Internet Engineering Task Force (IETF)
Request for Comments: 6216
Category: Informational
ISSN: 2070-1721

C. Jennings
Cisco Systems
K. Ono
Columbia University
R. Sparks
B. Hibbard, Ed.
Tekelec

April 2011

Example Call Flows Using Session Initiation Protocol (SIP) Security Mechanisms

Abstract

This document shows example call flows demonstrating the use of Transport Layer Security (TLS), and Secure/Multipurpose Internet Mail Extensions (S/MIME) in Session Initiation Protocol (SIP). It also provides information that helps implementers build interoperable SIP software. To help facilitate interoperability testing, it includes certificates used in the example call flows and processes to create certificates for testing.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Not all documents approved by the IESG are a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6216.

Table of Contents

1. Introduction

This document is informational and is not normative on any aspect of SIP.

SIP with TLS ([RFC5246]) implementations are becoming very common. Several implementations of the S/MIME ([RFC5751]) portion of SIP ([RFC3261]) are also becoming available. After several interoperability events, it is clear that it is difficult to write these systems without any test vectors or examples of "known good" messages to test against. Furthermore, testing at the events is often hindered due to the lack of a commonly trusted certification authority to sign the certificates used in the events. This document addresses both of these issues by providing messages that give detailed examples that implementers can use for comparison and that can also be used for testing. In addition, this document provides a common certificate and private key that can be used to set up a mock Certification Authority (CA) that can be used during the SIP interoperability events. Certificate requests from the users will be signed by the private key of the mock CA. The document also provides some hints and clarifications for implementers.

A simple SIP call flow using SIPS URIs and TLS is shown in Section 3. The certificates for the hosts used are shown in Section 2.2, and the CA certificates used to sign these are shown in Section 2.1.

The text from Section 4.1 through Section 4.3 shows some simple SIP call flows using S/MIME to sign and encrypt the body of the message. The user certificates used in these examples are shown in Section 2.3. These host certificates are signed with the same mock CA private key.

Section 5 presents a partial list of items that implementers should consider in order to implement systems that will interoperate.

Scripts and instructions to make certificates that can be used for interoperability testing are presented in Appendix A, along with methods for converting these to various formats. The certificates used while creating the examples and test messages in this document are made available in Appendix B.

Binary copies of various messages in this document that can be used for testing appear in Appendix C.

