Domain Security Services using S/MIME

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 describes how the S/MIME protocol can be processed and generated by a number of components of a communication system, such as message transfer agents, guards and gateways to deliver security services. These services are collectively referred to as ‘Domain Security Services’. The mechanisms described in this document are designed to solve a number of interoperability problems and technical limitations that arise when different security domains wish to communicate securely - for example when two domains use incompatible messaging technologies such as X.400 and SMTP/MIME. This document is also applicable to organisations and enterprises that have internal PKIs which are not accessible by the outside world, but wish to interoperate securely using the S/MIME protocol.

This draft is being discussed on the ’ietf-smime’ mailing list. To subscribe, send a message to:
   ietf-smime-request@imc.org
with the single word
   subscribe
in the body of the message. There is a Web site for the mailing list at <http://www.imc.org/ietf-smime/>.

Acknowledgements

Significant comments were made by Trevor Freeman, Russ Housley, Dave Kemp, Jim Schaad, Greg Colla and Michael Zolotarev.

1. Introduction

The S/MIME [1] series of standards define a data encapsulation format for the provision of a number of security services including data integrity, confidentiality, and authentication. S/MIME is designed for use by messaging clients to deliver security services to distributed messaging applications.
There are many circumstances when it is not desirable or practical to provide end-to-end (desktop-to-desktop) security services, particularly between different security domains. An organisation that is considering providing end-to-end security services will typically have to deal with some if not all of the following issues:

1) Heterogeneous Message Access Methods: Users are accessing mail using mechanisms which re-format messages, such as using Web browsers. Message reformatting in the Message Store makes end-to-end encryption and signature validation impossible.

2) Message screening and audit: Server-based mechanisms such as searching for prohibited words or other content, virus scanning, and audit, are incompatible with end-to-end encryption.

3) PKI deployment issues: There may not be any certificate paths between two organisations. Or an organisation may be sensitive about aspects of its PKI and unwilling to expose them to outside access. For either of these reasons, direct end-to-end signature validation and encryption are impossible.

4) Heterogeneous Message transports: One organisation using X.400 wishes to communicate with another using SMTP. Message reformatting at gateways makes end-to-end encryption and signature validation impossible.

This document describes an approach to solving these problems by providing message security services at the level of a domain or an organisation. This document specifies how these ‘domain security services’ can be provided using the S/MIME protocol. Domain security services may replace or complement mechanisms at the desktop. For example, a domain may decide to provide desktop-to-desktop signatures but domain-to-domain encryption services. Or it may allow desktop-to-desktop services for intra-domain use, but enforce domain-based services for communication with other domains.

Whether or not a domain based service is inherently better or worse than desktop based solutions is an open question. Some experts believe that only end-to-end solutions can be truly made secure, while others believe that the benefits offered by such things as content checking at domain boundaries offers considerable increase in practical security for many real systems. The additional service of allowing signature checking at several points on a communications path is also an extra benefit in many situations. This debate is outside the scope of this document. What is offered here is a set of tools that integrators can tailor in different ways to meet different needs in different circumstances.

Message transfer agents (MTAs), guards, firewalls and protocol translation gateways all provide domain security services. As with desktop based solutions, these components must be resilient against a wide variety of attacks intended to subvert the security services. Therefore, careful consideration should be given to security of these components, to make sure that their siting and configuration minimises the possibility of attack.

Throughout this draft the terms MAY, MUST, MUST NOT and SHOULD are used in capital letters. This conforms to the definitions in [2].

2. Overview of Domain Security Services
This section gives an informal overview of the security services that are provided by S/MIME between different security domains. These services are provided by a combination of mechanisms in the sender’s and recipient’s domains.

Later sections describe definitively how these services map onto elements of the S/MIME protocol.

The following security mechanisms are specified in this document:

1. Domain signature
2. Review signature
3. Additional attributes signature
4. Domain encryption and decryption

The term ‘security domain’ as used in this document is defined as a collection of hardware and personnel operating under a single security authority and performing a common business function. Members of a security domain will of necessity share a high degree of mutual trust, due to their shared aims and objectives.

A security domain is typically protected from direct outside attack by physical measures and from indirect (electronic) attack by a combination of firewalls and guards at network boundaries. The interface between two security domains is termed a ‘security boundary’. One example of a security domain is an organisational network ('Intranet').

