An Exploration of S/MIME Error Conditions in Popular Mail User Agent Interfaces

Bachelor Arbeit

Maria Sigal

Arbeitsgruppe Sichere Identität
Fachbereich Mathematik und Informatik
Freie Universität Berlin

5. April 2018
A set of standards for secure email messaging defined by IETF.
Uses:

1. digital signatures
2. encryption
S/MIME provides:

1. authentication
2. message integrity
3. non-repudiation of origin
4. data confidentiality
S/MIME Certificates

Bindings between public key and identities
S/MIME in Mail on MacOS

for example:

Maria Sigal
Test Case #14
To: Kim Jong Un
Security: Signed (Maria Sigal), Encrypted

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
Errors in Implementation

1. the programm has to be vulnerability-free
2. developer do implementation mistakes
find exploit to:

1. forge a signature on an arbitrary email message
2. obtain decrypted content.
3. disrupt email client services.
Methods:

create complex email message structures:

1. manipulating S/MIME objects.

2. manipulating other objects related to S/MIME such as MIME, CMS.

3. applying a different combination of security operations.
MIME

The Multipurpose Internet Mail Extensions (MIME) is RFC Internet standard RFC which defines the data format of emails.

1. textual message bodies in character set other than US-ASCII.

2. an extensible set of different formats for non-textual message bodies like images, application programs, videos.

3. messages bodies with multiple parts.

4. textual header information in character sets other than US-ASCII
Multipart alternative

From: Maria Sigal <sigalmaria@gmail.com>
Content-Type: multipart/alternative;
   boundary="Apple-Mail=_0986B0D4-DE68-4489-80E6-DC7266BE5142"
Mime-Version: 1.0 (Mac OS X Mail 10.3 (3273))
Subject: Html
X-Universally-Unique-Identifier: B71EE87B-E4F0-4A2F-8EFF-ED6050BB6439
Message-Id: <8DBC8FC9-829D-4690-AB17-ECAE09280F24@gmail.com>
Date: Wed, 4 Apr 2018 14:48:36 +0200
To: Maria Sigal <sigalmaria@gmail.com>

--Apple-Mail=_0986B0D4-DE68-4489-80E6-DC7266BE5142
Content-Transfer-Encoding: 7bit
Content-Type: text/plain;
   charset=uas-ascii

Kishuf
Ze kishuf

--Apple-Mail=_0986B0D4-DE68-4489-80E6-DC7266BE5142
Content-Transfer-Encoding: 7bit
Content-Type: text/html;
   charset=us-ascii

<html><body style="word-wrap: break-word; -webkit-ntsp-mode: space; -webkit-line-break: after-white-space;"
   class="">
   Kishuf&nbsp;&lt;font color="#9929bd" class="">
   kishuf</font>&nbsp;
   </body></html>

--Apple-Mail=_0986B0D4-DE68-4489-80E6-DC7266BE5142--
Hi,

See the attached image.

--Apple-Mail_A-1429141-F66C-4576-B5CD-0264A782656
Content-Transfer-Encoding: 7bit
Content-Type: text/plain;
charset=us-ascii

Hi

See the attached image.

--Apple-Mail_A-1429141-F66C-4576-B5CD-0264A782656
Content-Transfer-Encoding: 7bit
Content-Type: text/html;
charset=us-ascii

<html>
<head>
<meta http-equiv="Content-Type" content="text/html charset=us-ascii"/>
</head>
<body>

</body>
</html>

--Apple-Mail_A-1429141-F66C-4576-B5CD-0264A782656
Content-Transfer-Encoding: 7bit
Content-Type: text/html;
charset=us-ascii

<html>
<head>
<meta http-equiv="Content-Type" content="text/html charset=us-ascii"/>
</head>
<body>

</body>
</html>

--Apple-Mail_A-1429141-F66C-4576-B5CD-0264A782656
Content-Transfer-Encoding: base64
Content-Disposition: inline;

filename=nickinajaj.png
Content-Type: image/png;
name="nickinajaj.png"
Content-ID: <33268039-3000-4F82-98A7-A018E7587E74>

--Apple-Mail_A-1429141-F66C-4576-B5CD-0264A782656
Content-Transfer-Encoding: base64
Content-Disposition: inline;

filename=nickinajaj.png
Content-Type: image/png;
name="nickinajaj.png"
Content-ID: <33268039-3000-4F82-98A7-A018E7587E74>
cryptographic message syntax (CMS)

1. IETF standard for cryptographically protected messages.

2. can sign and encrypt arbitrary message content.