2. Certificates

2.1. CA Certificates

The certificate used by the CA to sign the other certificates is shown below. This is an X.509v3 ([X.509]) certificate. Note that the X.509v3 Basic Constraints in the certificate allows it to be used as a CA, certification authority. This certificate is not used directly in the TLS call flow; it is used only to verify user and host certificates.

```
Version: 3 (0x2)
Serial Number:
    96:a3:84:17:4e:ef:8a:4c
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, ST=California, L=San Jose, O=sipit,
        OU=Sipit Test Certificate Authority
Validity
    Not Before: Jan 27 18:36:05 2011 GMT
    Not After : Jan  3 18:36:05 2111 GMT
Subject: C=US, ST=California, L=San Jose, O=sipit,
        OU=Sipit Test Certificate Authority
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
        Modulus (2048 bit):
            00:ab:1f:91:61:f1:1c:c5:cd:a6:7b:16:9b:b7:14:
            79:e4:30:9e:98:d0:ec:07:b7:bd:77:d7:d1:f5:5b:
            2c:e2:ee:e6:b1:b0:f0:85:fa:a5:bc:cb:cc:cf:69:
            2c:4f:fc:50:ef:9d:31:2b:c0:59:ea:fb:64:6f:1f:
            55:a7:3d:fd:70:d2:56:db:14:99:17:92:70:ac:26:
            f8:34:41:70:d9:c0:03:91:6a:ba:d1:11:8f:ac:12:
            31:de:b9:19:70:8d:5d:a7:7d:8b:19:cc:40:3f:ae:
            ff:de:1f:db:94:b3:46:77:6c:ae:ae:ff:3e:d6:84:
            5b:c2:de:0b:26:65:d0:91:c7:70:4b:c7:0a:4a:bf:
            c7:97:04:dd:ba:58:47:cb:e0:2b:23:76:87:65:c5:
            55:34:10:ab:27:1f:1c:f8:30:3d:b0:9b:ca:a2:81:
            72:4c:bd:60:fe:f7:21:fe:0b:db:0b:db:e9:5b:01:
            36:d4:28:15:6b:79:eb:d0:91:1b:21:59:b8:0e:aa:
            bf:d5:b1:6c:70:37:a3:3f:a5:7d:0e:95:46:f6:f6:
            58:67:83:75:42:37:18:0b:a4:41:39:b2:2f:6c:80:
            2c:78:ec:a5:0f:be:9c:10:f8:c0:0b:0d:73:99:9e:
            0d:d7:97:50:cb:cc:45:34:23:49:41:85:22:24:ad:
            29:c3
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Subject Key Identifier:
        95:45:7E:5F:2B:EA:65:98:12:91:04:F3:63:C7:68:9A:58:16:77:27
```

```
        X509v3 Authority Key Identifier:
            95:45:7E:5F:2B:EA:65:98:12:91:04:F3:63:C7:68:9A:58:16:77:27

        X509v3 Basic Constraints:
            CA:TRUE
        Signature Algorithm: sha1WithRSAEncryption
    06:5f:9e:ae:a0:9a:bc:b5:b9:5b:7e:97:33:cc:df:63:98:98:
    94:cb:0d:66:a9:83:e8:aa:58:2a:59:a1:9e:47:31:a6:af:5c:
    3f:a2:25:86:f8:df:05:92:b7:db:69:a1:69:72:87:66:c5:ab:
    35:89:01:37:19:c9:74:eb:09:d1:3f:88:7b:24:13:42:ca:2d:
    fb:45:e6:cc:4b:f8:21:78:f3:f5:97:ec:09:92:24:a2:f0:e6:
    94:8d:97:4a:00:94:00:bd:25:b8:17:2c:52:53:5d:cc:5c:48:
    a4:a1:1d:2d:f6:50:55:13:a4:d3:b2:a2:f4:f1:b9:6d:48:5e:
    5c:f3:de:e0:fc:59:09:a1:d9:14:61:65:bf:d8:3f:b9:ba:2e:
    7c:ed:5c:24:9b:6b:ca:aa:5f:f1:c1:1e:b0:a8:da:82:0f:fb:
    4c:71:3b:4d:7b:38:c8:e3:8a:2a:19:34:44:26:0b:ea:f0:47:
    38:46:28:65:04:e2:01:52:dd:ec:3d:e5:f5:53:74:77:74:75:
    6d:c6:d9:c2:0a:ac:3b:b8:98:5c:55:53:34:74:52:a8:26:b1:
    2f:30:22:d0:8b:b7:f3:a0:dd:68:07:33:d5:ae:b7:81:b2:94:
    58:72:4e:7c:c6:72:2f:bd:6c:69:fb:b5:17:a8:2a:8d:d7:2c:
    91:06:c8:0c
```

The certificate content shown above and throughout this document was rendered by the OpenSSL "x509" tool. These dumps are included only as informative examples. Output may vary among future revisions of the tool. At the time of this document's publication, there were some irregularities in the presentation of Distinguished Names (DNs). In particular, note that in the "Issuer" and "Subject" fields, it appears the intent is to present DNs in Lightweight Directory Access Protocol (LDAP) format. If this was intended, the spaces should have been omitted after the delimiting commas, and the elements should have been presented in order of most-specific to least-specific. Please refer to Appendix A of [RFC4514]. Using the "Issuer" DN from above as an example and following guidelines in [RFC4514], it should have instead appeared as:

```
Issuer: OU=Sipit Test Certificate Authority,O=sipit,L=San Jose,
        ST=California,C=US
```