2.1 Domain Signature

A Domain signature is an S/MIME signature generated on behalf of a set of users in a domain. A Domain signature can be used to authenticate information sent between domains, for example, when two 'Intranets' are connected using the Internet. It can be used when two domains employ incompatible signature schemes internally or when there are no certification links between their PKIs. In both cases messages from the originator’s domain are signed over the original message and signature (if present) using an algorithm, key, and certificate which can be processed by the recipient(s). A domain signature is sometimes referred to as an "organisational signature".

2.2 Review Signature

A third party may review messages before they are forwarded to the final recipient(s) who may be in the same or a different security domain. Organisational policy and good security practice often require that messages be reviewed before they are released to external recipients. Having reviewed a message, an S/MIME signature is added to it - a review signature. An agent MAY check the review signature at the domain boundary, to ensure that only reviewed messages are released.

2.3 Additional Attributes Signature

A third party MAY add additional attributes to a signed message. An S/MIME signature is used for this purpose - an additional attributes signature. An example of an additional attribute is the ‘Equivalent Label’ attribute defined in ESS [3].

2.4 Domain Encryption and Decryption

Domain encryption is S/MIME encryption performed on behalf of a collection of users in a domain. Domain encryption can be used to
protect information between domains, for example, when two ‘Intranets’ are connected using the Internet. It can also be used when end users do not have encryption capabilities at the desktop, or when two domains employ incompatible encryption schemes internally. In the latter case messages from the originator’s domain are re-encrypted using an algorithm, key, and certificate which can be decrypted by the recipient(s) or an entity in their domain.

3. Mapping of the Signature Services to the S/MIME Protocol

This section describes the S/MIME Protocol elements that are used to provide the security services described above. ESS [3] introduces the concept of triple-wrapped messages that are first signed, then encrypted, then signed again. This document also uses this concept of triple-wrapping. In addition, this document also uses the concept of ‘signature encapsulation’. ‘Signature encapsulation’ denotes a complete signed message that is wrapped in a second signature, the second signature covering both the content and the first (inner) signature.

Signature Encapsulation MAY be performed on the inner and/or the outer signature of a triple-wrapped message. The term ‘parallel signatures’ means two or more signatures calculated over the same content. This capability is described in CMS [3], where a set of one or more SignerInfos can be attached to signed data.

For example, the originator signs a message which is then encapsulated with an ‘additional attributes’ signature. This is then encrypted. A reviewer then signs this encrypted data, which is then encapsulated by a domain signature.

3.1 Naming conventions and Signature Types

An entity receiving an S/MIME signed message would normally expect the signature to be that of the originator of the message. However, the message security services defined in this draft require the recipient to be able accept messages signed by other entities and the originator. When other entities sign the message the name in the certificate will not match the message senders name. An S/MIME implementation would flag an error if there were a mismatch between the name in the certificate and the message sender’s name. (This check prevents a number of types of masquerade attack.)

To resolve this incompatibility, this document defines a naming convention that specifies the form that the signing agents name SHOULD take. Adherence to this naming convention avoids the problems of uncontrolled naming and the possible masquerade attacks that this would produce.

As an assistance to implementation, a signed attribute is defined to be included in the S/MIME signature – the ‘signature type’ attribute. On receiving a message containing this attribute, the naming convention checks are invoked.

Implementations conforming to this standard MUST support the naming convention for signature generation and verification. Implementations conforming to this standard MUST recognise the signature type attribute for signature verification. Implementations conforming to this standard SHOULD support the signature type attribute for signature generation; however, this is not mandated.

3.1.1 Naming conventions
The following naming conventions are specified for agents generating signatures specified in this document:

* For a domain signature, an agent generating this signature MUST be named 'domain-signing-authority'.

* For a review signature, an agent generating this signature MUST be named 'review-authority'.

* For an additional attributes signature, an agent generating this signature MUST be named 'attribute-authority'.

This name shall appear in the ‘common name (CN)’ component of the subject field in the X.509 certificate. Additionally, if the certificate contains an SMTP e-mail address, this name shall appear in the end entity component of the address – on the left-hand side of the '@' symbol.

In the case of a domain signature, an additional naming rule is defined: the ‘name mapping rule’. The name mapping rule states that for a domain signing authority, the domain component of its name MUST be the same as, or an ascendant of, the domain name of the message originator(s) that it is representing. The domain component is defined as follows:

* In the case of an X.500 distinguished subject name of an X.509 certificate, the domain component is the country, organisation, organisational unit, state, and locality components of the distinguished name.

* If the certificate contains an SMTP e-mail address, the domain component is defined to be the SMTP address component on the right-hand side of the '@' symbol.