3. allows multiple encapsulations.
CMS EnvelopedData
CMS SignedData

Diagram showing the structure of CMS SignedData, including:
- Version
- (Set of) Digest Algorithms
- Content Info
- Set of certificates
- Set of CRLs
- Signer Info
- Content
- Content type
- Version
- Signer ID (issuer and ser. no.)
- Digest Algorithm
- Authenticated Attributes
- Digest Encryption Alg.
- Encrypted digest (signature)
nested CMS

1. any combination of cryptographic operations could be applied to the message.
2. signBy(Maria) → signBy(Donald) → encrypt
3. encrypt → encrypt → sign
4. etc.
encrypted message with S/MIME

Content-Type: application/pkcs7-mime; name="smime.p7m"; smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="smime.p7m"
Content-Description: S/MIME Encrypted Message

MIAGCSqGSIb3DQEHA6CAMIACAQAwgFvMIIBawIBADBTMEsxFzAVBgNVBAMMDk1hcmllhIFNpZ2Fs
IENBMQswCQYDVQQGEwJERTEjMCEGCSqGSIb3DQEJARYuc2lnYWxtyX3jPyUBnbWFpbC5jb20CBDRn
XYEwDQYJKoZIhvCAQEBBQAEEgADekYz5GkttFzLJzbJRh28x2gQKEWwc/fo8ljUNHw6jejch3xv5
bcOGCmqNwHRaYAovu0ye5gviAN5TtAkSAMmBK1bZbsY4QmX3qnvjHki/ZkCafRvKfZ9W9My/BRH/
/gSurMvX+60nF0Ay00w03EFyVD8tYJI4uz2GCj901b7CLEJgcLr20NZwNDLG01WsCy5yqفزvY8
BIF/E3o4edMHeaV1D031GnVKrdkUDSjzAwh3tThkfRni0KPiCF4x5i6lkVouTgBN30dclo+SqeKK
D+7qFDB+50eLrFC+mUE7XQs/88BYD16sYh9SLWTUEDyTxzVCo0BExcUpfm3dNZCzCABkgqkhiG
9w0BBwEwH0YJYIZIAWUDBAEgBBaIHgD7q8xQA50GBUTR96poIAEggP0NItmt9M88YNJwXu5Kmdny
4rz8wylQBPfGg+KZI5EtBwAqqVhYcB8SnK+I1ixVDDntFLDClC0HDK9ZtB1lha22DupeE2KSBQDCCX
fTLP2JtNiNksWlobo887RWhNqHyj1tpFV029g3Ms+t0ak8nzFBy+JORSUxvC0ombbP0mPA92ZaAX
6ahEuAdbT8d+4sZ/F15+beBSthyuG9T40zaHl0wYQUPe0KuPfR0zaR3J7bCrTq

signed message with S/MIME
S/MIME could be also nested

According to the standard, each individual layer of CMS is wrapped with MIME. For example, for sign → encrypt

1. a MIME entity of type text/plain is created.
2. the MIME entity is processed to the CMS SignedData object.
3. the CMS SignedData is encoded with base64 and wrapped with S/MIME.
4. S/MIME message is processed to the CMS EnvelopedData object.
5. the CMS EnvelopedData is encoded with base64 and wrapped with S/MIME.
Test cases