The ASN.1 ([X.683]) parse of the CA certificate is shown below.

```
  0:l= 949 cons: SEQUENCE
  4:l= 669 cons:  SEQUENCE
  8:l=   3 cons:   cont [ 0 ]
 10:l=   1 prim:    INTEGER           :02
 13:l=   9 prim:   INTEGER           :96A384174EEF8A4C
 24:l=  13 cons:   SEQUENCE
```

```
 26:l=   9 prim:    OBJECT            :sha1WithRSAEncryption
 37:l=   0 prim:    NULL
 39:l= 112 cons:   SEQUENCE
 41:l=  11 cons:    SET
 43:l=   9 cons:     SEQUENCE
 45:l=   3 prim:      OBJECT            :countryName
 50:l=   2 prim:      PRINTABLESTRING   :US
 54:l=  19 cons:    SET
 56:l=  17 cons:     SEQUENCE
 58:l=   3 prim:      OBJECT            :stateOrProvinceName
 63:l=  10 prim:      UTF8STRING
  43 61 6c 69 66 6f 72 6e-69 61                      California
 75:l=  17 cons:    SET
 77:l=  15 cons:     SEQUENCE
 79:l=   3 prim:      OBJECT            :localityName
 84:l=   8 prim:      UTF8STRING
  53 61 6e 20 4a 6f 73 65-                           San Jose
 94:l=  14 cons:    SET
 96:l=  12 cons:     SEQUENCE
 98:l=   3 prim:      OBJECT            :organizationName
103:l=   5 prim:      UTF8STRING
  73 69 70 69 74                                     sipit
110:l=  41 cons:    SET
112:l=  39 cons:     SEQUENCE
114:l=   3 prim:      OBJECT            :organizationalUnitName
119:l=  32 prim:      UTF8STRING
  53 69 70 69 74 20 54 65-73 74 20 43 65 72 74 69   Sipit Test Certi
  66 69 63 61 74 65 20 41-75 74 68 6f 72 69 74 79   ficate Authority
153:l=  32 cons:   SEQUENCE
155:l=  13 prim:    UTCTIME           :110127183605Z
170:l=  15 prim:    GENERALIZEDTIME   :21110103183605Z
187:l= 112 cons:   SEQUENCE
189:l=  11 cons:    SET
191:l=   9 cons:     SEQUENCE
193:l=   3 prim:      OBJECT            :countryName
198:l=   2 prim:      PRINTABLESTRING   :US
202:l=  19 cons:    SET
204:l=  17 cons:     SEQUENCE
206:l=   3 prim:      OBJECT            :stateOrProvinceName
211:l=  10 prim:      UTF8STRING
  43 61 6c 69 66 6f 72 6e-69 61                      California
223:l=  17 cons:    SET
225:l=  15 cons:     SEQUENCE
227:l=   3 prim:      OBJECT            :localityName
232:l=   8 prim:      UTF8STRING
  53 61 6e 20 4a 6f 73 65-                           San Jose
242:l=  14 cons:    SET
244:l=  12 cons:     SEQUENCE
```

```
246:l=   3 prim:      OBJECT            :organizationName
251:l=   5 prim:      UTF8STRING
  73 69 70 69 74                                  sipit
258:l=  41 cons:    SET
260:l=  39 cons:     SEQUENCE
262:l=   3 prim:      OBJECT            :organizationalUnitName
267:l=  32 prim:      UTF8STRING
  53 69 70 69 74 20 54 65-73 74 20 43 65 72 74 69   Sipit Test Certi
  66 69 63 61 74 65 20 41-75 74 68 6f 72 69 74 79   ficate Authority
301:l= 290 cons:   SEQUENCE
305:l=  13 cons:    SEQUENCE
307:l=   9 prim:     OBJECT            :rsaEncryption
318:l=   0 prim:     NULL
320:l= 271 prim:    BIT STRING
  00 30 82 01 0a 02 82 01-01 00 ab 1f 91 61 f1 1c   .0...........a..
  c5 cd a6 7b 16 9b b7 14-79 e4 30 9e 98 d0 ec 07   ...{....y.0.....
  b7 bd 77 d7 d1 f5 5b 2c-e2 ee e6 b1 b0 f0 85 fa   ..w...[,........
