/revoked sender public key identifier in received email

This mitigation is required to counter MITM changing recipient key identifier to one that has expired or been revoked

At Foundation Grade the product is required to warn user endpoint before attempting to decrypt or digitally verify an incoming encrypted email using expired or revoked key data

This is to assist identification of attack sources.

DEV.2.M597: Do not add email headers that could assist an attacker

This mitigation is required to counter exploitation of encrypted email client model/software details reported in email headers

At Foundation Grade the product is required to provide administrator configuration options to disable the inclusion of the encrypted email client's model and software identifier in outgoing email headers

This is to help protect against an attacker exploiting a known weakness in a given version of the client software

DEV.3 - Design >> Encryption

DEV.3.M66: Ephemeral keys protected from high risk processes

This mitigation is required to counter compromised device exfiltrating keys

At Foundation Grade the product is required to use operating system mechanisms (process separation, etc) to protect ephemeral secrets

DEV.3.M349: Sanitise temporary variables

This mitigation is required to counter reading remnant volatile memory

At Foundation Grade the product is required to sanitise temporary variables containing sensitive information as soon as no longer required

A secure erase must consist of at least one complete overwrite with a fixed or random pattern and subsequent verification.

DEV.3.M571: Ensure nonce mechanisms do not repeat

This mitigation is required to counter exploitation of repeated protective number-used-once ('nonce')

At Foundation Grade the product is required to ensure that a given number-once-used ('nonce') value will only occur once during the operational lifetime of the key it is protecting

DEV.3.M601: Enforce cryptographic key lifetimes

This mitigation is required to counter key/passphrase being used enough times to significantly increase the chances of a brute-force attack succeeding

At Foundation Grade the product is required to prevent a private key from being used to apply cryptographic protection when not within its validity period

Note: An expired key may be used to *remove* previously applied cryptographic protection (i.e. it could be used to decrypt/verify a previously encrypted/signed message, but not to produce a new encrypted/signed message).

DEV.3.M604: Only use approved algorithms to cryptographically protect outgoing emails

This mitigation is required to counter exploitation of weak cryptographic algorithm

At Foundation Grade the product is required to only use approved algorithms to encrypt and digitally sign outgoing emails (approved algorithms are listed in the Interoperability section)

DEV.4 - Design >> Key Store

DEV.4.M349: Sanitise temporary variables

This mitigation is required to counter reading remnant volatile memory

At Foundation Grade the product is required to sanitise temporary variables containing sensitive information as soon as no longer required

A secure erase must consist of at least one complete overwrite with a fixed or random pattern and subsequent verification.

DEV.4.M576: Retain expired/revoked keys for message decryption & recovery

This mitigation is required to counter loss of asymmetric key required to decrypt, e.g. due to expiry/revocation

At Foundation Grade the product is required to retain expired/revoked keys, but ensure that their usage is restricted to just message decryption and digital signature verification

DEV.4.M595: Prevent export of non-encrypted private keys

This mitigation is required to counter export of secrets through an available API

At Foundation Grade the product is required to prevent the export of long term secrets, such as private keys, through any available API, unless authenticated as a privileged user

It is recommended that the product encrypts long term secrets before exporting them (using algorithms given in the Interoperability section).

DEV.4.M596: Long term keys protected from high risk processes

This mitigation is required to counter compromised device exfiltrating keys

This mitigation is required to counter exploitation of unintended information disclosure to leak keys/secrets

At Foundation Grade the product is required to use O/S mechanisms, such as user privileges and the O/S certificate store (or other protected certificate store) to ensure that unencrypted private keys cannot be retrieved by a standard user

DEV.5 - Design >> Messaging Interface

DEV.5.M593: Reject incoming emails not intended for local user endpoint

This mitigation is required to counter interception of non-encrypted email by eavesdropper or user endpoint other than intended recipients

At Foundation Grade the product is required to reject incoming emails that are not intended for the user endpoint currently operating the desktop email encryption product