For example, a domain signing authority acting on behalf of John Doe of the Acme corporation, whose distinguished name is ‘cn=John Doe, ou=marketing, o=acme, c=us’ and whose e-mail address is John.Doe@marketing.acme.com, could have a certificate containing a distinguished name of ‘cn=domain-signing-authority, o=acme, c=us’ and an e-mail address of ‘domain-signing-authority@acme.com’.

Any message received where the domain component of the domain signing agents name does not match, or is not an ascendant of, the originator’s domain name MUST be rejected.

This naming rule prevents agents from one organisation masquerading as domain signing authorities on behalf of another. For the other types of signature defined in this document, no such named mapping rule is defined.

Implementations conforming to this standard MUST support this name mapping convention as a minimum. Implementations MAY choose to supplement this convention with other locally defined conventions. However, these MUST be agreed between sender and recipient domains prior to secure exchange of messages.

On verifying the signature, a receiving agent MUST ensure that the naming convention has been adhered to. Any message that violates the convention shall be rejected as invalid.
3.1.2 Signature type attribute

An S/MIME authenticated attribute is also used to indicate the type of signature. This should be used in conjunction with the naming conventions specified in the previous section. When an S/MIME signed message containing the signature type attribute is received it triggers the software to verify that the correct naming convention has been used.

The ASN.1 [4] notation of this attribute is:

\[
\text{SignatureType} ::= \text{SEQUENCE OF OBJECT IDENTIFIER} \\
\text{id-signatureType OBJECT IDENTIFIER ::= \{ iso (1) member-body (2) us (840) rsadsi (113549) <TBD>\}}
\]

If present, the SignatureType attribute MUST be an authenticated attribute, as defined in [5]. If the SignatureType attribute is absent the recipient SHOULD assume that the signature is that of the message originator.

Each of the signature types defined here are generated and processed exactly as described in [5]. They are distinguished by the presence of the following values in the SignatureType authenticated attribute:

- id-sigtype-domain-sig OBJECT IDENTIFIER ::= \{ id-signatureType 2 \} for a domain signature.
- id-sigtype-add-attrib-sig OBJECT IDENTIFIER ::= \{ id-signatureType 3 \} for an additional attributes signature.
- id-sigtype-review OBJECT IDENTIFIER ::= \{ id-signatureType 4 \} for a review signature.

For completeness, an attribute type is also specified for an originator signature. However, this signature type is optional. It is defined as follows:

- id-sigtype-originator-sig OBJECT IDENTIFIER ::= \{ id-signatureType 1 \} for an originator’s signature.

The originator signature MUST NOT encapsulate other signatures. The other signature types specified in this document MAY encapsulate other signatures. All the signature types MAY be added in parallel to other signatures as documented in [5].

A SignerInfo MUST NOT include multiple instances of SignatureType. An authenticated attribute representing a SignatureType MAY include multiple instances of different SignatureType values as an AttributeValue of attrValues [5], as long as the SignatureType ‘additional attributes’ is not present.

The following sections describe the conditions under which each of these types of signature may be generated, and how they are processed.

3.2 Domain Signature Generation and Verification

A ‘domain signature’ is a proxy signature generated on a user’s behalf in the user’s domain. The signature MUST adhere to the naming conventions in 3.1.1, including the name mapping convention. A ‘domain signature’ on a message authenticates the fact that the message has originated in that domain. Before signing, a process generating a
‘domain signature’ MUST first satisfy itself of the authenticity of the message originator. This is achieved by one of two methods. Either the ‘originator’s signature’ is checked, if S/MIME signatures are used inside a domain. Or if not, some mechanism external to S/MIME is used, such as the physical address of the originating client or an authenticated IP link.

If the originator’s authenticity is successfully verified by one of the above methods and all other signatures present are valid, a ‘domain signature’ MAY be added to a message in one of the following ways:

1) An unsigned message has a null signature added to it (i.e. the message is wrapped in a signedData that has no signerInfo attached), and then a ‘domain signature’ is added as defined in methods 2) or 3) below. The originator’s information is included as part of a header field in the encapsulated message.

2) Signature Encapsulation is used to wrap the original signed message with a ‘domain signature’.

3) The original signed message has a ‘domain signature’ added in parallel.

An entity generating a domain signature MUST do so using a certificate containing a subject name that follows the naming convention specified in 3.1.1.

When a ‘domain signature’ is applied the mlExpansionHistory and eSSSecurityLabel attributes MUST be copied from other signerInfos as stated in [3].

If the originator’s authenticity is not successfully verified or all the signatures present are not valid, a ‘domain signature’ MUST NOT be generated.

On reception, the ‘domain signature’ SHOULD be used to verify the authenticity of a message. A check MUST be made to ensure that both the naming convention and the name mapping convention have been used as specified in this standard.