  a5 bc cb cc cf 69 2c 4f-fc 50 ef 9d 31 2b c0 59   .....i,O.P..1+.Y
  ea fb 64 6f 1f 55 a7 3d-fd 70 d2 56 db 14 99 17   ..do.U.=.p.V....
  92 70 ac 26 f8 34 41 70-d9 c0 03 91 6a ba d1 11   .p.&.4Ap....j...
  8f ac 12 31 de b9 19 70-8d 5d a7 7d 8b 19 cc 40   ...1...p.].}...@
  3f ae ff de 1f db 94 b3-46 77 6c ae ae ff 3e d6   ?.......Fwl...>.
  84 5b c2 de 0b 26 65 d0-91 c7 70 4b c7 0a 4a bf   .[...&e...pK..J.
  c7 97 04 dd ba 58 47 cb-e0 2b 23 76 87 65 c5 55   .....XG..+#v.e.U
  34 10 ab 27 1f 1c f8 30-3d b0 9b ca a2 81 72 4c   4..'...0=.....rL
  bd 60 fe f7 21 fe 0b db-0b db e9 5b 01 36 d4 28   .`..!.......[.6.(
  15 6b 79 eb d0 91 1b 21-59 b8 0e aa bf d5 b1 6c   .ky....!Y......l
  70 37 a3 3f a5 7d 0e 95-46 f6 f6 58 67 83 75 42   p7.?.}..F..Xg.uB
  37 18 0b a4 41 39 b2 2f-6c 80 2c 78 ec a5 0f be   7...A9./l.,x....
  9c 10 f8 c0 0b 0d 73 99-9e 0d d7 97 50 cb cc 45   ......s.....P..E
  34 23 49 41 85 22 24 ad-29 c3 02 03 01 00 01      4#IA."$.)......
595:l=  80 cons:   cont [ 3 ]
597:l=  78 cons:    SEQUENCE
599:l=  29 cons:     SEQUENCE
601:l=   3 prim:      OBJECT            :X509v3 Subject Key Identifier
606:l=  22 prim:      OCTET STRING
  04 14 95 45 7e 5f 2b ea-65 98 12 91 04 f3 63 c7   ...E~_+.e.....c.
  68 9a 58 16 77 27                                 h.X.w'
630:l=  31 cons:     SEQUENCE
632:l=   3 prim:      OBJECT            :X509v3 Authority Key Identifier
637:l=  24 prim:      OCTET STRING
  30 16 80 14 95 45 7e 5f-2b ea 65 98 12 91 04 f3   0....E~_+.e.....
  63 c7 68 9a 58 16 77 27-                          c.h.X.w'
663:l=  12 cons:     SEQUENCE
665:l=   3 prim:      OBJECT            :X509v3 Basic Constraints
670:l=   5 prim:      OCTET STRING
  30 03 01 01 ff                                    0....
677:l=  13 cons:  SEQUENCE
```

```
679:l=   9 prim:   OBJECT            :sha1WithRSAEncryption
690:l=   0 prim:   NULL
692:l= 257 prim:  BIT STRING
  00 06 5f 9e ae a0 9a bc-b5 b9 5b 7e 97 33 cc df   ..._.......[~.3..
  63 98 98 94 cb 0d 66 a9-83 e8 aa 58 2a 59 a1 9e   c.....f....X*Y..
  47 31 a6 af 5c 3f a2 25-86 f8 df 05 92 b7 db 69   G1..\?.%.......i
  a1 69 72 87 66 c5 ab 35-89 01 37 19 c9 74 eb 09   .ir.f..5..7..t..
  d1 3f 88 7b 24 13 42 ca-2d fb 45 e6 cc 4b f8 21   .?.{$.B.-.E..K.!
  78 f3 f5 97 ec 09 92 24-a2 f0 e6 94 8d 97 4a 00   x......$......J.
  94 00 bd 25 b8 17 2c 52-53 5d cc 5c 48 a4 a1 1d   ...%..,RS].\H...
  2d f6 50 55 13 a4 d3 b2-a2 f4 f1 b9 6d 48 5e 5c   -.PU........mH^\
  f3 de e0 fc 59 09 a1 d9-14 61 65 bf d8 3f b9 ba   ....Y....ae..?..
  2e 7c ed 5c 24 9b 6b ca-aa 5f f1 c1 1e b0 a8 da   .|.\$.k.._......
  82 0f fb 4c 71 3b 4d 7b-38 c8 e3 8a 2a 19 34 44   ...Lq;M{8...*.4D
  26 0b ea f0 47 38 46 28-65 04 e2 01 52 dd ec 3d   &...G8F(e...R..=
  e5 f5 53 74 77 74 75 6d-c6 d9 c2 0a ac 3b b8 98   ..Stwtum.....;..
  5c 55 53 34 74 52 a8 26-b1 2f 30 22 d0 8b b7 f3   \US4tR.&./0"....
  a0 dd 68 07 33 d5 ae b7-81 b2 94 58 72 4e 7c c6   ..h.3......XrN|.
  72 2f bd 6c 69 fb b5 17-a8 2a 8d d7 2c 91 06 c8   r/.li....*..,...
  0c                                                .
```