DEV.6 - Design >> PRNG

DEV.6.M66: Ephemeral keys protected from high risk processes

This mitigation is required to counter compromised device exfiltrating keys

At Foundation Grade the product is required to use operating system mechanisms (process separation, etc) to protect ephemeral secrets

DEV.6.M138: State the Security Strength required for random numbers

This mitigation is required to counter predictable key generation due to a weak entropy source

This mitigation is required to counter the prediction of randomly generated values due to a weak entropy source

This mitigation is required to counter prediction of randomly generated values due to a weak PRNG

At Foundation Grade the product is required to employ an entropy source of sufficient Security Strength for all random number generation required in the operation of the product

The developer must state the Security Strength required of their entropy source based on analysis of all random numbers used in the product. At this grade, the Security Strength is likely to be 128 bits for products that do not use elliptic curve cryptography. For elliptic curve-based asymmetric mechanisms it is likely to be 256 bits, and for finite field based asymmetric mechanisms it is likely to be 192 bits.

DEV.6.M140: Smooth output of entropy source with approved PRNG

This mitigation is required to counter predictable key generation due to a weak entropy source

This mitigation is required to counter the prediction of randomly generated values due to a weak entropy source

This mitigation is required to counter prediction of randomly generated values due to a weak PRNG

At Foundation Grade the product is required to employ a PRNG of sufficient Security Strength for all random number generation required in the operation of the product

For more details on a suitable PRNG, please see the Process for Performing Foundation Grade Evaluations.

DEV.6.M141: Reseed PRNG as required

This mitigation is required to counter the prediction of randomly generated values due to repeating PRNG output

At Foundation Grade the product is required to follow an approved reseeding methodology

DEV.6.M290: Employ an approved entropy source

This mitigation is required to counter predictable key generation due to a weak entropy source

This mitigation is required to counter the prediction of randomly generated values due to a weak entropy source

This mitigation is required to counter prediction of randomly generated values due to a weak PRNG

At Foundation Grade the product is required to generate random bits using an entropy source whose entropy generation capability is understood

The developer must provide a detailed description of the entropy source used, giving evidence that it can generate sufficient entropy for use in the device, including an estimate of entropy per bit.

If a hardware noise source is used, then the manufacturer's name, the part numbers and details of how this source is integrated into the product must be supplied. If a software entropy source is employed, the API calls used must be provided. Where appropriate, details must be given of how the output of multiple entropy sources are combined.

DEV.6.M142: Perform statistical testing of generated entropy prior to smoothing

This mitigation is required to counter the prediction of randomly generated values due to a weak entropy source

This mitigation is required to counter prediction of randomly generated values due to a weak PRNG

At Foundation Grade the product is required to employ a PRNG of sufficient Security Strength for all random number generation required in the operation of the product

For more details on a suitable PRNG, please see the Process for Performing Foundation Grade Evaluations.

DEV.6.M349: Sanitise temporary variables

This mitigation is required to counter reading remnant volatile memory

At Foundation Grade the product is required to sanitise temporary variables containing sensitive information as soon as no longer required

A secure erase must consist of at least one complete overwrite with a fixed or random pattern and subsequent verification.

DEV.7 - Design >> Policy

DEV.7.M594: Policy to ensure correct application of encryption in email client

This mitigation is required to counter user only partially encrypting email (e.g. email has unencrypted attachment)

This mitigation is required to counter user failing to apply encryption to a message that requires confidentiality protection

At Foundation Grade the product is required to ensure all email content that requires encryption gets encrypted before being sent over the untrusted network (e.g. Internet)

B. Verification Mitigations

VER.M341: Audit permissions on product install

This mitigation is required to counter exploitation of a privileged local service

At Foundation Grade the evaluator will audit any system permissions and ACLs set or altered by the product during installation to ensure that no changes are made, which would give a standard user the ability to modify any components that run with higher privileges (either product or system provided).

VER.M80: Protocol robustness testing

This mitigation is required to counter discovery of a vulnerability in the implementation of the protocol

At Foundation Grade the evaluator will perform testing using commercial fuzzing tools

Fuzz testing is described in more detail in the Process for Performing Foundation Grade Evaluations.

