        Secure/Multipurpose Internet Mail Extensions (S/ MIME) Version 4.0
                          Certificate Handling
                       draft-ietf-lamps-rfc5750-bis-07

Abstract

   This document specifies conventions for X.509 certificate usage by
   Secure/Multipurpose Internet Mail Extensions (S/MIME) v4.0 agents.
   S/MIME provides a method to send and receive secure MIME messages,
   and certificates are an integral part of S/MIME agent processing.
   S/MIME agents validate certificates as described in RFC 5280, the
   Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
   S/MIME agents must meet the certificate processing requirements in
   this document as well as those in RFC 5280. This document obsoletes
   RFC 5750.

Contributing to this document

   The source for this draft is being maintained in GitHub. Suggested
   changes should be submitted as pull requests at <https://github.com/
   lamps-wg/smime>. Instructions are on that page as well. Editorial
   changes can be managed in GitHub, but any substantial issues need to
   be discussed on the LAMPS mailing list.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on December 23, 2018.

Copyright Notice

Table of Contents

1. Introduction

   S/MIME (Secure/Multipurpose Internet Mail Extensions) v4.0, described
   in [I-D.ietf-lamps-rfc5751-bis], provides a method to send and
   receive secure MIME messages. Before using a public key to provide
   security services, the S/MIME agent MUST verify that the public key
   is valid. S/MIME agents MUST use PKIX certificates to validate
   public keys as described in the Internet X.509 Public Key
   Infrastructure (PKIX) Certificate and CRL Profile [RFC5280]. S/MIME
   agents MUST meet the certificate processing requirements documented
   in this document in addition to those stated in [RFC5280].

   This specification is compatible with the Cryptographic Message
   Syntax (CMS) RFC 5652 [RFC5652] in that it uses the data types
   defined by CMS. It also inherits all the varieties of architectures
   for certificate-based key management supported by CMS.

   This document obsoletes [RFC5750]. The most significant changes
   revolve around changes in recommendations around the cryptographic
   algorithms used by the specification. More details can be found in
   Section 1.6.

1.1. Definitions

   For the purposes of this document, the following definitions apply.

   ASN.1: Abstract Syntax Notation One, as defined in ITU-T X.680
   [X.680].

   Attribute certificate (AC): An X.509 AC is a separate structure from
   a subject's public key X.509 certificate. A subject may have

   multiple X.509 ACs associated with each of its public key X.509
   certificates. Each X.509 AC binds one or more attributes with one of
   the subject's public key X.509 certificates. The X.509 AC syntax is
   defined in [RFC5755].

   Certificate: A type that binds an entity's name to a public key with
   a digital signature. This type is defined in the Internet X.509
   Public Key Infrastructure (PKIX) Certificate and CRL Profile
   [RFC5280]. This type also contains the distinguished name of the
   certificate issuer (the signer), an issuer-specific serial number,
   the issuer's signature algorithm identifier, a validity period, and
   extensions also defined in that document.

   Certificate Revocation List (CRL): A type that contains information
   about certificates whose validity an issuer has prematurely revoked.
   The information consists of an issuer name, the time of issue, the
   next scheduled time of issue, a list of certificate serial numbers
   and their associated revocation times, and extensions as defined in
   [RFC5280]. The CRL is signed by the issuer. The type intended by
   this specification is the one defined in [RFC5280].

   Receiving agent: Software that interprets and processes S/MIME CMS
   objects, MIME body parts that contain CMS objects, or both.

   Sending agent: Software that creates S/MIME CMS objects, MIME body
   parts that contain CMS objects, or both.

   S/MIME agent: User software that is a receiving agent, a sending
   agent, or both.

1.2. Conventions Used in This Document

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in BCP
   14 [RFC2119] [RFC8174] when, and only when, they appear in all
   capitals, as shown here.

   We define the additional requirement levels:

   SHOULD+ This term means the same as SHOULD. However, the authors
           expect that a requirement marked as SHOULD+ will be promoted
           at some future time to be a MUST.

   SHOULD- This term means the same as SHOULD. However, the authors
           expect that a requirement marked as SHOULD- will be demoted
           to a MAY in a future version of this document.

   MUST-   This term means the same as MUST. However, the authors
           expect that this requirement will no longer be a MUST in a
           future document. Although its status will be determined at a
           later time, it is reasonable to expect that if a future
           revision of a document alters the status of a MUST-
           requirement, it will remain at least a SHOULD or a SHOULD-.

   The term RSA in this document almost always refers to the PKCS#1 v1.5
   RSA signature algorithm even when not qualified as such. There are a
   couple of places where it refers to the general RSA cryptographic
   operation; these can be determined from the context where it is used.

1.3. Compatibility with Prior Practice S/MIME

   S/MIME version 4.0 agents ought to attempt to have the greatest
   interoperability possible with agents for prior versions of S/MIME.

   S/MIME version 2 is described in RFC 2311 through RFC 2315 inclusive
   [SMIMEv2], S/MIME version 3 is described in RFC 2630 through RFC 2634
   inclusive and RFC 5035 [SMIMEv3], and S/MIME version 3.1 is described
   in RFC 3850, RFC 3851, RFC 3852, RFC 2634, and RFC 5035 [SMIMEv3.1].
   RFC 2311 also has historical information about the development of
   S/MIME.

   Appendix A contains information about algorithms that were used for
   prior versions of S/MIME but are no longer considered to meet modern
   security standards. Support of these algorithms may be needed to
   support historic S/MIME artifacts such as messages or files, but
   SHOULD NOT be used for new artifacts.

1.4. Changes from S/MIME v3 to S/MIME v3.1

   This section reflects the changes that were made when S/MIME v3.1 was
   released. The RFC2119 langauage may have superceeded in later
   versions.