2.2. Host Certificates

The certificate for the host example.com is shown below. Note that the Subject Alternative Name is set to example.com and is a DNS type. The certificates for the other hosts are shown in Appendix B.

```
Version: 3 (0x2)
Serial Number:
    96:a3:84:17:4e:ef:8a:4f
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, ST=California, L=San Jose, O=sipit,
        OU=Sipit Test Certificate Authority
Validity
    Not Before: Feb  7 19:32:17 2011 GMT
    Not After : Jan 14 19:32:17 2111 GMT
Subject: C=US, ST=California, L=San Jose, O=sipit, CN=example.com
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
        Modulus (2048 bit):
            00:dd:74:06:02:10:c2:e7:04:1f:bc:8c:b6:24:e7:
            9b:94:a3:48:37:85:9e:6d:83:12:84:50:1a:8e:48:
            b1:fa:86:8c:a7:80:b9:be:52:ec:a6:ca:63:47:84:
            ad:f6:74:85:82:16:7e:4e:36:40:0a:74:2c:20:a9:
            6a:0e:6a:7f:35:cf:70:71:63:7d:e9:43:67:81:4c:
            ea:b5:1e:b7:4c:a3:35:08:7b:21:0d:2a:73:07:63:
            9d:8d:75:bf:1f:d4:8e:e6:67:60:75:f7:ea:0a:7a:
```

```
                6c:90:af:92:45:e0:62:05:9a:8a:10:98:dc:7c:54:
                8b:e4:61:95:3b:04:fc:10:50:ef:80:45:ba:5e:84:
                97:76:c1:20:25:c1:92:1d:89:0a:f7:55:62:64:fa:
                e8:69:a2:62:4c:67:d3:08:d9:61:b5:3d:16:54:b6:
                b7:44:8d:59:2b:90:d4:e9:fb:c7:7d:87:58:c3:12:
                ac:33:78:00:50:ba:07:05:b3:b9:01:1a:63:55:6c:
                e1:7a:ec:a3:07:ae:3b:02:83:a1:69:e0:c3:dc:2d:
                61:e9:b2:e3:b3:71:c8:a6:cf:da:fb:3e:99:c7:e5:
                71:b9:c9:17:d4:ed:bc:a0:47:54:09:8c:6e:6d:53:
                9a:2c:c9:68:c6:6f:f1:3d:91:1a:24:43:77:7d:91:
                69:4b
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Subject Alternative Name:
            DNS:example.com, URI:sip:example.com
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Subject Key Identifier:
            CC:06:59:5B:8B:5E:D6:0D:F2:05:4D:1B:68:54:1E:FC:F9:43:19:17
        X509v3 Authority Key Identifier:
            95:45:7E:5F:2B:EA:65:98:12:91:04:F3:63:C7:68:9A:58:16:77:27

        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
        X509v3 Extended Key Usage:
            TLS Web Server Authentication, 1.3.6.1.5.5.7.3.20
        Signature Algorithm: sha1WithRSAEncryption
    6a:9a:d1:db:00:4b:90:86:b0:53:ea:6f:30:31:89:1e:9b:09:
    14:bd:6f:b9:02:aa:6f:58:ee:30:03:b8:a1:fd:b3:41:72:ff:
    b3:0d:cb:76:a7:17:c6:57:38:06:13:e5:f3:e4:30:17:4d:f7:
    97:b5:f3:74:e9:81:f8:f4:55:a3:0d:f5:82:38:c3:98:43:52:
    1f:84:cd:1a:b4:a3:45:9f:3d:e2:31:fd:cb:a2:ad:ed:60:7d:
    fa:d2:aa:49:2f:41:a9:80:01:bb:ed:b6:75:c9:97:69:7f:0c:
    91:60:f1:c4:5a:36:e8:5c:ac:e1:a8:e7:9a:55:e5:e0:cd:01:
    f4:de:93:f4:38:6c:c1:71:d2:fd:cd:1b:5d:25:eb:90:7b:31:
    41:e7:37:0e:e5:c0:01:48:91:f7:34:dd:c6:1f:74:e6:34:34:
    e6:cd:93:0f:3f:ce:94:ad:91:d9:e2:72:b1:9f:1d:d3:a5:7d:
    5e:e2:a4:56:c5:b1:71:4d:10:0a:5d:a6:56:e6:57:1f:48:a5:
    5c:75:67:ea:ab:35:3e:f6:b6:fa:c1:f3:8a:c1:80:71:32:18:
    6c:33:b5:fa:16:5a:16:e1:a1:6c:19:67:f5:45:68:64:6f:b2:
    31:dc:e3:5a:1a:b2:d4:87:89:96:fd:87:ba:38:4e:0a:19:07:
    03:4b:9b:b1
```