VER.M347: Verify update mechanism

This mitigation is required to counter installing compromised software using the update process

At Foundation Grade the evaluator will validate the developer's assertions regarding the suitability and security of their update process

The update process must provide a mechanism by which updates can be authenticated before they are applied.
The process and any configuration required must be documented within the Security Procedures.

VER.M570: (Bespoke Protocol ONLY) Review protocol strength rationale

This mitigation is required to counter exploitation of vulnerabilities in the bespoke key exchange or digital signature protocol
This mitigation is required to counter exploitation of bespoke protocol vulnerability

At Foundation Grade the evaluator will review an analysis of the protocol provided by the developer to ensure it is logical and consistent

The developer must also provide the evaluator with a rationale as to why their bespoke email protection protocol provides equivalent security to OpenPGP or S/MIME. This rationale must explain how the cryptographic mechanisms outlined elsewhere in the SC are applied to emails and why the developer believes these provide an equal level of protection to OpenPGP and S/MIME.

The evaluator must review the developer's analysis and rationale to ensure it is logically consistent. The evaluator is not expected to perform a detailed cryptographic analysis of the protocol - but must ensure that there is a reason to believe the assertions made by the developer about the cryptographic protection provided by the product.

VER.1 - Verify >> Decryption

VER.1.M4: Evaluation/Cryptocheck

This mitigation is required to counter exploitation of flaws in the cryptographic algorithm implementation

At Foundation Grade the evaluator will verify correct cryptographic operation of email message decryption functionality

VER.2 - Verify >> Email Message Processing

VER.2.M4: Evaluation/Cryptocheck

This mitigation is required to counter unencrypted sensitive data leaking into encrypted message payload

At Foundation Grade the evaluator will check encoding process does not leak unencrypted red data into any part of the encrypted email message

VER.2.M566: Robustness against compression and decompression errors

This mitigation is required to counter decompression of corrupted data causing software crash

This mitigation is required to counter compression algorithm leaking sensitive data

At Foundation Grade the evaluator will check that decompression errors are not fatal and that the compression process will not leak potentially sensitive information

VER.3 - Verify >> Encryption

VER.3.M4: Evaluation/Cryptocheck

This mitigation is required to counter user failing to specify encryption for email when confidentiality required

This mitigation is required to counter malware replacing a randomly generated CEK with a fixed pattern

This mitigation is required to counter exploitation of a cryptographic algorithm implementation error

At Foundation Grade the evaluator will check a user cannot subvert the encryption process

The evaluation team should construct an encryption policy within the product and then verify that an internal user cannot trivially create emails which are not subject to these rules.

At Foundation Grade the evaluator will ensure all cryptographic algorithms employed for security functionality have been validated as per the "Cryptographic Validation" section in the CPA Foundation Process document

VER.3.M602: Prevent application of cryptographic protection using expired keys

This mitigation is required to counter key/passphrase validity periods not enforced on new application of key/passphrase

At Foundation Grade the evaluator will check expired cryptographic keys cannot be used to generate a new shared secret or digital signature, etc

VER.4 - Verify >> Management Interface

VER.4.M56: Management protocol robustness testing

This mitigation is required to counter exploitation of vulnerabilities in the management interface protocol

At Foundation Grade the evaluator will perform testing using commercial fuzzing tools

Fuzz testing is described in more detail in the Process for Performing Foundation Grade Evaluations.

VER.5 - Verify >> Policy

VER.5.M598: Restrict access to policy settings

This mitigation is required to counter unauthorised modification of policy through privilege escalation

At Foundation Grade the evaluator will ensure that only an authorised administrator can modify the product's policy settings

C. Deployment Mitigations

DEP.M38: Use automated configuration tool

This mitigation is required to counter exploitation of an accidental misconfiguration

At Foundation Grade the deployment is required to be configured using automated tools if provided

DEP.M39: Audit log review

This mitigation is required to counter exploitation of a software logic error

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the deployment is required to regularly review audit logs for unexpected entries

DEP.M46: User least privilege

This mitigation is required to counter taking advantage of existing user privilege