   Version 1 and version 2 CRLs MUST be supported.

   Multiple certification authority (CA) certificates with the same
   subject and public key, but with overlapping validity periods, MUST
   be supported.

   Version 2 attribute certificates SHOULD be supported, and version 1
   attributes certificates MUST NOT be used.

   The use of the MD2 digest algorithm for certificate signatures is
   discouraged, and security language was added.

Clarified use of email address use in certificates. Certificates
that do not contain an email address have no requirements for
verifying the email address associated with the certificate.

Receiving agents SHOULD display certificate information when
displaying the results of signature verification.

Receiving agents MUST NOT accept a signature made with a certificate
that does not have at least one of the the digitalSignature or
nonRepudiation bits set.

Clarifications for the interpretation of the key usage and extended
key usage extensions.

1.5. Changes from S/MIME v3.1 to S/MIME v3.2

This section reflects the changes that were made when S/MIME v3.2 was
released. The RFC2119 langauage may have superceeded in later
versions.

Conventions Used in This Document: Moved to Section 1.2. Added
definitions for SHOULD+, SHOULD-, and MUST-.

Section 1.1: Updated ASN.1 definition and reference.

Section 1.3: Added text about v3.1 RFCs.

Section 3: Aligned email address text with RFC 5280. Updated note
            to indicate emailAddress IA5String upper bound is 255
            characters. Added text about matching email addresses.

Section 4.2: Added text to indicate how S/MIME agents locate the
              correct user certificate.

Section 4.3: RSA with SHA-256 (PKCS #1 v1.5) added as MUST; DSA with
              SHA-256 added as SHOULD+; RSA with SHA-1, DSA with SHA-1,
              and RSA with MD5 changed to SHOULD-; and RSASSA-PSS with
              SHA-256 added as SHOULD+. Updated key sizes and changed
              pointer to PKIX RFCs.

Section 4.4.1: Aligned with PKIX on use of basic constraints
                extension in CA certificates. Clarified which extension
                is used to constrain end entities from using their keys
                to perform issuing authority operations.

Section 5: Updated security considerations.

   Section 7: Moved references from Appendix B to Section 6. Updated
               the references.

   Appendix A: Moved Appendix A to Appendix B. Added Appendix A to move
               S/MIME v2 Certificate Handling to Historic Status.

1.6. Changes since S/MIME 3.2

   This section reflects the changes that were made when S/MIME v4.0 was
   released. The RFC2119 langauage may have superceeded in later
   versions.

   Section 3: Require support for internationalized email addresses.

   Section 4.3: Mandated support for ECDSA with P-256 and Ed25519.
               Moved algorithms with SHA-1 and MD5 to historical status.
               Moved DSA support to historical status. Increased lower
               bounds on RSA key sizes.

   Appendix A: Add a new appendix for algorithms that are now considered
               to be historical.

2. CMS Options

   The CMS message format allows for a wide variety of options in
   content and algorithm support. This section puts forth a number of
   support requirements and recommendations in order to achieve a base
   level of interoperability among all S/MIME implementations. Most of
   the CMS format for S/MIME messages is defined in
   [I-D.ietf-lamps-rfc5751-bis].

2.1. Certificate Revocation Lists

   Receiving agents MUST support the Certificate Revocation List (CRL)
   format defined in [RFC5280]. If sending agents include CRLs in
   outgoing messages, the CRL format defined in [RFC5280] MUST be used.
   Receiving agents MUST support both v1 and v2 CRLs.

   All agents MUST be capable of performing revocation checks using CRLs
   as specified in [RFC5280]. All agents MUST perform revocation status
   checking in accordance with [RFC5280]. Receiving agents MUST
   recognize CRLs in received S/MIME messages.

   Agents SHOULD store CRLs received in messages for use in processing
   later messages.

2.2. Certificate Choices

Receiving agents MUST support v1 X.509 and v3 X.509 certificates as profiled in [RFC5280]. End-entity certificates MAY include an Internet mail address, as described in Section 3.

Receiving agents SHOULD support X.509 version 2 attribute certificates. See [RFC5755] for details about the profile for attribute certificates.

2.2.1. Historical Note about CMS Certificates

The CMS message format supports a choice of certificate formats for public key content types: PKIX, PKCS #6 extended certificates [PKCS6], and PKIX attribute certificates.

The PKCS #6 format is not in widespread use. In addition, PKIX certificate extensions address much of the same functionality and flexibility as was intended in the PKCS #6. Thus, sending and receiving agents MUST NOT use PKCS #6 extended certificates. Receiving agents MUST be able to parse and process a message containing PKCS #6 extended certificates although ignoring those certificates is expected behavior.

X.509 version 1 attribute certificates are also not widely implemented, and have been superseded with version 2 attribute certificates. Sending agents MUST NOT send version 1 attribute certificates.

2.3. CertificateSet

Receiving agents MUST be able to handle an arbitrary number of certificates of arbitrary relationship to the message sender and to each other in arbitrary order. In many cases, the certificates included in a signed message may represent a chain of certification from the sender to a particular root. There may be, however, situations where the certificates in a signed message may be unrelated and included for convenience.

Sending agents SHOULD include any certificates for the user's public key(s) and associated issuer certificates. This increases the likelihood that the intended recipient can establish trust in the originator's public key(s). This is especially important when sending a message to recipients that may not have access to the sender's public key through any other means or when sending a signed message to a new recipient. The inclusion of certificates in outgoing messages can be omitted if S/MIME objects are sent within a group of correspondents that has established access to each other's

certificates by some other means such as a shared directory or manual
certificate distribution. Receiving S/MIME agents SHOULD be able to
handle messages without certificates by using a database or directory
lookup scheme to find them.