The example host certificate above, as well as all the others presented in this document, are signed directly by a root CA. These certificate chains have a length equal to two: the root CA and the host certificate. Non-root CAs exist and may also sign certificates. The certificate chains presented by hosts with certificates signed by

non-root CAs will have a length greater than two. For more details on how certificate chains are validated, see Sections 6.1 and 6.2 of [RFC5280].

2.3. User Certificates

User certificates are used by many applications to establish user identity. The user certificate for fluffy@example.com is shown below. Note that the Subject Alternative Name has a list of names with different URL types such as a sip, im, or pres URL. This is necessary for interoperating with a Common Profile for Instant Messaging (CPIM) gateway. In this example, example.com is the domain for fluffy. The message could be coming from any host in *.example.com, and the address-of-record (AOR) in the user certificate would still be the same. The others are shown in Appendix B.1. These certificates make use of the Extended Key Usage (EKU) extension discussed in [RFC5924]. Note that the X509v3 Extended Key Usage attribute refers to the SIP OID introduced in [RFC5924], which is 1.3.6.1.5.5.7.3.20.

```
Version: 3 (0x2)
Serial Number:
    96:a3:84:17:4e:ef:8a:4d
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, ST=California, L=San Jose, O=sipit,
        OU=Sipit Test Certificate Authority
Validity
    Not Before: Feb  7 19:32:17 2011 GMT
    Not After : Jan 14 19:32:17 2111 GMT
Subject: C=US, ST=California, L=San Jose, O=sipit,
         CN=fluffy
Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (2048 bit)
        Modulus (2048 bit):
            00:a3:2c:59:0c:e9:bc:e4:ec:d3:9e:fb:99:02:ec:
            b1:36:3a:b7:d3:1d:4d:c3:3a:b6:ae:50:bd:5f:55:
            08:77:8c:7e:a4:e9:f0:68:31:28:8f:23:32:56:19:
            c3:22:97:a7:6d:fd:a7:22:2a:01:b5:af:61:bd:5f:
            7e:c1:14:e5:98:29:b4:34:4e:38:8a:26:ee:0d:da:
            db:27:b9:78:d6:ac:ac:04:78:32:98:c2:75:e7:6a:
            b7:2d:b3:3c:e3:eb:97:a5:ef:8b:59:42:50:17:7b:
            fe:a7:81:af:37:a7:e7:e3:1f:b0:8d:d0:72:2f:6c:
            14:42:c6:01:68:e1:8f:fd:56:4d:7d:cf:16:dc:aa:
            05:61:0b:0a:ca:ca:ec:51:ec:53:6e:3d:2b:00:80:
            fe:35:1b:06:0a:61:13:88:0b:44:f3:cc:fd:2b:0e:
            b4:a2:0b:a0:97:84:14:2e:ee:2b:e3:2f:c1:1a:9e:
            86:9a:78:6a:a2:4c:57:93:e7:01:26:d3:56:0d:bd:
```

```
            b0:2f:f8:da:c7:3c:01:dc:cb:2d:31:8c:6c:c6:5c:
            b4:63:e8:b2:a2:40:11:bf:ad:f8:6d:12:01:97:1d:
            47:f8:6a:15:8b:fb:27:96:73:44:46:34:d7:24:1c:
            cf:56:8d:d4:be:d6:94:5b:f0:a6:67:e3:dd:cf:b4:
            f2:d5
        Exponent: 65537 (0x10001)
X509v3 extensions:
    X509v3 Subject Alternative Name:
        URI:sip:fluffy@example.com, URI:im:fluffy@example.com,
           URI:pres:fluffy@example.com
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        85:97:09:B8:D3:55:37:24:8A:DC:DE:E3:91:72:E4:22:CF:98:87:52
    X509v3 Authority Key Identifier:
        95:45:7E:5F:2B:EA:65:98:12:91:04:F3:63:C7:68:9A:58:16:77:27

    X509v3 Key Usage:
        Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Extended Key Usage:
        E-mail Protection, 1.3.6.1.5.5.7.3.20
    Signature Algorithm: sha1WithRSAEncryption
a8:a9:8f:d8:8a:0b:88:ed:ff:4f:bf:e5:cd:8f:9e:7b:b8:e6:
f2:2c:aa:e3:23:5b:9a:71:5e:fd:20:a3:dd:d9:d3:c1:f2:e8:
f0:be:77:db:33:cc:8a:7b:4f:91:2b:8d:d6:f7:14:c3:8d:e0:
60:d3:34:50:bc:be:67:22:cd:f5:74:7b:f4:9a:68:a2:52:2b:
81:2f:46:d3:09:9f:25:c3:20:e8:10:d5:ef:38:7b:d1:17:d4:
f1:d7:54:67:56:f1:13:cf:2f:fc:8b:83:fc:14:e7:01:82:59:
83:cc:b1:8d:f0:c7:da:4e:b1:dc:cc:54:cf:6c:3b:47:47:59:
87:d9:16:ec:af:af:e1:12:13:23:1e:0a:db:f5:b5:ff:5d:ab:
15:0e:e3:25:91:00:0e:90:db:d8:07:11:90:81:01:3a:48:a8:
aa:9e:b0:62:d3:36:f0:0c:b7:2f:a7:17:92:52:36:29:14:0a:
d6:65:86:67:73:74:6e:aa:3c:ee:47:38:1e:c8:6e:06:81:85:
1c:2e:f0:b6:04:7d:6c:38:db:81:9c:b8:07:e3:07:be:f5:2f:
09:68:63:04:6b:87:0e:36:b9:a1:a3:fb:c8:30:0c:a0:63:8d:
6d:ab:0a:f8:44:b0:78:19:1a:38:7e:fa:6a:a1:d4:4b:4b:75:
75:bf:6f:09
```

Versions of these certificates that do not make use of EKU are also included in Appendix B.2

3. Call Flow with Message Over TLS

3.1. TLS with Server Authentication

The flow below shows the edited SSLDump output of the host example.com forming a TLS [RFC5246] connection to example.net. In this example, mutual authentication is not used. Note that the client proposed three protocol suites including TLS_RSA_WITH_AES_128_CBC_SHA defined in [RFC5246]. The certificate returned by the server contains a Subject Alternative Name that is set to example.net. A detailed discussion of TLS can be found in SSL and TLS [EKR-TLS]. For more details on the SSLDump tool, see the SSLDump Manual [ssldump-manpage].