A recipient MAY assume that successful verification of the domain signature also authenticates the message originator.

If there is an originator signature present, the name in that certificate SHOULD be used to identify the originator. This information can then be displayed to the recipient.

Alternatively, if a ‘domain signature’ has encapsulated a complete MIME-encoded message, the originator information (SMTP ‘From’ field) contained within it denotes the originator of the message.

If neither of these cases is true no assumptions can be made about the originator.

A domain signer can be assumed to have verified any signatures that it encapsulates. Therefore, it is not necessary to verify these signatures before treating the message as authentic. However, this standard does not preclude a recipient from attempting to verify any other signatures that are present.

The ‘domain signature’ is indicated by the presence of the value
Id-at-sigtype-domain-sig in a ‘signature type’ authenticated attribute.

There MAY be multiple ‘domain signature’ signatures in an S/MIME encoding.

3.3 Additional Attributes Signature generation and verification

The ‘additional attributes’ signature type indicates that the SignerInfo contains additional attributes that are associated with the message.

All attributes in the applicable SignerInfo MUST be treated as additional attributes. Successful verification of an ‘additional attributes’ signature means only that the attributes are authentically bound to the message. A recipient MUST NOT assume that its successful verification also authenticates the message originator.

An entity generating an ‘additional attributes’ signature MUST do so using a certificate containing a subject name that follows the naming convention specified in 3.1.1. On reception, a check MUST be made to ensure that the naming convention has been used.

A signer MAY include any of the attributes listed in [5] or in this document when generating an ‘additional attributes’ signature. The following attributes have a special meaning, when present in an ‘additional attributes’ signature:

1) Equivalent Label: label values in this attribute are to be treated as equivalent to the security label contained in an encapsulated SignerInfo, if present.

2) Security Label: the label value indicates the aggregate sensitivity of the inner message content plus any encapsulated signedData and envelopedData containers. The label on the original data is indicated by the value in the originator’s signature, if present.

An ‘additional attributes’ signature is indicated by the presence of the value Id-at-sigtype-add-attrib-sig in a ‘signature type’ authenticated attribute. No other Object Identifiers MAY be included in the sequence of OIDs if this value is present. An ‘additional attributes’ signature MAY be added in parallel with other signatures in a SET OF SignerInfos.

There MAY be multiple ‘additional attributes’ signatures in an S/MIME encoding.

3.4 Review Signature generation and verification

The review signature indicates that the signer has reviewed the message. Successful verification of a review signature means only that the signer has approved the message for onward transmission to the recipient(s). When the recipient is in another domain, a device on a domain boundary such as a Mail Guard or firewall may be configured to check review signatures. A recipient MUST NOT assume that its successful verification also authenticates the message originator.

An entity generating a signed review signature MUST do so using a certificate containing a subject name that follows the naming convention specified in 3.1.1. On reception, a check MUST be made to ensure that the naming convention has been used.

A review signature is indicated by the presence of the value
Id-at-sigtype-review-sig in a ‘signature type’ authenticated attribute.

There MAY be multiple review signatures in an S/MIME encoding.

3.5 Originator Signature

The ‘originator signature’ is used to indicate that the signer is the originator of the message and its contents. It is included in this document for completeness only. An originator signature is indicated either by the absence of the signature type attribute, or by the presence of the value id-sigtype-originator-sig in a ‘signature type’ authenticated attribute. There MUST be only one ‘originator signature’ signature present in an S/MIME encoding and it MUST be one of the inner most signatures.

4. Encryption and Decryption

Domain encryption is encryption performed by a third party on behalf of a set of originators in a domain. Domain decryption is decryption performed by a third party on behalf of a set of recipients in a domain.

Depending on security policy, messages may be encrypted for decryption by the final recipient and by a domain decryption agent in the originator’s and/or the recipient’s domain. This is achieved by generating a RecipientInfo for each type of agent that is transmitted along with the encrypted message.

The processes of domain encryption and decryption may be performed in combination, as shown below.

<table>
<thead>
<tr>
<th></th>
<th>Recipient Decryption</th>
<th>Domain Decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Originator Encryption</td>
<td>Case(a)</td>
<td>Case(c)</td>
</tr>
<tr>
<td>Domain Encryption</td>
<td>Case(b)</td>
<td>Case(d)</td>
</tr>
</tbody>
</table>

Case (a), encryption of messages by the originator for decryption by the final recipient(s), is described in CMS [5]. In Cases (b) and (d), encryption is performed not by the originator but by a third party in the sending domain. In Cases (c) and (d), decryption is performed not by the recipient(s) but by a third party in the destination domain.

