



Secure Message Delivery

S/MIME Payload Profile

Version 1.0 draft — 17 September 2009

National E-Health Transition Authority Ltd

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

www.nehta.gov.au**Disclaimer**

NEHTA makes the information and other material (“Information”) in this document available in good faith but without any representation or warranty as to its accuracy or completeness. NEHTA cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document Control

This document is maintained in electronic form. The current revision of this document is located on the NEHTA Web site and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is of the latest revision.

Copyright © 2009, NEHTA.

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the permission of NEHTA. All copies of this document must include the copyright and other information contained on this page.

Table of contents

Table of contents	iii
Document information	iv
Change history	iv
1 Introduction	1
1.1 Background	1
1.2 Purpose	1
1.3 Scope	1
1.4 References	1
1.4.1 Normative References	1
1.4.2 Informative References	2
1.5 Definitions, acronyms, abbreviations	2
1.5.1 Terminology	2
1.6 Overview	2
2 SMD S/MIME payload profile	3
2.1 Introduction	3
2.2 Conformance	3
2.3 CMS enveloped-data	3
2.3.1 Enveloped-data	3
2.3.2 Recipient information	4
2.3.3 Encrypted content information	4
2.4 S/MIME signing container	5
2.4.1 Container	5
2.5 CMS signed-data	6
2.5.1 Signed-data	6
2.5.2 Encapsulated content	6
2.5.3 Certificates	7
2.5.4 Signer information	7
2.6 MIME attachment	7
3 Certificates	9
3.1 Criteria	9
3.1.1 Notes (non-normative)	9

Document information

Change history

Version	Date	Comments
1.0 draft	2009-09-17	Draft

1 Introduction

1.1 Background

The Secure Message Delivery (SMD) Endpoint Specification defines service interface specifications for delivering a payload which is secured from end-to-end. To achieve this, the specification defines that the payload must be secured with XML Secured Payload (XSP) or optionally with S/MIME.

The S/MIME payload profile is designed as an interim solution to facilitate the adoption of SMD in the short term.

This document defines a profile of S/MIME which minimises choice associated with the S/MIME standard and also interoperates with existing deployments of messaging software.

1.2 Purpose

This document defines the format of the S/MIME payload used inside the SMD service interface specification. The constraints defined in this profile are intended to maximise interconnectivity across S/MIME payload encryption implementations.

1.3 Scope

This document does not define the contents being secured inside the S/MIME format. Payload specifications associated with SMD will be defined separately.

This is not a general profile of S/MIME. It is not designed to cover any use outside SMD. In particular, it is not intended to be a specification of how S/MIME is to be used for securing email communications.

The reader of this document is expected to have a detailed understanding of S/MIME, Cryptographic Message Syntax (CMS) and MIME.

1.4 References

1.4.1 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For updated references, the latest edition of the referenced document (including any amendments) applies.

- [RFC2119] IETF, RFC 2119: *Keywords for use in RFCs to Indicate Requirement Levels*, S. Bradner, March 1997, <http://ietf.org/rfc/rfc2119.txt>
- [RFC2045] IETF, RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet message Bodies*, N. Freed, N. Borenstein, November 1996, <http://www.ietf.org/rfc/rfc2045.txt>
- [RFC2630] IETF, RFC 2630: *Cryptographic Message Syntax*, R. Housley, June 1999, <http://www.ietf.org/rfc/rfc2630.txt>
- [RFC2633] IETF, RFC 2633: *S/MIME Version 3 Message Specification*, B. Ramsdell (editor), June 1999, <http://www.ietf.org/rfc/rfc2633.txt>

1.4.2 Informative References

[RFC2315] IETF, RFC 2315: *PKCS #7: Cryptographic Message Syntax Version 1.5*, B. Kaliski, March 1998,
<http://www.ietf.org/rfc/rfc2315.txt>

1.5 Definitions, acronyms, abbreviations

CBC	Cipher Block Chaining
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
MIME	Multipurpose Internet Mail Extensions
PKI	Public Key Infrastructure
S/MIME	Secure/Multipurpose Internet Mail Extensions

1.5.1 Terminology

The keywords **MUST**, **MUST NOT**, **SHOULD**, **SHOULD NOT**, and **MAY** in this document are to be interpreted as described in IETF's RFC 2119 [RFC2119].

1.6 Overview

Chapter 2 contains the criteria of the profile.

