Internet Security
Enhanced Security Services for S/MIME

Thomas Göttlicher

April 20, 2004

Contents

1 Introduction **3**

2 Technical **4**
2.1 Internet Layer 4
2.2 Compatibility 4
2.3 Triple Wrapping 5

3 Signed Receipts **7**

4 Security Labels **8**

5 Secure Mailing Lists **9**
5.1 Mail Loops 9
5.2 Receipts 10

6 Signed Certificates **11**
6.1 Attacks 11
6.1.1 Substitution Attack 11
6.1.2 Re-issue of Certificate Attack 12
6.1.3 Duplicate Certificate Authority Attack 12
6.2 Attack Responses 12
6.2.1 Response to the Substitution Attack 12
6.2.2 Response to the Re-issue of Certificate Attack 12
6.2.3 Response to the Duplicate Certificate Authority Attack 12

7 Security Considerations **13**
7.1 Signed Receipts 13
7.2 Security Labels 13
7.3 Mailing lists 14

Chapter 1

Introduction

S/MIME stands for Secure/Multipurpose Internet Mail Extensions.

S/MIME provides a consistent way to send and receive secure MIME data. Based on the popular Internet MIME standard, S/MIME provides the following cryptographic security services for electronic messaging applications: authentication, message integrity and non-repudiation of origin and privacy and data security.

S/MIME is a security enhanced version of the MIME internet email standard. Compared with PGP, the S/MIME format is more fixed.

S/MIME can be used by traditional mail user agents to add cryptographic security services to mail sent, and to interpret cryptographic security services in mail received.

Furthermore, S/MIME can be used in automated message transfer agents that use cryptographic security services that do not require any human intervention.

This paper deals with additional security services for S/MIME, which are described in RFC 2634 dealing with signed receipts, security labels, secure mailing lists and signing certificates and their depending techniques.

Chapter 2

Technical

2.1 Internet Layer

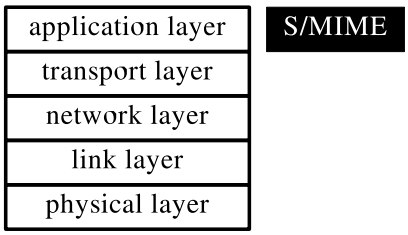

Figure 2.1: S/MIME's position at the internet stack

Figure 2.1 shows S/MIME's position at the internet stack. S/MIME is at the application layer. Users who want to use S/MIME and the enhanced security services for S/MIME only need an application that supports these features. There are is no need for changes or special requirements at the internet protocol.

2.2 Compatibility

Anyone who wants to use the enhanced security services for S/MIME, needs an email client, supporting S/MIME version 3. Users of the S/MIME version

3 are able to read message from a S/MIME version 2 sender, whereas users of the S/MIME version 2 cannot read messages from a S/MIME version sender.

2.3 Triple Wrapping

This section deals with a technique called "triple wrapping".

Triple wrapping is used when a message has to be signed, encrypted and signed again. Outer attributes may be added and signed by the message originator or intermediate agents e.g. a mail list agent. The inside signature is used for content integrity and the encryption provides privacy. The body contains confidential information including confidential attributes.

This is used by the S/MIME extensions which will be mentioned in the following chapters.

Steps for Triple Wrapping

These steps are needed to create a triple wrapped message:

1. Start with a message, called "original content".
2. Encapsulate the original content with the MIME Content-type headers.
3. Sign the result of step 2.
4. Add an MIME construct to the signed message from 3. The result is called the "inside signature".
5. Encrypt the result of step 4. This is called the "encrypted body".
6. Add the appropriate MIME headers.
7. Sign the result of step 6 as a single block.
8. Add the appropriate MIME headers. This is called the triple wrapped message.

Example

```
Content-type: multipart/signed;
   protocol="application/pkcs7-signature";
   boundary=outerboundary

--outerboundary
Content-type: application/pkcs7-mime;                    )
   smime-type=enveloped-data                             )
                                                         )
Content-type: multipart/signed;                        ! )
   protocol="application/pkcs7-signature";             ! )
   boundary=innerboundary                              ! )
                                                       ! )
--innerboundary                                        ! )
Content-type: text/plain                             % ! )
                                                     % ! )
Original content                                     % ! )
                                                       ! )
--innerboundary                                        ! )
Content-type: application/pkcs7-signature              ! )
                                                       ! )
inner SignedData block (eContent is missing)           ! )
                                                       ! )
--innerboundary--                                      ! )
--outerboundary
Content-type: application/pkcs7-signature

outer SignedData block (eContent is missing)

--outerboundary--
```

This example shows a triple wrapped message. The inner signature is computed over lines marked with a "%". Lines with a "!" are encrypted and the outer signature is computed over lines with a ")".

Chapter 3

Signed Receipts

Signed receipts are used to confirm the delivery of an email. The originator of the email will be informed by a message when the mail is received.

The sender can request receipts either from all recipients, from a specific list of recipients or from all recipients that did not receive the message as members of a mailing list.

Senders can indicate the receipts not only to be sent back to him but also to somebody else. It is likely that not even the sender gets a receipt.

When a receipt is requested, the receiving agent software should automatically create a receipt. The receiving agent must verify the signature of the message, before it processes a receipt-request. If the signature is invalid a receipt must not be sent.

In case that a receipt is requested more then once for the same message, each recipient should only return one signed receipt.