<table>
<thead>
<tr>
<th>Test Suite Name</th>
<th>Test Suite Id</th>
<th>Test Case Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>S/MIME basic functionality</td>
<td>101</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>S/MIME sign &amp; encrypt flaw</td>
<td>102</td>
<td>33, 34, 36, 38</td>
</tr>
<tr>
<td>S/MIME standards violation</td>
<td>103</td>
<td>25, 26</td>
</tr>
<tr>
<td>S/MIME encapsulation</td>
<td>104</td>
<td>11, 12, 13, 14, 27, 33, 34, 36, 37</td>
</tr>
<tr>
<td>CMS encapsulation</td>
<td>105</td>
<td>18, 19, 20, 21, 30, 31, 32</td>
</tr>
<tr>
<td>S/MIME as an email attachment</td>
<td>106</td>
<td>16, 17, 22</td>
</tr>
<tr>
<td>S/MIME with &quot;multipart/alternative&quot; messages</td>
<td>107</td>
<td>4, 5, 6, 10, 15</td>
</tr>
<tr>
<td>S/MIME with &quot;multipart/mixed&quot;</td>
<td>108</td>
<td>7, 8, 9, 23, 24, 28</td>
</tr>
<tr>
<td>S/MIME with multiple certificates</td>
<td>109</td>
<td>11, 12, 13, 14</td>
</tr>
</tbody>
</table>
S/MIME test tool development

1. importing of sender and recipient certificates.
2. generation of email messages according to properties defined.
3. sending generated email messages to the recipient.
4. saving the sent messages as a text file.
S/MIME test tool development

1. jdk 1.8
2. bcmail-jdk15on library of BouncyCastle
3. JavaMail API version 1.6.0.
4. java.security
Tested Email Clients

1. Mail on IOS 11.2.5
2. Thunderbird 52.6.0 on macOS Sierra 10.12.16
3. Mail 10.03 on macOS Sierra 10.12.16
running the tests

gatter:smiitest testing berutie1S java -jar target/SmiiteTest.jar -prop "config/config.properties" runner 14
Connecting to smtp server
mail.smtp.host: smtp.gmail.com
mail.smtp.port: 587
Executing Test Case 14

The following message sent:

[

Date: Fri, 23 Feb 2018 14:02:42 +0100 (CET)
From: trumpcondaljunco@gmail.com
To: kimjungun84kk@gmail.com
Message-ID: <155840798885.5.1519309952727@gmail.com>
Subject: Test Case #14
MIME-Version: 1.0
Content-Type: application/pkcs7-mime; name=smite-type.properties; smite-type=encoded; data
Content-Transfer-Encoding: base64