At Foundation Grade the deployment is required to ensure all user accounts have the fewest privileges required to enable business functionality

DEP.M131: Operating system verifies signatures

This mitigation is required to counter installation of a malicious privileged local service

At Foundation Grade the deployment is required to enable signature verification for applications, services and drivers in the host operating system, where supported and where the product makes use of it

DEP.M159: Update product

This mitigation is required to counter exploitation of a software logic error

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the deployment is required to update to the latest version where possible

DEP.M340: Address Space Layout Randomisation

This mitigation is required to counter exploitation of a software implementation error

At Foundation Grade the deployment is required to enable ASLR in the host Operating System where available

DEP.M348: Administrator authorised updates

This mitigation is required to counter installing compromised software using the update process

At Foundation Grade the deployment is required to confirm the source of updates before they are applied to the system

The administrator is required to have authorised the updates before use. If an automatic process is used, the administrator must also configure the product to authenticate updates.

The administrator is required to use the update process described within the Security Procedures.

DEP.M568: Interaction with user endpoint consistent with good practices for email security

This mitigation is required to counter exploitation of bad practice "encouraged" by encrypted email product

At Foundation Grade the deployment is required to ensure interactions with user endpoint are consistent with good practices for email security

For instance, avoiding requiring the user to retrieve encrypted email content by always clicking a link in a notification email message (user gets into habit of "blindly" clicking the link and then entering credentials to access the email).

DEP.M572: All public keys obtained from remote public key server/repository verify to a trusted entity

This mitigation is required to counter false key data set in remote public key server/repository

This mitigation is required to counter exploitation of general vulnerability in remote public key server/repository

This mitigation is required to counter spoofing of remote public key server/repository

At Foundation Grade the deployment is required to ensure that where a remote public key server/repository is used, all public keys retrieved from it can be verified back to a trusted key management entity

This is an entity (such as a CA) that is trusted by the security domains associated with the encrypted email deployment and which will have signed all the user endpoint public keys stored in the remote public key server/repository.

DEP.M580: Use email-aware anti-virus product

This mitigation is required to counter malware in infected domain spreading via encrypted email

At Foundation Grade the deployment is required to minimise risk of malware transfer through use of the latest email-aware anti-malware software (which is using the latest malware definitions)

To be deployed for use by all gateways, desktop clients and mail servers throughout the domain.

DEP.M582: Train users in appropriate use of email encryption

This mitigation is required to counter user failing to specify encryption for email when confidentiality required

This mitigation is required to counter sending a spoof email apparently from another trusted user containing malicious instructions

At Foundation Grade the deployment is required to only allow email encryption to be used by users who have been trained in (a) how to use email encryption and (b) good practices for secure email, such as recognising suspicious emails

DEP.M592: Protect against unauthorised access to host machine

This mitigation is required to counter unauthorised access to host machine

At Foundation Grade the deployment is required to limit the use of desktop email encryption to hosts with adequate operational safeguards

Specifically limit such hosts to (a) machines that are physically sited within the security domain's accreditation zone/s, or (b) work-issued laptops with encrypted hard disks, that are used strictly according to their security operating procedures (especially when used for remote working).

DEP.M600: Restrict the lifetime of a given long-term private key

This mitigation is required to counter key/passphrase being used enough times to significantly increase the chances of a brute-force attack succeeding

At Foundation Grade the deployment is required to constrain the lifetime of all long-term key agreement and digital signing keys to one year

I.e. once a year has passed after a key becomes active, the implementation shall then prevent further use of that key for encryption or digital signature generation.