Chapter 3 contains the criteria on certificates that must be used with the profile.

2 SMD S/MIME payload profile

2.1 Introduction

This is a profile for digitally signing and then encrypting.

2.2 Conformance

SM 1 Data conforming to this profile **MUST**:

- Be data in the format of the Cryptographic Message Syntax (CMS) *enveloped-data* as specified in section 2.3, and
- The plaintext of that *enveloped-data* is in the format of the S/MIME *signed data* as specified in section 2.4, and
- The signature part of that S/MIME *signed data* is in the format of the CMS *signed-data* as specified in section 2.5, and
- The data that is signed is a MIME message with the payload data represented as an attachment as specified in section 2.6.

The terms *enveloped-data*, *signed data* and *signed-data* refer to specific data structures defined in the CMS and S/MIME specifications.

2.3 CMS enveloped-data

2.3.1 Enveloped-data

SM 2 The data **MUST** comply with the Cryptographic Message Syntax *enveloped-data* as defined by [RFC2630].

SM 3 The enveloped-data **MUST NOT** contain *originator information*.

SM 4 The enveloped-data **MUST NOT** contain any *unprotect attributes*.

2.3.1.1 Notes (non-normative)

The actual representation of the enveloped-data is not specified by this profile, but will be defined by the context in which the data is used.

- When used inside the SMD specification, the encrypted container is represented as base64 encoded data inside an XML element.
- The enveloped-data could also be transmitted over other protocols. For example, if it is sent as an S/MIME email, it will be base64 encoded and placed inside a S/MIME part. In which case, appropriate *Content-Type* and *Content-Transfer-Encoding* headers would be added to the MIME headers.

The Cryptographic Message Syntax (CMS) is very similar to PKCS #7 [RFC2315] upon which it was based. This document is written in terms of CMS since it is the current specification at the time of writing, but in the way it is used it is not expected to be materially different from PKCS #7.

The *enveloped-data* contains up to five items (though this profile only uses three of them):

- Version (determined by the CMS specification to be version zero)
- Originator information (not used by this profile)
- Recipient information (see section 2.3.2)
- Encrypted content information (see section 2.3.3)
- Unprotected attributes (not used by this profile)

2.3.2 Recipient information

- SM 5 All *recipientInfo* **MUST** use the per-recipient “key transport” management technique.
- SM 6 All *keyTransportRecipientInfo* **MUST** identify the recipient’s public certificate using its issuer and serial number.

2.3.2.1 Notes (non-normative)

CMS allows for multiple recipients (i.e. multiple parties that can decrypt the enveloped-data). The *recipientInfos* item is a set of *recipientInfo* items. The CMS specification says there must be at least one member in that set. At least one of those *recipientInfo* items will be for the intended receiver so they can decrypt the data, but there can also be additional recipients. The above criteria apply to every *recipientInfo* item (not just the one for the intended receiver).

To reduce the risk of interoperability problems these criteria are applied to every *recipientInfo* item, even though they only need to apply to the *recipientInfo* for the intended receiver and other rules could apply to the other *recipientInfo* items that might be present.

This profile does not require multiple receivers to be used, but they can be useful in some situations. For example, the message creator can include themselves as one of the recipients so that they can archive the encrypted message and decrypt it using their private key. It has also been used when the message creator has several different certificates for the receiving party, such as during the transition period when an old certificate is about to expire and a new one has been issued.

CMS supports three different key management techniques. This profile only uses the “key transport” technique (*KeyTransRecipientInfo*), which means using the asymmetric encryption of PKI to protect the symmetric encryption key used to encrypt the actual data. The other two mechanisms, key agreement (*KeyAgreeRecipientInfo*) and pre-shared symmetric keys (*KEKRecipientInfo*), are not used by this profile.

CMS supports two ways of identifying which public certificate was used. This profile only uses the “issuer and serial number” to identify the certificate. The other mechanism, Subject Key Identifiers, is not used by this profile.

RSA encryption is the asymmetric encryption algorithm used to encrypt the symmetric key that is used to decrypt the encrypted content information. The use of RSA encryption is determined by the type of certificates being used. The use of RSA certificates is specified in Chapter 3.