A sending agent SHOULD include at least one chain of certificates up
to, but not including, a certification authority (CA) that it
believes that the recipient may trust as authoritative. A receiving
agent MUST be able to handle an arbitrarily large number of
certificates and chains.

Agents MAY send CA certificates, that is, cross-certificates, self-
issued certificates, and self-signed certificates. Note that
receiving agents SHOULD NOT simply trust any self-signed certificates
as valid CAs, but SHOULD use some other mechanism to determine if
this is a CA that should be trusted. Also note that when
certificates contain Digital Signature Algorithm (DSA) public keys
the parameters may be located in the root certificate. This would
require that the recipient possess both the end-entity certificate
and the root certificate to perform a signature verification, and is
a valid example of a case where transmitting the root certificate may
be required.

Receiving agents MUST support chaining based on the distinguished
name fields. Other methods of building certificate chains MAY be
supported.

Receiving agents SHOULD support the decoding of X.509 attribute
certificates included in CMS objects. All other issues regarding the
generation and use of X.509 attribute certificates are outside of the
scope of this specification. One specification that addresses
attribute certificate use is defined in [RFC3114].

3. Using Distinguished Names for Internet Mail

End-entity certificates MAY contain an Internet mail address. Email
addresses restricted to 7-bit ASCII characters use the pkcs-9-at-
emailAddress OID (see below) and are encoded as described in
Section 4.2.1.6 of [RFC5280]. Internationalized Email address names
use the OID defined in [I-D.ietf-lamps-eai-addresses] and are encoded
as described there. The email address SHOULD be in the
subjectAltName extension, and SHOULD NOT be in the subject
distinguished name.

Receiving agents MUST recognize and accept certificates that contain
no email address. Agents are allowed to provide an alternative
mechanism for associating an email address with a certificate that
does not contain an email address, such as through the use of the

agent's address book, if available. Receiving agents MUST recognize both ASCII and internationalized email addresses in the subjectAltName field. Receiving agents MUST recognize email addresses in the Distinguished Name field in the PKCS #9 [RFC2985] emailAddress attribute:

pkcs-9-at-emailAddress OBJECT IDENTIFIER ::=
 { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs-9(9) 1 }

Note that this attribute MUST be encoded as IA5String and has an upper bound of 255 characters. The right side of the email address SHOULD be treated as ASCII-case-insensitive.

Comparing of email addresses is fraught with peril. [I-D.ietf-lamps-eai-addresses] defines the procedure for doing comparison of Internationalized email addresses. For ASCII email addresses the domain component (right-hand side of the '@') MUST be compared using a case-insensitive function. The local name component (left-hand side of the '@') SHOULD be compared using a case-insensitive function. Some localities may perform other transformations on the local name component before doing the comparison, however an S/MIME client cannot know what specific localities do.

Sending agents SHOULD make the address in the From or Sender header in a mail message match an Internet mail address in the signer's certificate. Receiving agents MUST check that the address in the From or Sender header of a mail message matches an Internet mail address in the signer's certificate, if mail addresses are present in the certificate. A receiving agent SHOULD provide some explicit alternate processing of the message if this comparison fails; this might be done by displaying or logging a message that shows the recipient the mail addresses in the certificate or other certificate details.

A receiving agent SHOULD display a subject name or other certificate details when displaying an indication of successful or unsuccessful signature verification.

All subject and issuer names MUST be populated (i.e., not an empty SEQUENCE) in S/MIME-compliant X.509 certificates, except that the subject distinguished name (DN) in a user's (i.e., end-entity) certificate MAY be an empty SEQUENCE in which case the subjectAltName extension will include the subject's identifier and MUST be marked as critical.

4. Certificate Processing

   S/MIME agents need to provide some certificate retrieval mechanism in
   order to gain access to certificates for recipients of digital
   envelopes. There are many ways to implement certificate retrieval
   mechanisms. [X.500] directory service is an excellent example of a
   certificate retrieval-only mechanism that is compatible with classic
   X.500 Distinguished Names. The IETF has published [RFC8162] which
   describes an experimental protocol to retrieve certificates from the
   Domain Name System (DNS). Until such mechanisms are widely used,
   their utility may be limited by the small number of the
   correspondent's certificates that can be retrieved. At a minimum,
   for initial S/MIME deployment, a user agent could automatically
   generate a message to an intended recipient requesting the
   recipient's certificate in a signed return message.

   Receiving and sending agents SHOULD also provide a mechanism to allow
   a user to "store and protect" certificates for correspondents in such
   a way so as to guarantee their later retrieval. In many
   environments, it may be desirable to link the certificate retrieval/
   storage mechanisms together in some sort of certificate database. In
   its simplest form, a certificate database would be local to a
   particular user and would function in a similar way as an "address
   book" that stores a user's frequent correspondents. In this way, the
   certificate retrieval mechanism would be limited to the certificates
   that a user has stored (presumably from incoming messages). A
   comprehensive certificate retrieval/storage solution might combine
   two or more mechanisms to allow the greatest flexibility and utility
   to the user. For instance, a secure Internet mail agent might resort
   to checking a centralized certificate retrieval mechanism for a
   certificate if it cannot be found in a user's local certificate
   storage/retrieval database.

   Receiving and sending agents SHOULD provide a mechanism for the
   import and export of certificates, using a CMS certs-only message.
   This allows for import and export of full certificate chains as
   opposed to just a single certificate. This is described in
   [RFC5751].

   Agents MUST handle multiple valid certification authority (CA)
   certificates containing the same subject name and the same public
   keys but with overlapping validity intervals.

4.1. Certificate Revocation Lists

   In general, it is always better to get the latest CRL information
   from a CA than to get information stored in an incoming messages. A
   receiving agent SHOULD have access to some CRL retrieval mechanism in

order to gain access to certificate revocation information when
validating certification paths. A receiving or sending agent SHOULD
also provide a mechanism to allow a user to store incoming
certificate revocation information for correspondents in such a way
so as to guarantee its later retrieval.

