Message Takeover Attacks against S/MIME

Falko Strenzke

cryptosource GmbH,
Darmstadt, Germany
fstrenzke@cryptosource.de

1 Preamble

This document describes a new vulnerability of S/MIME which applies to other cryptographic standards as well. The document and its content must be treated as confidential and only be distributed on a need-to-know principle within your institution. Publication is not allowed. Redistribution to other parties is only allowed with written consent from the author.

2 Message Takeover Attacks

Message takeover attacks are a new class of attacks against naïve sign & encrypt as for instance implemented by S/MIME. As apparently generally unknown, naïve sign & encrypt which allows outer signatures (i.e. encrypt-then-sign) fails to provide integrity protection even for signed-then-encrypted emails, when we consider the integrity of the plaintext as composed by the original author. This flaw will soon be made public us.

A trivial form of message takeover attacks was previously known[1]. It is applicable when the sender applies encrypt-then-sign instead of sign-then-encrypt. The latter is the commonly accepted choice for email composition of clients today.

The message takeover attack works as follows: Eve, the attacker, strips off the original signature from a signed-then-encrypted message from Alice to Bob, the transfer of which he blocks, potentially alters the message, signs it himself and then forwards it to Bob who will receive it as a message in the encrypt-then-sign format from Eve. The problem is that S/MIME allows both sign-then-encrypt and encrypt-then-sign, thus it is possible for Eve to generate validly signed emails in this manner.

In this way, even in the setting where Bob only accepts signed emails, Eve is capable to inject messages to him as originating from her which were authored, signed, and encrypted by Alice.

This potentially leads to a number of practical attacks, however, here we only point out that if Bob replies to the message, i.e. to Eve's email account, with the message history, possibly in an encrypted email, Eve learns the contents of the message from Alice to Bob. Bob might also disclose other information based

[1] http://world.std.com/~dtd/sign_encrypt/sign_encrypt7.html

on the believed legitimation of Eve which he might conclude from her displayed knowledge in the forged email.

2.1 Technical Description of the Attack

In the following, we give a technical description of a successful message takeover attack against Thunderbird 38.4.0. The setting is that Alice sends a signed-then-encrypted message to Bob. In order to carry out the attack successfully, Eve first needs to learn the exact message format of Alice's signed emails. This preparatory step is described in the following paragraph.

Eve learns the Email Format used by Alice In order for Eve to learn the length of the MIME signature appended to the clear text she requests from Alice a signed email. The format of the plaintext message in our example attack is as in Figure 1. Eve determines the number of blocks to remove from end of the ciphertext in order to strip off the signature, where she keeps intact the last two blocks in order not to disturb the padding.

This results in the plaintext ending after the attachment, just before the signature part

```
Content-Type: application/pkcs7-signature; name="smime.p7s"
```

would follow. The appending of the original two final blocks causes a characteristic CBC-seam in the form of a corrupted eight-byte block:

```
Content-Type: image/jpeg;
 name="logo.jpg"
Content-Transfer-Encoding: base64
Content-ID: <part1.04000902.07080108@cryptosource.de>
Content-Disposition: inline;
 filename="logo.jpg"

/9j/4AAQSkZJRgABAQEB7gHuAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEB
... << attachment base64 encoded >>
4bHW6zWPxIAHjIJq5fPibhsHAjmwoY3cEibh2SILIsmiCar4g7dPnamOS7twsuoopk6dAH//
2Q==
--------------020007030300050007040_}#O|^K^H/0--
```