2.3.3 Encrypted content information

- SM 7 The *encryptedContentInfo* **MUST** use one of these two algorithms:
- RC2 in CBC mode (*rc2-cbc*) with an effective-key-bits of 128, or
 - Triple-DES CBC (*des-ede3-cbc*)

2.3.3.1 Obligations on message creators

- SM 8 Message creators **MUST** be able to create messages using either *rc2-cbc*, *des-ede3-cbc*, or both.

If the message creator supports both, different messages can use one or the other, each message will use only one.

2.3.3.2 Obligations on message consumers

- SM 9 Message consumers **MUST** be able to consume messages that use the *des-ede3-cbc* encryption algorithm.

SM 10 Message consumers **MUST** be able to consume messages that use the rc2-cbc encryption algorithm.

2.3.3.3 Notes (non-normative)

These are the symmetric encryption algorithms used to encrypt the plaintext. There are two possible algorithms.

Message creators can choose which one they send. Message consumers must be capable of receiving both.

It is recommended that message creators use triple-DES CBC, if possible. Triple-DES is a mandatory part of the CMS specification, whereas RC2 is an optional part of the CMS specification. Even though RC2 is optional in the CMS specification, this profile makes it mandatory for message consumers to support it.

2.4 S/MIME signing container

2.4.1 Container

SM 11 The signing container **MUST** be an S/MIME signed message as defined by [RFC2632].

2.4.1.1 Obligations on message creators

SM 12 Message creators **MUST** be able to create S/MIME that uses either opaque signatures, cleartext signatures, or both.

If the message creator supports both, different messages can use one or the other, but each message will use only one.

2.4.1.2 Obligations on message consumers

SM 13 Message consumers **MUST** be able to consume S/MIME that uses opaque signatures.

SM 14 Message consumers **MUST** be able to consume S/MIME that uses cleartext signatures.

2.4.1.3 Notes (non-normative)

This S/MIME signing container is the plaintext of the enveloped-data that was defined in section 2.3.

S/MIME defines how to sign data that is formatted as a MIME entity. It supports two types of signatures: opaque signatures and cleartext signatures.

- An opaque signature is where the data being signed is inside the CMS *signed-data* content. With an opaque signature, the S/MIME signing container comprises of a single part (the CMS *signed-data*).
- A cleartext signature is one where the data being signed is represented outside the CMS *signed-data* content. This is sometimes referred to as a detached signature. With cleartext signing, the S/MIME signing container contains two parts: the data being signed and the CMS *signed-data* content.

Message creators can choose which one they send. Message consumers must be capable of receiving both.

It is recommended that message creators use opaque signing, if possible. This is because the majority of legacy implementations use opaque signing and it reduces of problems of incorrect MIME implementations corrupting the signature. This is opposite to the recommendation of the S/MIME specification, because the objectives of this profile are different from that of the S/MIME specification: the S/MIME specification is more concerned with

making the email content available to existing email clients that do not support S/MIME.

In the CMS standard, the correct MIME content type for the signing container is "application/pkcs7-mime". Some legacy systems incorrectly use the draft MIME type of "application/x-pkcs7-mime". Developers should be aware of this and decide on how their implementations handle these incorrectly tagged messages.

2.5 CMS signed-data

2.5.1 Signed-data

SM 15 The signed-data content **MUST** comply with the Cryptographic Message Syntax *signed-data* content type as defined by [RFC2630].

SM 16 The signed-data content **MUST NOT** contain any Certificate Revocation Lists (CRLs).

SM 17 The signed-data **MUST** contain exactly one *signer information*.

2.5.1.1 Notes (non-normative)

This CMS signed-data is the signature part of the S/MIME signing container that was defined in section 2.4.

Signed-data is the structure used by CMS to represent digital signatures.

The signed-data type can contain a list of Certificate Revocation Lists. This feature is not used by this profile.

The signed-data type can contain zero or many signer information items, to allow for zero or many signatures. This profile specifies there will be exactly one signature.

The signed-data contains up to six items (though this profile only uses five of them):

- Version (determined by the CMS specification)
- Digest algorithms (determined by the CMS specification)
- Encapsulated content information (see section 2.5.2)
- Certificates (see section 2.5.3)
- Certificate revocation lists (not used by this profile)
- Signer information (see section 2.5.4)

2.5.2 Encapsulated content

SM 18 The encapsulated content **MUST** contain *data content*.

2.5.2.1 Obligations on message creators

SM 19 Message creators **MUST** be able to create encapsulated content that supports either opaque signatures or cleartext signatures.