Receiving and sending agents SHOULD retrieve and utilize CRL
information every time a certificate is verified as part of a
certification path validation even if the certificate was already
verified in the past. However, in many instances (such as off-line
verification) access to the latest CRL information may be difficult
or impossible. The use of CRL information, therefore, may be
dictated by the value of the information that is protected. The
value of the CRL information in a particular context is beyond the
scope of this specification but may be governed by the policies
associated with particular certification paths.

All agents MUST be capable of performing revocation checks using CRLs
as specified in [RFC5280]. All agents MUST perform revocation status
checking in accordance with [RFC5280]. Receiving agents MUST
recognize CRLs in received S/MIME messages.

4.2. Certificate Path Validation

In creating a user agent for secure messaging, certificate, CRL, and
certification path validation should be highly automated while still
acting in the best interests of the user. Certificate, CRL, and path
validation MUST be performed as per [RFC5280] when validating a
correspondent's public key. This is necessary before using a public
key to provide security services such as verifying a signature,
encrypting a content-encryption key (e.g., RSA), or forming a
pairwise symmetric key (e.g., Diffie-Hellman) to be used to encrypt
or decrypt a content-encryption key.

Certificates and CRLs are made available to the path validation
procedure in two ways: a) incoming messages, and b) certificate and
CRL retrieval mechanisms. Certificates and CRLs in incoming messages
are not required to be in any particular order nor are they required
to be in any way related to the sender or recipient of the message
(although in most cases they will be related to the sender).
Incoming certificates and CRLs SHOULD be cached for use in path
validation and optionally stored for later use. This temporary
certificate and CRL cache SHOULD be used to augment any other
certificate and CRL retrieval mechanisms for path validation on
incoming signed messages.

When verifying a signature and the certificates that are included in
the message, if a signingCertificate attribute from RFC 2634 [ESS] or

a signingCertificateV2 attribute from RFC 5035 [ESS] is found in an
S/MIME message, it SHALL be used to identify the signer's
certificate. Otherwise, the certificate is identified in an S/MIME
message, either using the issuerAndSerialNumber, which identifies the
signer's certificate by the issuer's distinguished name and the
certificate serial number, or the subjectKeyIdentifier, which
identifies the signer's certificate by a key identifier.

When decrypting an encrypted message, if a
SMIMEEncryptionKeyPreference attribute is found in an encapsulating
SignedData, it SHALL be used to identify the originator's certificate
found in OriginatorInfo. See [RFC5652] for the CMS fields that
reference the originator's and recipient's certificates.

4.3. Certificate and CRL Signing Algorithms and Key Sizes

Certificates and Certificate Revocation Lists (CRLs) are signed by
the certificate issuer. Receiving agents:

- MUST support ECDSA with curve P-256 with SHA-256.

- MUST support EdDSA with curve 25519 using PureEdDSA mode.

- MUST- support RSA PKCS#1 v1.5 with SHA-256.

- SHOULD support RSASSA-PSS with SHA-256.

Implementations SHOULD use deterministic generation for the parameter
'k' for ECDSA as outlined in [RFC6979]. EdDSA is defined to generate
this parameter deterministically.

The following are the RSA and RSASSA-PSS key size requirements for
S/MIME receiving agents during certificate and CRL signature
verification:

         key size <= 2047 : SHOULD NOT (see Historic Considerations)
2048 <= key size <= 4096 : MUST (see Security Considerations)
4096 < key size   : MAY (see Security Considerations)

The signature algorithm object identifiers for RSA PKCS#1 v1.5 and
RSASSA-PSS with SHA-256 using 1024-bit through 3072-bit public keys
are specified in [RFC4055] and the signature algorithm definition is
found in [FIPS186-2] with Change Notice 1.

The signature algorithm object identifiers for RSA PKCS#1 v1.5 and
RSASSA-PSS with SHA-256 using 4096-bit public keys are specified in
[RFC4055] and the signature algorithm definition is found in
[RFC3447].

For RSASSA-PSS with SHA-256 see [RFC4056].

For ECDSA see [RFC5758] and [RFC6090]. The first reference provides
the signature algorithm's object identifier and the second provides
the signature algorithm's definition. Curves other than curve P-256
MAY be used as well.

For EdDSA see [I-D.ietf-curdle-pkix] and [RFC8032]. The first
reference provides the signature algorithm's object identifier and
the second provides the signature algorithm's definition. Other
curves than curve 25519 MAY be used as well.

4.4. PKIX Certificate Extensions

PKIX describes an extensible framework in which the basic certificate
information can be extended and describes how such extensions can be
used to control the process of issuing and validating certificates.
The LAMPS Working Group has ongoing efforts to identify and create
extensions that have value in particular certification environments.
Further, there are active efforts underway to issue PKIX certificates
for business purposes. This document identifies the minimum required
set of certificate extensions that have the greatest value in the
S/MIME environment. The syntax and semantics of all the identified
extensions are defined in [RFC5280].

Sending and receiving agents MUST correctly handle the basic
constraints, key usage, authority key identifier, subject key
identifier, and subject alternative names certificate extensions when
they appear in end-entity and CA certificates. Some mechanism SHOULD
exist to gracefully handle other certificate extensions when they
appear in end-entity or CA certificates.

Certificates issued for the S/MIME environment SHOULD NOT contain any
critical extensions (extensions that have the critical field set to
TRUE) other than those listed here. These extensions SHOULD be
marked as non-critical unless the proper handling of the extension is
deemed critical to the correct interpretation of the associated
certificate. Other extensions may be included, but those extensions
SHOULD NOT be marked as critical.