Eve Blocks, Modifies and then Forwards an Email from Alice to Bob In the second step, Eve intercepts a signed and encrypted email from Alice to Bob so that the message never reaches Bob. She performs the truncation of the ciphertext as she determined in the previous step. She might also introduce further modifications of the CBC ciphertext to change the plaintext (given that she has partial knowledge of the plaintext). Then Eve applies an outer S/MIME signature with her own private key to the enveloped data before she sends the

```
Content-Type: multipart/signed; protocol="application/pkcs7-
    signature"; micalg=sha-512; boundary="------------
    ms040709010602000909080300"

--------------ms040709010602000909080300
Content-Type: multipart/alternative;
 boundary="------------020409060108070406070101"

This is a multi-part message in MIME format.
--------------020409060108070406070101
Content-Type: text/plain; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<< plaintext email content >>

--------------020409060108070406070101
Content-Type: multipart/related;
 boundary="------------020007030300050007040305"


--------------020007030300050007040305
Content-Type: text/html; charset=utf-8
Content-Transfer-Encoding: quoted-printable

<html>
<< html email content >>
</html>

--------------020007030300050007040305
Content-Type: image/jpeg;
 name="logo.jpg"
Content-Transfer-Encoding: base64
Content-ID: <part1.04000902.07080108@cryptosource.de>
Content-Disposition: inline;
 filename="logo.jpg"

/9j/4AAQSkZJRgABAQEB7gHuAAD/2
    wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEB
... << attachment base64 encoded >>
--------------020007030300050007040305--

--------------020409060108070406070101--

--------------ms040709010602000909080300
Content-Type: application/pkcs7-signature; name="smime.p7s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7s"
Content-Description: S/MIME Cryptographic Signature

MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgMFADCABgkqhkiG9w0BBwEAAKCC

... << S/MIME signature base64 encoded >>
--------------ms040709010602000909080300--
```

Fig. 1. Example of the multipart/signed email format which is mostly used by email clients.

email to Bob. Bob receives an email which he understands to originate from Eve's email address and carries a valid signature as displayed by his email client. Apparently, the Thunderbird email client ignored the character sequence resulting from the CBC-seam and also that the plaintext mime-content was announced as "multipart/signed", as in the first line of Figure 1, and then no signature attachment was found.

3 Mitigation of Message Takeover Attacks

The fundamental flaw leading to the attacks is within the S/MIME standard. However, even without the repair of that standard, countermeasures are possible.

3.1 Restricting the S/MIME Format

One option is for the receiving client to consider only messages with inner signatures as validly signed. This a is valid countermeasure, since the attacker is incapable of producing signatures of unknown message content (if the content is known to him completely then he can simply forge the message from scratch). This solution cannot be backward compatible to the current S/MIME specification as currently mere outer signatures are also considered valid. However, since to our knowledge basically all clients already produce emails in the proposed format, this incompatibility would hardly be relevant.

Note that already the possibility for Eve to only strip off signatures without the possibility of generating new ones would be a drawback. With the countermeasure proposed here, this cannot be prevented. Accordingly, the countermeasure from the next section should be implemented in any case.

3.2 Mitigation the Application Layer: Stricter MIME-Parsing

In the previous section we discussed a measure necessary to achieve security on the cryptographic layer, which is the only sound solution. However, as that measure breaks compatibility with the current S/MIME specification, here we propose a mitigation on the application layer which retains compatibility to the S/MIME specification. Furthermore, these countermeasures also offer protection against attacks where Eve only strips off signatures and modifies the encrypted message without applying her own signature.

Before we come to the discussion of these measures, we wish to point out why mitigation on the application layer can never reliably compensate a broken cryptographic layer. This is due to the following considerations: first of all, application processing is generally far too complex to allow for systematic of formal security analysis, and thus the trust in their effect can never be as high as that in a sound cryptographic solution. Furthermore, application processing is subject to varying requirements, and thus the additional security mechanisms must be feared to be rather unstable from version to version. Another problem is that such mitigating measures will hardly find a suitable place in any specification. Since they are out of the scope of S/MIME, they cannot be included there, and since it is not the task of the MIME processing to enforce security goals of S/MIME, this is also not the right place.

Now we turn to the measures that should be implemented by clients functioning on the basis of the broken S/MIME specification when parsing the MIME content of an encrypted email:

1. Parsing should be as strict as possible. Any inconsistencies, like the lack of an inner signature in a mail that is headed by "Content-Type: multipart/signed", irregular boundary tags etc., should lead to the display of an invalid signature. The presence of a signature part ("Content-Type: application/pkcs7-signature") in the message should always lead to the interpretation of this as the message carrying an inner signature, even when it is not expected from the previous context, and also here any inconsistencies in the MIME message should lead to the signature shown as invalid.
2. The presence of invalid character codes should also lead to invalid signatures.
3. Furthermore, when the above measures detect an irregular MIME message, also the encryption result should be indicated as potentially manipulated.

It should be unnecessary to say that it is not possible to have any guarantee that implementation of these measures achieves reliable defence against attacks. However, the straightforward practical attacks as devised in this work will not function in that simple form any more.