The receiving agent must not request a signed receipt for a signed receipt, because this can create infinity loops.

Chapter 4

Security Labels

Security Labels are a set of security information regarding the sensitivity of the content that is protected by the S/MIME encapsulation.

The security labels are used for access control. The receiving agent software examines the security labels and determines whether the recipient is allowed to see the contents of the message or not.

Security Labels must be signed attributes. Before processing a security label, the signature must be verified and the signature hast to be valid.

Basically these classifications are defined as: "unmarked", "unclassified", "restricted", "confidential", "secret" and "top-secret". Other values can be defined by any organization.

Equivalent-Labels

Since organizations have been allowed to define their own security policies, many different security policies exist. Cooperating organizations want to define equivalences between their different security policies. The sending agent can send a list of equivalent security levels, called Equivalent-Labels attribute.

The receiving agent has the option to process Equivalent-Labels attributes. It will only proceeds the EquivalentLables if it trusts the signer.

If the receiving agent understands the original security label, it has to ignore all Equivalent-Labels.

Chapter 5

Secure Mailing Lists

This chapter deals with secure mailing lists and their purposes. Sending agents must create recipient-specific data structures for each recipient of an encrypted message. When sending a message to a large number of recipients, the sending agent needs a lot resources.

Mail list agents can take a single message and perform the recipient-specific encryption.

5.1 Mail Loops

When dealing with mailing lists we have an issue called "mail loop": One mailing list is member of a second mailing list and reverse, the second is a member of the first. Mailing list agents have to prevent mail loops. Each time a mail list agent expands a message, it adds its own identifier to the message history. If the mailing list agent finds its own unique identifier in the history, a mail loop has happened. The mail list agent must not send the message to the list again. A human mail list administrator is to be notified, to solve the problem.

5.2 Receipts

When a message is send to a mailing list and a signed receipt is requested, the mailing list agent has to process the request by applying the local receipt policy.

So the mailing list agent often needs to propagate forward the receipt policy, and to do so it adds "insteadOf", "inAdditionTo", "none" to the history. Only the last recipient needs to process the receipt policy by reading the history.

If the originator has not requested a receipt, he must not get a receipt even if the mailing list says so.

The mailing list agent may inform the originator if he requested a receipt and but the mailing list agent substitutes the request.

Chapter 6

Signed Certificates

A signed certificate binds a name to a public key with a digital signature. At this sensitive part hackers might take the offensive. This chapter deals with different attacks and how to prevent them.

6.1 Attacks

6.1.1 Substitution Attack

The substitution attack is a simple substitution of one certificate for another. The issuer and serial number in the SignerInfo is modified to refer to a new certificate.

Version 1

Version 1 of the substitution attack is a simple DoS-Attack where an invalid certificate is substituted for the valid. As a result, the message is unverifiable, as the public key no longer matches the public key used to sign.

Version 2

The second substitutes a valid certificate for the original valid certificate where the public keys match. Thus, the message is validated under different constraints as the originator intended.

6.1.2 Re-issue of Certificate Attack

This attack deals with a certificate authority re-issuing a signing certificate. It may become more frequent as certificate authorities re-issue their own root certificates.

6.1.3 Duplicate Certificate Authority Attack

A rogue entity sets up a certificate authority that tries to duplicate the structure of an existing certificate authority and issues a new certificate with the same public key the signer uses.

6.2 Attack Responses

6.2.1 Response to the Substitution Attack

The denial of service attack cannot be prevented. There is no way to automatically identify the attack because it is indistinguishable from a message corruption. There is no practical way to prevent users from getting new certificates with the same public key.

6.2.2 Response to the Re-issue of Certificate Attack

A certificate authority should never re-issue a certificate with different attributes.

6.2.3 Response to the Duplicate Certificate Authority Attack

The only way to prevent the duplicate is never to trust a duplicated certificate authority!

Chapter 7

Security Considerations

The last chapter considers some security issues concerning signed receipts, security labels and mailing lists.

7.1 Signed Receipts

A recipient must not send back a reply if it cannot validate the signature. The reason for this is that an attacker could get information about the host and its software by reading the receipt.

Senders should encrypt receipts to prevent a passive attacker from gleaning information.

7.2 Security Labels

Senders must not rely on recipients' processing software to process security labels correctly. Some S/MIME clients may not understand security labels but display a labeled message.

For example an error response is sent to the originator and that error bounces back. It is unlike that the bounce message will have a proper security label. The S/MIME mail client would show the confidential content.

7.3 Mailing lists

Mailing lists that encrypt its content may be targets for DoS-Attacks if they do not prevent Mail-Loops. Using simple RFC822-Header spoofing, it is easy to subscribe one encrypted mailing list to another, thereby setting up an infinite loop.

If a mailing list agent receives a large number of undecryptable messages it shall notify an admin, because it may be a ciphertext attack.

Bibliography

[1] RFC 822 Standard for the Format of ARPA Internet Text Messages

[2] RFC 2311 S/MIME Version 2 Message Specification

[3] RFC 2315 PKCS #7: Cryptographic Message Syntax Version 1.5

[4] RFC 2630 Cryptographic Message Syntax

[5] RFC 2632 S/MIME Version 3 Certificate Handling

[6] RFC 2633 S/MIME Version 3 Message Specification

[7] RFC 2634 Enhanced Security Services for S/MIME