Interpretation and syntax for all extensions MUST follow [RFC5280],
unless otherwise specified here.

4.4.1. Basic Constraints

The basic constraints extension serves to delimit the role and
position that an issuing authority or end-entity certificate plays in
a certification path.

For example, certificates issued to CAs and subordinate CAs contain a basic constraints extension that identifies them as issuing authority certificates. End-entity certificates contain the key usage extension that restrains end-entities from using the key when performing issuing authority operations (see Section 4.4.2).

As per [RFC5280], certificates MUST contain a basicConstraints extension in CA certificates, and SHOULD NOT contain that extension in end-entity certificates.

4.4.2. Key Usage Certificate Extension

The key usage extension serves to limit the technical purposes for which a public key listed in a valid certificate may be used. Issuing authority certificates may contain a key usage extension that restricts the key to signing certificates, certificate revocation lists, and other data.

For example, a certification authority may create subordinate issuer certificates that contain a key usage extension that specifies that the corresponding public key can be used to sign end user certificates and sign CRLs.

If a key usage extension is included in a PKIX certificate, then it MUST be marked as critical.

S/MIME receiving agents MUST NOT accept the signature of a message if it was verified using a certificate that contains the key usage extension without at least one of the digitalSignature or nonRepudiation bits set. Sometimes S/MIME is used as a secure message transport for applications beyond interpersonal messaging; in such cases, the S/MIME-enabled application can specify additional requirements concerning the digitalSignature or nonRepudiation bits within this extension.

If the key usage extension is not specified, receiving clients MUST presume that both the digitalSignature and nonRepudiation bits are set.

4.4.3. Subject Alternative Name

The subject alternative name extension is used in S/MIME as the preferred means to convey the email address(es) that correspond(s) to the entity for this certificate. If the local portion of the email address is ASCII, it MUST be encoded using the rfc822Name CHOICE of the GeneralName type as described in [RFC5280], Section 4.2.1.6. If the local portion of the email address is not ASCII, it MUST be encoded using the otherName CHOICE of the GeneralName type as

described in [I-D.ietf-lamps-eai-addresses], Section 3. Since the
SubjectAltName type is a SEQUENCE OF GeneralName, multiple email
addresses MAY be present.

4.4.4. Extended Key Usage Extension

The extended key usage extension also serves to limit the technical
purposes for which a public key listed in a valid certificate may be
used. The set of technical purposes for the certificate therefore
are the intersection of the uses indicated in the key usage and
extended key usage extensions.

For example, if the certificate contains a key usage extension
indicating digital signature and an extended key usage extension that
includes the email protection OID, then the certificate may be used
for signing but not encrypting S/MIME messages. If the certificate
contains a key usage extension indicating digital signature but no
extended key usage extension, then the certificate may also be used
to sign but not encrypt S/MIME messages.

If the extended key usage extension is present in the certificate,
then interpersonal message S/MIME receiving agents MUST check that it
contains either the emailProtection or the anyExtendedKeyUsage OID as
defined in [RFC5280]. S/MIME uses other than interpersonal messaging
MAY require the explicit presence of the extended key usage extension
or other OIDs to be present in the extension or both.

5. IANA Considertions

This document has no new IANA considerations.

6. Security Considerations

All of the security issues faced by any cryptographic application
must be faced by a S/MIME agent. Among these issues are protecting
the user's private key, preventing various attacks, and helping the
user avoid mistakes such as inadvertently encrypting a message for
the wrong recipient. The entire list of security considerations is
beyond the scope of this document, but some significant concerns are
listed here.

When processing certificates, there are many situations where the
processing might fail. Because the processing may be done by a user
agent, a security gateway, or other program, there is no single way
to handle such failures. Just because the methods to handle the
failures have not been listed, however, the reader should not assume
that they are not important. The opposite is true: if a certificate
is not provably valid and associated with the message, the processing

software should take immediate and noticeable steps to inform the end user about it.

Some of the many places where signature and certificate checking might fail include:

- no Internet mail addresses in a certificate match the sender of a message, if the certificate contains at least one mail address

- no certificate chain leads to a trusted CA

- no ability to check the CRL for a certificate

- an invalid CRL was received

- the CRL being checked is expired

- the certificate is expired

- the certificate has been revoked

There are certainly other instances where a certificate may be invalid, and it is the responsibility of the processing software to check them all thoroughly, and to decide what to do if the check fails.

It is possible for there to be multiple unexpired CRLs for a CA. If an agent is consulting CRLs for certificate validation, it SHOULD make sure that the most recently issued CRL for that CA is consulted, since an S/MIME message sender could deliberately include an older unexpired CRL in an S/MIME message. This older CRL might not include recently revoked certificates, which might lead an agent to accept a certificate that has been revoked in a subsequent CRL.

When determining the time for a certificate validity check, agents have to be careful to use a reliable time. In most cases the time used SHOULD be the current time, some exceptions to this would be:

- The time the message was received is stored in a secure manner and is used at a later time to validate the message.

- The time in a SigningTime attribute found in a counter signature attribute which has been successfully validated.

The SigningTime attribute could be deliberately set to direct the receiving agent to check a CRL that could have out-of-date revocation status for a certificate, or cause an improper result when checking the Validity field of a certificate. This could be done either by

the sender of the message, or an attacker which has compromised the
key of the sender.

In addition to the Security Considerations identified in [RFC5280],
caution should be taken when processing certificates that have not
first been validated to a trust anchor. Certificates could be
manufactured by untrusted sources for the purpose of mounting denial
of service or other attacks. For example, keys selected to require
excessive cryptographic processing, or extensive lists of CRL
Distribution Point (CDP) and/or Authority Information Access (AIA)
addresses in the certificate, could be used to mount denial-of-
service attacks. Similarly, attacker-specified CDP and/or AIA
addresses could be included in fake certificates to allow the
originator to detect receipt of the message even if signature
verification fails.

