Internet Draft
draft-ietf-smime-rfc2632bis-07.txt
June 4, 2004
Expires December 4, 2004

Editor: Blake Ramsdell,
Sendmail, Inc.

S/MIME Version 3.1 Certificate Handling

Status of this memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Abstract

This document specifies conventions for X.509 certificate usage by S/MIME (Secure/Multipurpose Internet Mail Extensions) 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 3280, 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 3280.

1. Overview

S/MIME (Secure/Multipurpose Internet Mail Extensions), described in [SMIME-MSG], 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 [KEYM]. S/MIME agents MUST meet the certificate processing

requirements documented in this document in addition to those stated in [KEYM].

This specification is compatible with the Cryptographic Message Syntax [CMS] 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.

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.208.

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 [ACAUTH].

BER: Basic Encoding Rules for ASN.1, as defined in ITU-T X.209.

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 [KEYM]. 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 [KEYM]. The CRL is signed by the issuer. The type intended by this specification is the one defined in [KEYM].

DER: Distinguished Encoding Rules for ASN.1, as defined in ITU-T X.690.

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 Compatibility with Prior Practice of S/MIME

S/MIME version 3.1 agents should 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 and S/MIME version 3 is described in RFC 2630 through RFC 2634 inclusive. RFC 2311 also has historical information about the development of S/MIME.

1.3 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [MUSTSHOULD].

1.4 Changes Since S/MIME v3 (RFC 2632)

Version 1 and Version 2 CRLs MUST be supported.

Multiple 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 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 the digitalSignature or nonRepudiation bit set.

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

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 [SMIME-MSG].

2.1 CertificateRevocationLists

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

All agents MUST be capable of performing revocation checks using CRLs as specified in [KEYM]. All agents MUST perform revocation status checking in accordance with [KEYM]. 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 CertificateChoices

Receiving agents MUST support v1 X.509 and v3 X.509 identity certificates as profiled in [KEYM]. 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 [ACAUTH] 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.

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 using a database or directory lookup scheme.

A sending agent SHOULD include at least one chain of certificates up to, but not including, a Certificate 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, certificates which can be considered the "root" of other chains, and which MAY be self-signed. 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 DSA public keys the parameters may be located in the root certificate. This would require that the recipient possess both the end-entity certificate as well as 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 [SECLABEL].

3. Using Distinguished Names for Internet Mail

End-entity certificates MAY contain an Internet mail address as described in [RFC-2822]. The address must be an "addr-spec" as defined in Section 3.4.1 of that specification. 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. Receiving agents MUST recognize email addresses in the subjectAltName field. Receiving agents MUST recognize email addresses in the Distinguished Name field in the PKCS #9 [PKCS9] 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.

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, if present, 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, which may be to display a message that shows the recipient the 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 identity certificates, except that the subject 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

A receiving agent needs 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. Another method under consideration by the IETF is to provide certificate retrieval services as part of the existing Domain Name System (DNS). Until such mechanisms are widely used, their utility may be limited by the small number of 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 that 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 a "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 may combine two or more mechanisms to allow the greatest flexibility and utility to the user. For instance, a secure Internet mail agent may resort to checking a centralized certificate retrieval mechanism for a certificate if it can not 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 [SMIME-MSG].

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 away from incoming messages. A receiving agent SHOULD have access to some certificate revocation list (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 [KEYM]. All agents MUST perform revocation status checking in accordance with [KEYM]. Receiving agents MUST recognize CRLs in received S/MIME messages.

4.2 Certification 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 [KEYM] 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 (ex: RSA); or forming a pairwise symmetric key (ex: 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.

4.3 Certificate and CRL Signing Algorithms

Certificates and Certificate Revocation Lists (CRLs) are signed by the certificate issuer. A receiving agent MUST be capable of verifying the signatures on certificates and CRLs made with id-dsa-with-sha1 [CMSALG].

A receiving agent MUST be capable of verifying the signatures on certificates and CRLs made with md5WithRSAEncryption and sha1WithRSAEncryption signature algorithms with key sizes from 512 bits to 2048 bits described in [CMSALG].