2.5.2.2 Obligations on message consumers

SM 20 Message consumers **MUST** be able to consume encapsulated content that supports opaque signatures.

SM 21 Message consumers **MUST** be able to consume encapsulated content that supports cleartext signatures.

2.5.2.3 Notes (non-normative)

Only *data content* is allowed in the encapsulated content to ensure compatibility between all implementations of this profile. For example, *compressed data content* is not permitted, because it is not supported by all S/MIME implementations.

See section 2.4.1.3 for a description of opaque signing vs cleartext signing. Section 2.4.1 and this section both refer to the same issue: the criteria in that section refer to how it affects the S/MIME container; the criteria in this section refer to how it affects the CMS signed-data item.

2.5.3 Certificates

SM 22 The signed-data **MUST** include the signing certificate corresponding to the signer information.

SM 23 The signed-data **SHOULD NOT** include any other certificate.

2.5.3.1 Notes (non-normative)

The signing certificate can be used to validate the signature without having to obtain the certificates by alternative means.

If other certificates are needed to validate the certificate chain, these criteria say they should not be included in the signed-data. Instead they will need to be obtained from another source. This encourages the receiver to explicitly trust the certificates in the chain, rather than assuming all the certificates provided in the signed-data are to be trusted.

2.5.4 Signer information

SM 24 The signing certificate **MUST** be identified using its issuer and serial number.

SM 25 The signer information **MUST** use SHA-1 as the digest algorithm.

SM 26 The signer information **MUST** use the RSA encryption algorithm (rsaEncryption) for signing.

2.5.4.1 Obligations on message consumers

SM 27 Message consumers **MUST** handle or ignore any other attribute that might be present in the signer information part of the *signed-data*.

2.5.4.2 Notes (non-normative)

The *signer information* represents one signature over the attributes being signed. As defined in section 2.5.1, this profile only has one signature and therefore it contains exactly one *signer information*.

The attributes being signed in the signer information will usually contain at least the contentType, signingTime, and messageDigest. The CMS specification specifies that the contentType and messageDigest attributes are mandatory. The S/MIME specification specifies that the signingTime should also be present.

The attributes being signed in the signer information can contain additional attributes. Some of them are recommended by the S/MIME specification, but they are designed for use in an email context and have less relevance in this context. Message consumers need to handle the presence of other attributes that it does not use and are not required to be produced by this profile.

2.6 MIME attachment

SM 28 The payload data **MUST** be represented as the one and only attachment in a MIME message as define by [RFC2045].

SM 29 The payload data **MUST NOT** use a compressed transfer encoding.

2.6.1.1 Notes (non-normative)

This is the data that was signed by the CMS signed-data that was defined in section 2.5.

S/MIME signs MIME entities, so any data to be signed must be represented as a MIME message. The data will have to be represented as a MIME attachment. Since this profile has been designed to protect one piece of data, only one MIME attachment is permitted.

For binary data (such as HL7 v2 messages) it is recommended that they are sent as an attachment with MIME type of application/octet-string and transfer encoding of base64. Some legacy systems incorrectly transmit HL7 v2 messages using the text/plain MIME type and marked as 7bit encoding. The 7bit encoding has precise rules about maximum line lengths and how line endings are represented, but raw HL7 v2 does not meet those rules. Developers should be aware of this and decide on how their implementations handle these incorrectly formatted messages.

3 Certificates

The criteria in this chapter define the types of X.509 PKI certificates that can be used with the profile.

3.1 Criteria

- SM 30 Certificates used with the profile **MUST** be globally uniquely identifiable with their issuer and serial number.
- SM 31 Certificates used with this profile **MUST** support RSA encryption.
- SM 32 Certificates used with this profile **MUST NOT** be *attribute certificates*.

3.1.1 Notes (non-normative)

These certificate criteria are strongly linked with the S/MIME profile.

For example, the S/MIME profile requires certificates to be identified using their issuer and serial number. Therefore, the certificates used must be identifiable in this manner. Most certificate issuer will assign certificates with unique serial numbers, so this criterion is usually satisfied.

The S/MIME profile expects that the symmetric key is encrypted using RSA encryption. The algorithm being used is determined by the certificates being used, rather than the implementation of S/MIME. This is why the RSA encryption criterion is placed on the certificates, rather than in the S/MIME profile.