A client implementation that conforms to this standard MUST support cases (a) and (c) for transmission, and cases (a) and (b) for reception.

A Domain Encryption implementation that conforms to this standard MUST support cases (b) and (d), for transmission, and cases (c) and (d) for reception.

The process of encryption and decryption is documented in CMS [5]. The only additional requirement introduced by domain encryption and decryption is for greater flexibility in the management of keys, as described in the following subsections. As with signatures, a naming convention and name mapping convention are used to locate the correct key.

The mechanisms described below are applicable both to key agreement and key transport systems, as documented in CMS [5]. The phrase ‘encryption key’ is used as a collective term to cover the key management keys used by both techniques.
4.1 Domain Encryption Naming Conventions

A domain encryption agent MUST be named ‘domain-confidentiality-authority’. Also a domain decryption agent MUST be named ‘domain-confidentiality-authority’. This name MUST appear in the ‘common name (CN)’ component of the subject field in the X.509 certificate. Additionally, if the certificate contains an SMTP e-mail address, this name MUST appear in the end entity component of the address - on the left-hand side of the ‘@’ symbol.

Along with this naming convention, an additional naming rule is defined: the ‘name mapping rule’. The name mapping rule states that for an encryption agent, the domain component of its name MUST be the same as, or an ascendant of, the domain name of the set of entities that it represents. The domain component is defined as follows:

* In the case of an X.500 distinguished name of an X.509 certificate, the domain component is the country, organisation, organisational unit, state, and locality components of the distinguished name.

* If the certificate contains an SMTP e-mail address, the domain component is defined to be the SMTP address component on the right-hand side of the ‘@’ symbol.

For example, an encryption authority acting on behalf of John Doe of the Acme corporation, whose distinguished name is ‘cn=John Doe,ou=marketing, o=acme,c=us’ and whose e-mail address is John.Doe@marketing.acme.com, could have a certificate containing a distinguished name of ‘cn=domain-confidentiality-authority, o=acme,c=us’ and an e-mail address of ‘domain-confidentiality-authority@acme.com’. The key associated with this certificate would be used for encrypting messages for John Doe.

Any message received where the domain component of the domain encrypting agents name does not match, or is not an ascendant of, the domain name of the entities it represents MUST be rejected.

This naming rule prevents messages being encrypted for the wrong domain decryption agent.

Implementations conforming to this standard MUST support this name mapping convention as a minimum. Implementations may choose to supplement this convention with other locally defined conventions. However, these MUST be agreed between sender and recipient domains prior to sending any messages.

4.2 Domain Encryption Key Management

Domain encryption is encryption performed by a third party on behalf of a set of originators in a domain. Domain Encryption is shown as cases (b) and (d) in the above table.

Domain encryption uses a domain-wide encryption key from the sender’s domain. Information about this key is conveyed to the recipient in the RecipientInfo field as specified in CMS [5]. A domain encryption agent MUST be named according to the naming convention specified in section 4.1. This is so that the same key can be used on reply to a domain-encrypted message.

The domain encryption agent extracts the recipients address from the message and uses this to obtain the recipients domain-confidentiality-
authority public key and/or the recipients public key. For example, the recipients address is used as an index for a directory search. The directory search MAY return the recipients certificate and/or a domain-confidentiality-authority attribute that contains the location of the recipient’s domain decrypting agents certificate. If the directory search returns no certificates then encryption can not be performed and the message MUST NOT be sent. If one or both certificates are available then the originator’s domain encrypting agent encrypts the message for the recipient and the recipient’s domain decrypting agent.

4.3 Domain Decryption Key Management

Domain decryption is decryption performed by a third party on behalf of a set of recipients in a domain.

Domain Decryption is shown as cases (c) and (d) in the above table. In these cases, the encryption process has used a domain-wide encryption key for the recipient(s)’ domain, that has been obtained by using the recipient’s address (See example in section 4.2).

5. Security Considerations

Implementations MUST protect all private keys. Compromise of the signer’s private key permits masquerade.

Similarly, compromise of the content-encryption key may result in disclosure of the encrypted content.

Compromise of key material is regarded as an even more serious issue for domain security services than for an S/MIME client. This is because compromise of the private key may in turn compromise the security of a whole domain. Therefore, great care should be used when considering its protection.

6. References


7. Authors’ Addresses

Tim Dean
DERA Malvern
St. Andrews Road
Malvern
Worcs
WR14 3PS

Phone: +44 (0) 1684 894239