Because of the security issues surrounding MD2 [RC95], and in light of current use, md2WithRSAEncryption MAY be supported.

4.4 PKIX Certificate Extensions

PKIX describes an extensible framework in which the basic certificate information can be extended and how such extensions can be used to

control the process of issuing and validating certificates. The PKIX Working Group has ongoing efforts to identify and create extensions which 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 which have the greatest value in the S/MIME environment. The syntax and semantics of all the identified extensions are defined in [KEYM].

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 [KEYM], unless otherwise specified here.

4.4.1 Basic Constraints Certificate Extension

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

For example, certificates issued to CAs and subordinate CAs contain a basic constraint extension that identifies them as issuing authority certificates. End-entity certificates contain an extension that constrains the certificate from being an issuing authority certificate.

Certificates SHOULD 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 which contain a key usage extension which 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 which contains the key usage extension without either the digitalSignature or nonRepudiation bit 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 the digitalSignature and nonRepudiation bits are set.

4.4.3 Subject Alternative Name Extension

The subject alternative name extension is used in S/MIME as the preferred means to convey the RFC-2822 email address(es) that correspond to the entity for this certificate. Any RFC-2822 email addresses present MUST be encoded using the rfc822Name CHOICE of the GeneralName type. Since the SubjectAltName type is a SEQUENCE OF GeneralName, multiple RFC-2822 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 which 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 it contains

either the emailProtection or the anyExtendedKeyUsage OID as defined in [KEYM]. 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. 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 has 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 matches 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.

At the Selected Areas in Cryptography '95 conference in May 1995, Rogier and Chauvaud presented an attack on MD2 that can nearly find collisions [RC95]. Collisions occur when one can find two different messages that generate the same message digest. A checksum operation in MD2 is the only remaining obstacle to the success of the attack. For this reason, the use of MD2 for new applications is discouraged. It is still reasonable to use MD2 to verify existing signatures, as

the ability to find collisions in MD2 does not enable an attacker to find new messages having a previously computed hash value.

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 recent 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. Unless it is from a trusted agent, this time MUST NOT be the SigningTime attribute found in an S/MIME message. For most sending agents, 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.

A. Normative References

[ACAUTH] "An Internet Attribute Certificate Profile for Authorization", RFC 3281

[CMS] "Cryptographic Message Syntax (CMS)", draft-ietf-smime-3369bis-04

[CMSALG] "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370

[KEYM] "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", RFC 3280

[KEYMALG] "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile ", RFC 3279

[MUSTSHOULD] "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119

[PKCS9] "PKCS #9: Selected Object Classes and Attribute Types Version 2.0", RFC 2985

[RFC-2822], "Internet Message Format", RFC 2822

[SMIME-MSG] "S/MIME Version 3 Message Specification ", Internet Draft draft-ietf-smime-msg

B. Informative References

[CERTV2] "S/MIME Version 2 Certificate Handling", RFC 2312

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

[RC95] Rogier, N. and Chauvaud, P., "The compression function of MD2 is not collision free," Presented at Selected Areas in Cryptography '95, May 1995

[SECLABEL] "Implementing Company Classification Policy with the S/MIME Security Label", RFC 3114

[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

[X.501] ITU-T Recommendation X.501 (1997) | ISO/IEC 9594-2:1997, Information technology - Open Systems Interconnection - The Directory: Models

[X.509] ITU-T Recommendation X.509 (1997) | ISO/IEC 9594-8:1997, Information technology - Open Systems Interconnection - The Directory: Authentication framework

[X.520] ITU-T Recommendation X.520 (1997) | ISO/IEC 9594-6:1997, Information technology - Open Systems Interconnection - The Directory: Selected attribute types.

C. Acknowledgements

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.

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 due to the fact that they made direct contributions to this document.

Bill Flanigan
Trevor Freeman
Elliott Ginsburg
Paul Hoffman
Russ Housley
David P. Kemp
Michael Myers
John Pawling
Denis Pinkas
Jim Schaad

D. Editor's address

Blake Ramsdell
Sendmail, Inc.
704 228th Ave NE #775
Sammamish, WA 98074

blake@sendmail.com