DEP.1 - Deploy >> Account Management

DEP.1.M575: Certificates unambiguously identify associated user

This mitigation is required to counter replacement of valid recipient certificate with one associated with a compromised key

This mitigation is required to counter impersonation of a valid user to obtain an email encryption certificate

At Foundation Grade the deployment is required to ensure that signed certificates are only generated for legitimate users of the email encryption system

DEP.1.M578: Immediate revocation of user keys suspected of being compromised

This mitigation is required to counter attacker, masquerading as authorised user, sending malware in encrypted payload

This mitigation is required to counter attacker gaining access to user endpoint's desktop account to access their encrypted email private key(s)

This mitigation is required to counter user endpoint becoming untrusted (e.g. carrying out malicious activities)

At Foundation Grade the deployment is required to have processes to revoke any asymmetric key such that it cannot be used to encrypt or generate a signature with immediate effect

It must, for instance, be possible for the system administrator(s) to instigate this revocation mechanism in response to a user reporting suspected compromise of their account credentials (after authenticating that the user is who they say).

DEP.1.M579: Comprehensive user encrypted email account management

This mitigation is required to counter attacker, masquerading as authorised user, sending malware in encrypted payload

This mitigation is required to counter attacker capturing encrypted email account no longer used by user endpoint

This mitigation is required to counter user endpoint becoming untrusted (e.g. carrying out malicious activities)

This mitigation is required to counter a user's public key being erroneously revoked

At Foundation Grade the deployment is required to provide active user account management for the encrypted email system covering enrolment, revocation, reinstatement & deletion

The deployment must also ensure that all user endpoint public keys that are stored in a remote public key server/repository are updated according to the state of that user or user's key (e.g. if the user is revoked, the deployment must remove their keys from the public key server/repository).

DEP.2 - Deploy >> Email Message Processing

DEP.2.M573: (SMIME Protocol ONLY) Availability of infrastructure for revocation checks

This mitigation is required to counter replacement of valid recipient certificate with one associated with a compromised key

This mitigation is required to counter a private signing key of a trusted key management entity becoming compromised

At Foundation Grade the deployment is required to ensure latest key revocation details are available prior to validating a given certificate

DEP.2.M588: Check raw email content for malware

This mitigation is required to counter malware in infected domain spreading via encrypted email

At Foundation Grade the deployment is required to quarantine incoming and outgoing emails that have been identified as containing malware

Apply anti-malware checks immediately prior to encryption and immediately following decryption.

DEP.3 - Deploy >> Key Store

DEP.3.M124: Plan for recovery from compromise of long term secrets/keys

This mitigation is required to counter unauthorised export of secrets through physical interfaces

This mitigation is required to counter compromised device exfiltrating keys

This mitigation is required to counter exploitation of unintended information disclosure to leak keys/secrets

At Foundation Grade the deployment is required to provide secure means to replace compromised long term keys

Such as any trusted keys that are used to verify user endpoint key data held in remote public key servers/repositories.

DEP.4 - Deploy >> Management Interface

DEP.4.M50: Role based access control

This mitigation is required to counter unauthorised use of privilege to modify policy

At Foundation Grade the deployment is required to enforce a separate account for policy management with respect to other host device accounts

DEP.4.M51: Audit

This mitigation is required to counter unauthorised use of privilege to modify policy

At Foundation Grade the deployment is required to have its policy modification events recorded and audited, protected from account administrators

DEP.4.M53: Local management authentication

This mitigation is required to counter exploitation of poorly protected management interface

At Foundation Grade the deployment is required to enforce management activities to be authenticated via username/passphrase

This is intended to include serial console access, etc.

IV. GLOSSARY

24. The following definitions are used in this document:

Term	Description
ACL	Access Control List
API	Application Programmer Interface
ASN.1	Abstract Syntax Notation - notation describing data structures representing data and defining how it is encoded
ASLR	Address Space Layout Randomisation
ASCII Armor	[See Radix-64]
AV	Anti-virus
BCC	Blind Carbon Copy (as used in emails)
BER	Basic Encoding Rules - used for ASN.1 encodings
BLACK Network	Unsecured and untrustworthy network.
CA	Certification Authority - issues certificates within a PKI
CBC	Cipher Block Chaining
CEK	Content Encryption Key - a symmetric key that is used to encrypt user data, such as MIME-encoded email content