RSA keys of less than 2048 bits are now considered by many experts to
be cryptographically insecure (due to advances in computing power),
and SHOULD no longer be used to sign certificates or CRLs. Such keys
were previously considered secure, so processing previously received
signed and encrypted mail may require processing certificates or CRLs
signed with weak keys. Implementations that wish to support previous
versions of S/MIME or process old messages need to consider the
security risks that result from accepting certificates and CRLs with
smaller key sizes (e.g., spoofed certificates) versus the costs of
denial of service. If an implementation supports verification of
certificates or CRLs generated with RSA and DSA keys of less than
2048 bits, it MUST warn the user. Implementers should consider
providing a stronger warning for weak signatures on certificates and
CRLs associated with newly received messages than the one provided
for certificates and CRLs associated with previously stored messages.
Server implementations (e.g., secure mail list servers) where user
warnings are not appropriate SHOULD reject messages with weak
cryptography.

If an implementation is concerned about compliance with National
Institute of Standards and Technology (NIST) key size
recommendations, then see [SP800-57].

7. References

7.1. Normative References

[FIPS186-2]
           National Institute of Standards and Technology (NIST),
           "Digital Signature Standard (DSS) [With Change Notice 1]",
           Federal Information Processing Standards
           Publication 186-2, January 2000.

[FIPS186-3]
          National Institute of Standards and Technology (NIST),
          "Digital Signature Standard (DSS)", Federal Information
          Processing Standards Publication 186-3, June 2009.

[I-D.ietf-lamps-eai-addresses]
          Melnikov, A. and W. Chuang, "Internationalized Email
          Addresses in X.509 certificates", draft-ietf-lamps-eai-
          addresses-18 (work in progress), March 2018.

[I-D.ietf-lamps-rfc5751-bis]
          Schaad, J., Ramsdell, B., and S. Turner, "Secure/
          Multipurpose Internet Mail Extensions (S/MIME) Version 4.0
          Message Specification", draft-ietf-lamps-rfc5751-bis-10
          (work in progress), June 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119,
          DOI 10.17487/RFC2119, March 1997,
          <https://www.rfc-editor.org/info/rfc2119>.

[RFC2634] Hoffman, P., Ed., "Enhanced Security Services for S/MIME",
          RFC 2634, DOI 10.17487/RFC2634, June 1999,
          <https://www.rfc-editor.org/info/rfc2634>.

[RFC2985] Nystrom, M. and B. Kaliski, "PKCS #9: Selected Object
          Classes and Attribute Types Version 2.0", RFC 2985,
          DOI 10.17487/RFC2985, November 2000,
          <https://www.rfc-editor.org/info/rfc2985>.

[RFC3279] Bassham, L., Polk, W., and R. Housley, "Algorithms and
          Identifiers for the Internet X.509 Public Key
          Infrastructure Certificate and Certificate Revocation List
          (CRL) Profile", RFC 3279, DOI 10.17487/RFC3279, April
          2002, <https://www.rfc-editor.org/info/rfc3279>.

[RFC3447] Jonsson, J. and B. Kaliski, "Public-Key Cryptography
          Standards (PKCS) #1: RSA Cryptography Specifications
          Version 2.1", RFC 3447, DOI 10.17487/RFC3447, February
          2003, <https://www.rfc-editor.org/info/rfc3447>.

[RFC4055] Schaad, J., Kaliski, B., and R. Housley, "Additional
          Algorithms and Identifiers for RSA Cryptography for use in
          the Internet X.509 Public Key Infrastructure Certificate
          and Certificate Revocation List (CRL) Profile", RFC 4055,
          DOI 10.17487/RFC4055, June 2005,
          <https://www.rfc-editor.org/info/rfc4055>.

[RFC4056] Schaad, J., "Use of the RSASSA-PSS Signature Algorithm in
           Cryptographic Message Syntax (CMS)", RFC 4056,
           DOI 10.17487/RFC4056, June 2005,
           <https://www.rfc-editor.org/info/rfc4056>.

[RFC5035] Schaad, J., "Enhanced Security Services (ESS) Update:
           Adding CertID Algorithm Agility", RFC 5035,
           DOI 10.17487/RFC5035, August 2007,
           <https://www.rfc-editor.org/info/rfc5035>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S.,
           Housley, R., and W. Polk, "Internet X.509 Public Key
           Infrastructure Certificate and Certificate Revocation List
           (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008,
           <https://www.rfc-editor.org/info/rfc5280>.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70,
           RFC 5652, DOI 10.17487/RFC5652, September 2009,
           <https://www.rfc-editor.org/info/rfc5652>.

[RFC5750] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet
           Mail Extensions (S/MIME) Version 3.2 Certificate
           Handling", RFC 5750, DOI 10.17487/RFC5750, January 2010,
           <https://www.rfc-editor.org/info/rfc5750>.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet
           Mail Extensions (S/MIME) Version 3.2 Message
           Specification", RFC 5751, DOI 10.17487/RFC5751, January
           2010, <https://www.rfc-editor.org/info/rfc5751>.

[RFC5755] Farrell, S., Housley, R., and S. Turner, "An Internet
           Attribute Certificate Profile for Authorization",
           RFC 5755, DOI 10.17487/RFC5755, January 2010,
           <https://www.rfc-editor.org/info/rfc5755>.

[RFC5758] Dang, Q., Santesson, S., Moriarty, K., Brown, D., and T.
           Polk, "Internet X.509 Public Key Infrastructure:
           Additional Algorithms and Identifiers for DSA and ECDSA",
           RFC 5758, DOI 10.17487/RFC5758, January 2010,
           <https://www.rfc-editor.org/info/rfc5758>.

[RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature
           Algorithm (DSA) and Elliptic Curve Digital Signature
           Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August
           2013, <https://www.rfc-editor.org/info/rfc6979>.

   [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC
              2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174,
              May 2017, <https://www.rfc-editor.org/info/rfc8174>.

   [SMIMEv3.2]
              "S/MIME version 3.2".

              This group of documents represents S/MIME version 3.2.
              This set of documents are [RFC2634], [RFC5750], [[This
              Document]], [RFC5652], and [RFC5035].

   [SMIMEv4.0]
              "S/MIME version 4.0".

              This group of documents represents S/MIME version 4.0.
              This set of documents are [RFC2634],
              [I-D.ietf-lamps-rfc5751-bis], [[This Document]],
              [RFC5652], and [RFC5035].

   [X.680]   "Information Technology - Abstract Syntax Notation One
              (ASN.1): Specification of basic notation. ITU-T
              Recommendation X.680 (2002) | ISO/IEC 8824-1:2002.".

7.2. Informational References

   [ESS]   "Enhanced Security Services for S/ MIME".

              This is the set of documents dealing with enhanced
              security services and refers to [RFC2634] and [RFC5035].

   [I-D.ietf-curdle-pkix]
              Josefsson, S. and J. Schaad, "Algorithm Identifiers for
              Ed25519, Ed448, X25519 and X448 for use in the Internet
              X.509 Public Key Infrastructure", draft-ietf-curdle-
              pkix-10 (work in progress), May 2018.

   [PKCS6]   RSA Laboratories, "PKCS #6: Extended-Certificate Syntax
              Standard", November 1993.

   [RFC2311] Dusse, S., Hoffman, P., Ramsdell, B., Lundblade, L., and
              L. Repka, "S/MIME Version 2 Message Specification",
              RFC 2311, DOI 10.17487/RFC2311, March 1998,
              <https://www.rfc-editor.org/info/rfc2311>.

   [RFC2312] Dusse, S., Hoffman, P., Ramsdell, B., and J. Weinstein,
              "S/MIME Version 2 Certificate Handling", RFC 2312,
              DOI 10.17487/RFC2312, March 1998,
              <https://www.rfc-editor.org/info/rfc2312>.

   [RFC2313] Kaliski, B., "PKCS #1: RSA Encryption Version 1.5",
              RFC 2313, DOI 10.17487/RFC2313, March 1998,
              <https://www.rfc-editor.org/info/rfc2313>.

   [RFC2314] Kaliski, B., "PKCS #10: Certification Request Syntax
              Version 1.5", RFC 2314, DOI 10.17487/RFC2314, March 1998,
              <https://www.rfc-editor.org/info/rfc2314>.

   [RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax
              Version 1.5", RFC 2315, DOI 10.17487/RFC2315, March 1998,
              <https://www.rfc-editor.org/info/rfc2315>.

   [RFC2630] Housley, R., "Cryptographic Message Syntax", RFC 2630,
              DOI 10.17487/RFC2630, June 1999,
              <https://www.rfc-editor.org/info/rfc2630>.

   [RFC2631] Rescorla, E., "Diffie-Hellman Key Agreement Method",
              RFC 2631, DOI 10.17487/RFC2631, June 1999,
              <https://www.rfc-editor.org/info/rfc2631>.

   [RFC2632] Ramsdell, B., Ed., "S/MIME Version 3 Certificate
              Handling", RFC 2632, DOI 10.17487/RFC2632, June 1999,
              <https://www.rfc-editor.org/info/rfc2632>.

   [RFC2633] Ramsdell, B., Ed., "S/MIME Version 3 Message
              Specification", RFC 2633, DOI 10.17487/RFC2633, June 1999,
              <https://www.rfc-editor.org/info/rfc2633>.

   [RFC3114] Nicolls, W., "Implementing Company Classification Policy
              with the S/MIME Security Label", RFC 3114,
              DOI 10.17487/RFC3114, May 2002,
              <https://www.rfc-editor.org/info/rfc3114>.

   [RFC3850] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail
              Extensions (S/MIME) Version 3.1 Certificate Handling",
              RFC 3850, DOI 10.17487/RFC3850, July 2004,
              <https://www.rfc-editor.org/info/rfc3850>.

   [RFC3851] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail
              Extensions (S/MIME) Version 3.1 Message Specification",
              RFC 3851, DOI 10.17487/RFC3851, July 2004,
              <https://www.rfc-editor.org/info/rfc3851>.

   [RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)",
              RFC 3852, DOI 10.17487/RFC3852, July 2004,
              <https://www.rfc-editor.org/info/rfc3852>.

[RFC6090]   McGrew, D., Igoe, K., and M. Salter, "Fundamental Elliptic
            Curve Cryptography Algorithms", RFC 6090,
            DOI 10.17487/RFC6090, February 2011,
            <https://www.rfc-editor.org/info/rfc6090>.

[RFC6151]   Turner, S. and L. Chen, "Updated Security Considerations
            for the MD5 Message-Digest and the HMAC-MD5 Algorithms",
            RFC 6151, DOI 10.17487/RFC6151, March 2011,
            <https://www.rfc-editor.org/info/rfc6151>.

[RFC6194]   Polk, T., Chen, L., Turner, S., and P. Hoffman, "Security
            Considerations for the SHA-0 and SHA-1 Message-Digest
            Algorithms", RFC 6194, DOI 10.17487/RFC6194, March 2011,
            <https://www.rfc-editor.org/info/rfc6194>.

[RFC8032]   Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital
            Signature Algorithm (EdDSA)", RFC 8032,
            DOI 10.17487/RFC8032, January 2017,
            <https://www.rfc-editor.org/info/rfc8032>.