MIAC0Ccs6Ib3DQEHAQgDAMICAOAQExZbANB3qg9hkgZBQMqEgEFACADgkchki09w8bBwQccCSAB1D
E6NvbH1bnQTVHlvZToSYXv8n6GjYXRbQ24vcGtjczbTbwlT7sbmFzT76ic21pvU0cUd17js
C2l0bWU2UHJzWlTlbnZIB69wZ2QFz62odPQ29u60yvdC1u0mFzZGluZzgqYFz
ZT5YQGpQDz5Q5W95L0mcC3B20b9w9u19Hd32Yw7z050y3m0Wx1HbFMTZ1c3pWhuucbd
Igk6QZ9u6y9vucdC3znKw56GjYjg0mg5nSyL1F6vYJ35e4hL1qfxN3w3ki23luJc2Q0idn
U36JlW183qRZ0918nxJ1UlBQ0FQr0zqnn232uTUL0jFmFSu1BR3EUUZe2E0QVZC050kieKNTUr
WMyjuvocSSu2C0FoyRmNMcckFTk3SUJX3Q1fZHRFZU8uddFdpF1U1RsK1DruDU3Fuj011M9BRRUpB
U11vYzJsb11xHeRZEpwVcbm3XnrB1Lqcomplexyc030SU2nRnS1hZRXEvUKs29sSWy2B6BUVc
Q10DF3uonNv5yE8L0utzvexcvnh1mfnIT79xZSnQV3e2SW1bVS81d6c118T7v311n4bDMN
Ck1EL3jKcEVNFs1ygrzrzx5v9V3Ql0wE0zQFzVw13Zkx0vNhD6YNTNT52hZzVd0USJ1i2y
YnZ1ITX18y1pYQXfS1ySy6v8C0nLZY39byV70Y1sH05sC1d1jHeWeg3TXVwK13ZlyP0Q1Z2
OnJU2Z1v13sH63F1k5skzVv1xdS2xJ68G3RFLxJ549ZXmWcN6Rt1TVnEMv2zy1M2F2
U1qJnqFMJUG3v21i2rKzmdz3Nj1k3jbdzaSJXNF3sAHnPvMr6B2rvX21a1d94SA3Qrendem38m9K
Q3V2e3jINrQNE69e3a1v3lHVLv2iz157Bvya3eITvWyv3F0vecm13sJvylurs3RnXBee4cmF0q1nwa6E5
PnTHeCekuiuhG1V1FQ2ndvBNEbdrchraJsNC0j11NE5EB03V35AFZ311k5w1v1REQkFfcU3CRFBN
TVhxbuK1vVpY73pZn1Q1cyhO9QJUnV31ewZU2w9b837s3rj0b19x2k6ShUJUVQNnC1b1h1u7Ta
Yz3v622Qg3mdazXN1XlR1ts6i7L1xKOFs-wjdJyoeK0hGnB2KX5vB3DTvNdcS3hfVdW8sz915f5p113LUDBx
YnVn8eEINW3dRJUTNcnd6ZBS00odz81vTjBqRERT0fjckdON91y13zUxycrbg9y4VUVZQY2hW
S9yYb1R5roQ5dVM3mszycdkDmzv5sEokv0nNFMFZe86EgpoDqozczZhBNjcwawPw6ewNxEEZU1h1y
blb1uvmhFv2bF3RuU0TFV7RtheU0FYfU3uv2Bz1IPm4n39c6tIcmo8TH4ac6oNeH5UEUyWdM8xnc3b
Dp32Yj3C13JU27z30XKwc2Zv66tL1u1qg3v01Nh1Ry580PxmJmmwV4E3BvY8Q0Zw2dBA11N1QYZ3
Dn11VUsbN0nR1I8xmJy09whRvBt9Y8OQ02w6ErdVv1pymr5tmUJH36yV58BS8n3d3dcvkhubbhHh
Y1V0Q6nAYVYKvWkL84vA0x3V84F10t8dbKE02NOy5q0a6NCw1e0d9kK04az1212b6s0Fr
Z49b1hA6jgyc3KJVEUZnJMBx1r50N3XNn0xQCF3V23nUXJ0X0i3D3X1fL6uV0r2sa0vOBnZS4a6Up
UDcEBoM8dGooR1RuWyXyM0RZBZLUwa0v87ZQezFpmA4y1111w66emTef1Lzntwq1h9rJrRINQ1Q4
T2Z3xu1j1rxc1kz8ZcV0J0M1iRuVmFtEt9iK1NeDqoeR0yeTR27C7n7uyVX1Q0Bwa8yZfVQG0wZv
36 out of 111 tests failed.
We sent the following message:

1. a MIME entity of type text/plain with some text is created and signed with S/MIME multipart/signed format.

2. another MIME entity of type text/plain with some text is created.

3. an email message of type multipart/mixed is created.

4. a MIME entity from step 2 is appended to the message from step 3.

5. a MIME entity from step 1 is appended as message/rfc822 (email attachment) to message from step 3.
Test Case #16
To: Kim Jong Un

Update on location:
Franz Merhing Platz 3, Berlin

Nada
From: trumpdonaldjunior@gmail.com
Subject: signedMessageForAttachment
Date: 3. April 2018 at 22:20:13 GMT+2
To: kimjongun84k@gmail.com

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
Donald Trump
signed S/MIME as an email attachment 2

Now we did aditional manipulations:

1. the external body part of type text/html ending with the tag <div style='visibility:hidden'>

2. the external body part was signed by Maria Sigal.