Term	Description
Certificate	Mechanism to associate cryptographic public keys with a user identities within a PKI, typically authenticated by a trusted third party, such as a CA
CFB	Cipher Feedback
CPA	Commercial Product Assurance
CRMF	Certificate Request Message Format - described in RFC 4211
DER	Distinguished Encoding Rules - a more constrained version of DER
DLP	Data Loss/Leak Prevention - systems that protect against unauthorised disclosure of sensitive data
DNS	Domain Name System
DoS	Denial of Service
DSA	Digital Signature Algorithm
ECDSA	Variant of DSA which uses Elliptic curve cryptography
Ephemeral	Typically refers to a “use-once” cryptographic key
FIPS	Federal Information Processing Standard (defined by NIST)
Fuzzing, fuzzer	Testing technique that provides invalid, unexpected, or random data to the inputs of an application or device
IBE	Identifier Based Encryption – type of PKC in which string representing an individual/organization is used as a public key.
IDPKC	Identity-based PKC (See IBE)
IETF	Internet Engineering Task Force
IMAP	Internet Messaging Access Protocol - an email retrieval mechanism (latest version at time of writing is defined in RFC 4880)
IMAP4rev1	Latest published version of IMAP at time of writing - defined in RFC 3501
IP	Internet Protocol
IPSEC	IP SECurity - protocol suite for securing IP communications
IV	Initialisation Vector – used in encryption feedback techniques
KDF	Key Derivation Function
KEK	Key Encryption Key - a symmetric key that is used to encrypt another symmetric key
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol - protocol for querying/modifying directory services data
MTA	Message Transfer Agent – software responsible for delivering emails
MUA	Message User Agent – email client application
MITM	Man In The Middle - an entity (usually malicious) that is able to intercept and modify messages sent between two points
NIST	National Institute of Standards and Technology
Nonce	Number-Once-Used - a counter or randomly-generated value used to ensure the input to a cryptographic algorithm (such as encryption) is varied each time it is invoked
OpenPGP	A standard that defines the application of PGP for secure email use (latest version at time of writing is defined in RFC 3156)
O/S	Operating System
PBE	Password-Based Encryption - a cryptographic mechanism that applies encryption using a passphrase
PDU	Protocol Data Unit
PGP	Pretty Good Privacy - a computer program for the encryption and decryption of data
PKC	Public Key Cryptography
PKCS	Public-Key Cryptography Standards - as published by RSA Security
PKI	Public Key Infrastructure - architecture that binds cryptographic public keys with user identities
POP	Post Office Protocol - an email retrieval mechanism
POP3	Latest published version of POP at time of writing - defined in RFC 1939
PRNG	Pseudo-Random Number Generation/Generator (see RNG)
Radix-64	Mechanism to encode OpenPGP content for transport

Term	Description
Red, Red-side	The more sensitive security domain, which is typically assumed to be the target of an attack.
RED Network	Secure/Sensitive network
Revocation, revoke	Process in which an entity's public key is marked as untrusted (e.g. through compromise)
RFC	Request For Comment - an IETF memorandum on Internet systems and standards
RNG	Random Number Generation/Generator
Salt	Random input to a key derivation function (such a function could be a PBE algorithm)
SC	[See Security Characteristic]
Security Characteristic	A standard which describes necessary mitigations which must be present in a completed product, its evaluation or usage, particular to a type of security product.
SHA	Secure Hash Algorithm - defined in the NIST publication: FIPS PUB 180-2
SHA-256	256 bit variant of SHA
S/MIME	Secure Multipurpose Internet Mail Extensions
S/MIMEv3.2	Latest published version of POP at time of writing - defined in RFC 5751
SMTP	Simple Mail transfer Protocol - an email sending mechanism (latest version at time of writing is defined in RFC 5321)
SNMP	Simple Network Management Protocol - a UDP-based network protocol
SNMPv3	Latest version of SNMP at time of writing - defined in RFCs 3411-3418
SSH	Secure Shell - a network protocol for remote administration of Unix computers
SSL	Secure Sockets Layer - see TLS
TCP	Transmission Control Protocol
TLS	Transport Layer Security (latest version at time of writing is defined in RFC 5246)
Web of Trust	Equivalent of PKI for OpenPGP - typically flatter hierarchy