[RFC8162]   Hoffman, P. and J. Schlyter, "Using Secure DNS to
            Associate Certificates with Domain Names for S/MIME",
            RFC 8162, DOI 10.17487/RFC8162, May 2017,
            <https://www.rfc-editor.org/info/rfc8162>.

[SMIMEv2]   "S/MIME version v2".

            This group of documents represents S/MIME version 2. This
            set of documents are [RFC2311], [RFC2312], [RFC2313],
            [RFC2314], and [RFC2315].

[SMIMEv3]   "S/MIME version 3".

            This group of documents represents S/MIME version 3. This
            set of documents are [RFC2630], [RFC2631], [RFC2632],
            [RFC2633], [RFC2634], and [RFC5035].

[SMIMEv3.1]
            "S/MIME version 3.1".

            This group of documents represents S/MIME version 3.1.
            This set of documents are [RFC2634], [RFC3850], [RFC3851],
            [RFC3852], and [RFC5035].

[SP800-57]
            National Institute of Standards and Technology (NIST),
            "Special Publication 800-57: Recommendation for Key
            Management", August 2005.

   [X.500]   "ITU-T Recommendation X.500 (1997) | ISO/IEC 9594- 1:1997,
                Information technology - Open Systems Interconnection -
                The Directory: Overview of concepts, models and
                services.".

Appendix A. Historic Considerations

A.1. Signature Algorithms and Key Sizes

   There are a number of problems with validating certificates on
   sufficiently historic messages. For this reason it is strongly
   suggested that UAs treat these certificates differently from those on
   current messages.   These problems include:

   - CAs are not required to keep certificates on a CRL beyond one
      update after a certificate has expired. This means that unless
      CRLs are cached as part of the message it is not always possible
      to check if a certificate has been revoked. The same problems
      exist with OCSP responses as they may be based on a CRL rather
      than on the certificate database.

   - RSA and DSA keys of less than 2048 bits are now considered by many
      experts to be cryptographically insecure (due to advances in
      computing power). Such keys were previously considered secure, so
      processing of historic certificates will often result in the use
      of weak keys. Implementations that wish to support previous
      versions of S/MIME or process old messages need to consider the
      security risks that result from smaller key sizes (e.g., spoofed
      messages) versus the costs of denial of service.

      [SMIMEv3.1] set the lower limit on suggested key sizes for
      creating and validation at 1024 bits. Prior to that the lower
      bound on key sizes was 512 bits.

   - Hash functions used to validate signatures on historic messages
      may longer be considered to be secure (see below). While there
      are not currently any known practical pre-image or second pre-
      image attacks against MD5 or SHA-1, the fact they are no longer
      considered to be collision resistant implies that the security
      level of any signature that is created with that these hash
      algorithms should also be considered as suspect.

   The following algorithms have been called out for some level of
   support by previous S/MIME specifications:

   - RSA with MD5 was dropped in [SMIMEv4.0]. MD5 is no longer
      considered to be secure as it is no longer collision-resistant.
      Details can be found in [RFC6151].

- RSA and DSA with SHA-1 were dropped in [SMIMEv4.0]. SHA-1 is no longer considered to be secure as it is no longer collision-resistant. The IETF statement on SHA-1 can be found in [RFC6194] but it is out-of-date relative to the most recent advances.

- DSA with SHA-256 support was dropped in [SMIMEv4.0]. DSA was dropped as part of a general movement from finite fields to elliptic curves. Issues have come up dealing with non-deterministic generation of the parameter 'k' (see [RFC6979]).

For 512-bit RSA with SHA-1 see [RFC3279] and [FIPS186-2] without Change Notice 1, for 512-bit RSA with SHA-256 see [RFC4055] and [FIPS186-2] without Change Notice 1.

For 512-bit DSA with SHA-1 see [RFC3279] and [FIPS186-2] without Change Notice 1, for 512-bit DSA with SHA-256 see [RFC5758] and [FIPS186-2] without Change Notice 1, for 1024-bit DSA with SHA-1 see [RFC3279] and [FIPS186-2] with Change Notice 1, for 1024-bit through 3072 DSA with SHA-256 see [RFC5758] and [FIPS186-3]. In either case, the first reference provides the signature algorithm's object identifier and the second provides the signature algorithm's definition.

Appendix B. Moving S/MIME v2 Certificate Handling to Historic Status

The S/MIME v3 [SMIMEv3], v3.1 [SMIMEv3.1], v3.2 [SMIMEv3.2], and v4.0 (this document) are backward compatible with the S/MIME v2 Certificate Handling Specification [SMIMEv2], with the exception of the algorithms (dropped RC2/40 requirement and added DSA and RSASSA-PSS requirements). Therefore, it is recommended that RFC 2312 [SMIMEv2] be moved to Historic status.

Appendix C. Acknowledgments

Many thanks go out to the other authors of the S/MIME v2 RFC: Steve Dusse, Paul Hoffman, and Jeff Weinstein. Without v2, there wouldn't be a v3, v3.1, v3.2 or v4.0.

A number of the members of the S/MIME Working Group have also worked very hard and contributed to this document. Any list of people is doomed to omission, and for that I apologize. In alphabetical order, the following people stand out in my mind because they made direct contributions to this document.

Bill Flanigan, Trevor Freeman, Elliott Ginsburg, Alfred Hoenes, Paul Hoffman, Russ Housley, David P. Kemp, Michael Myers, John Pawling, and Denis Pinkas.

The version 4 update to the S/MIME documents was done under the
auspices of the LAMPS Working Group.

Authors' Addresses

Jim Schaad
August Cellars

Email: ietf@augustcellars.com

Blake Ramsdell
Brute Squad Labs, Inc.

Email: blaker@gmail.com

Sean Turner
sn3rd

Email: sean@sn3rd.com