3. email message is spoofed to make it look coming from Donal Trump
Hi Kim, The party is canceled. Donald
Hi Kim, The party is canceled. Donald

---

Betreff: sendSignedEmailAsAttachment2Certificates
Von: trumpdonaldjunior@gmail.com
Datum: 03.04.18, 22:23
An: kimjongun84k@gmail.com

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
Location:
Rothschild Boulevard 32, Tel Aviv

Donald Trump
encrypted S/MIME as an email attachment

1. the external message has some text

2. the email message has an encrypted email message as attachment
Mail MacOs

no indication for encryption performed.

Donald Trump

Test Case #22
To: Kim Jong Un

Hi,
Have you seen my last message? waiting for reply.

Donald Trump
From: trumpdonaldjunior@gmail.com
Subject: Test Case 22 attached email
Date: 3. April 2018 at 22:39:02 GMT+2
To: kimjongun84k@gmail.com

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
Here is the location:
Rothschild Boulevard 32, Tel Aviv
Donald Trump
The email attachment is included in the reply message.
Thunderbird MacOS

The email attachment is not included in the reply message.

Am 03.04.18 um 22:39 schrieb uplietruump100@gmail.com:

| Hi,  
| Have you seen my last message? waiting for reply.  
| Donald Trump  

Von: Kim Jong Un <kimjongun84k@gmail.com>  
An: uplietruump100@gmail.com  

Betreff: Re: Test Case #22
Sign $\rightarrow$ Encrypt $\rightarrow$ Sign

1. the MIME entity of type text/plain is signed by Donald Trump
2. The result of 1 is encrypted for Kim
3. the result of 2 is signed by Maria Sigal
Sign → Encrypt → Sign in Mail MacOS

From: Maria Sigal

Test Case #14

To: Kim Jong Un

Security: Signed (Maria Sigal), Encrypted

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
On 3. Apr 2018, at 23:28, marie anglais@fu-berlin.de wrote:

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
Hi,
sure!
kisses :* <3 <3

Am 03.04.18 um 23:50 schrieb maria.sigal@fu-berlin.de:

Hi Kim,
I am throwing a big party with fireworks tonight, please come.
The location:
Rothschild Boulevard 32, Tel Aviv
CMS encapsulation

1. the MIME entity is packed in the CMS EnvelopedData.

2. the CMS EnvelopedData is signed with CMS SingedData by Donald Trump.

3. The result is wrapped with S/MIME.
CMS encapsulation in Thunderbird MacOs

Messages are not downloaded from the server.

Thunderbird
The current operation on 'Inbox' did not succeed. The mail server for account kimjongun84k@gmail.com responded: System Error (Failure).
CMS encapsulation in Mail IOS

Denial-of-service - any messages later sent to the recipient also not downloaded.
Surreptitious Forwarding Attack

\[ A \rightarrow B \{ \{ \text{"sales strategy"} \}^a \}^B \] 
\[ B \rightarrow C \{ \{ \text{"sales strategy"} \}^a \}^C \]
Encrypt-and-Sign Attack

\[ A \rightarrow B\{\{"new invention \}\}^B \]^a \]
\[ B \rightarrow C\{\{"new invention \}\}^B \]^c \]
1. Sign the recipient’s name into the plaintext
2. Encrypt the sender’s name into the plaintext
3. Incorporate both names
4. Sign the signed and encrypted message again
5. Encrypt the signed ciphertext again
Sign → Encrypt → Sign

Not supported. The same output as with the sign-then-encrypt message.

☆ Donald Trump
Test Case #33
To: Kim Jong Un
Security: Signed (Donald Trump), Encrypted

Hi Kim,
i am throwing a big party with fireworks tonight, please come.
Location, Top Secret:
Franz Mehring Platz 3, Berlin
Encrypt → Sign → Encrypt

Not supported. The same output as with the sign-then-encrypt message.
Danke