This example does not use the Server Extended Hello (see [RFC5246]).

```
New TCP connection #1: example.com(50738) <-> example.net(5061)
1 1  0.0004 (0.0004)  C>SV3.1(101)  Handshake
      ClientHello
        Version 3.1
        random[32]=
          4c 09 5b a7 66 77 eb 43 52 30 dd 98 4d 09 23 d3
          ff 81 74 ab 04 69 bb 79 8c dc 59 cd c2 1f b7 ec
        cipher suites
        TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
        TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
        TLS_DHE_RSA_WITH_AES_256_SHA
        TLS_RSA_WITH_AES_256_CBC_SHA
        TLS_DSS_RSA_WITH_AES_256_SHA
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
        TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
        TLS_DHE_RSA_WITH_AES_128_CBC_SHA
        TLS_RSA_WITH_AES_128_CBC_SHA
        TLS_DHE_DSS_WITH_AES_128_CBC_SHA
        TLS_ECDHE_RSA_WITH_DES_192_CBC3_SHA
        TLS_ECDH_RSA_WITH_DES_192_CBC3_SHA
        TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
        TLS_RSA_WITH_3DES_EDE_CBC_SHA
        TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
        TLS_ECDHE_RSA_WITH_RC4_128_SHA
        TLS_ECDH_RSA_WITH_RC4_128_SHA
        TLS_RSA_WITH_RC4_128_SHA
        TLS_RSA_WITH_RC4_128_MD5
        TLS_DHE_RSA_WITH_DES_CBC_SHA
        TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
        TLS_RSA_WITH_DES_CBC_SHA
        TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
        TLS_DHE_DSS_WITH_DES_CBC_SHA
```

```
          TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
          TLS_RSA_EXPORT_WITH_RC4_40_MD5
          compression methods
                    NULL
   1 2  0.0012 (0.0007)  S>CV3.1(48)  Handshake
         ServerHello
           Version 3.1
           random[32]=
             4c 09 5b a7 30 87 74 c7 16 98 24 d5 af 35 17 a7
             ef c3 78 0c 94 d4 94 d2 7b a6 3f 40 04 25 f6 e0
           session_id[0]=

           cipherSuite         TLS_RSA_WITH_AES_256_CBC_SHA
           compressionMethod                     NULL
   1 3  0.0012 (0.0000)  S>CV3.1(1858)  Handshake
         Certificate
   1 4  0.0012 (0.0000)  S>CV3.1(14)  Handshake
         CertificateRequest
           certificate_types                   rsa_sign
           certificate_types                   dss_sign
           certificate_types                 unknown value
         ServerHelloDone
   1 5  0.0043 (0.0031)  C>SV3.1(7)  Handshake
         Certificate
   1 6  0.0043 (0.0000)  C>SV3.1(262)  Handshake
         ClientKeyExchange
   1 7  0.0043 (0.0000)  C>SV3.1(1)  ChangeCipherSpec
   1 8  0.0043 (0.0000)  C>SV3.1(48)  Handshake
   1 9  0.0129 (0.0085)  S>CV3.1(170)  Handshake
   1 10 0.0129 (0.0000)  S>CV3.1(1)  ChangeCipherSpec
   1 11 0.0129 (0.0000)  S>CV3.1(48)  Handshake
   1 12 0.0134 (0.0005)  C>SV3.1(32)  application_data
   1 13 0.0134 (0.0000)  C>SV3.1(496)  application_data
   1 14 0.2150 (0.2016)  S>CV3.1(32)  application_data
   1 15 0.2150 (0.0000)  S>CV3.1(336)  application_data
   1 16 12.2304 (12.0154)  S>CV3.1(32)  Alert
   1    12.2310 (0.0005)  S>C  TCP FIN
   1 17 12.2321 (0.0011)  C>SV3.1(32)  Alert
```

3.2. MESSAGE Transaction Over TLS

Once the TLS session is set up, the following MESSAGE request (as defined in [RFC3428] is sent from fluffy@example.com to kumiko@example.net. Note that the URI has a SIPS URL and that the VIA indicates that TLS was used. In order to format this document, the <allOneLine> convention from [RFC4475] is used to break long lines. The actual message does not contain the line breaks contained within those tags.

```
MESSAGE sips:kumiko@example.net:5061 SIP/2.0
<allOneLine>
Via: SIP/2.0/TLS 192.0.2.2:15001;
     branch=z9hG4bK-d8754z-c785a077a9a8451b-1---d8754z-;
     rport=50738
</allOneLine>
Max-Forwards: 70
To: <sips:kumiko@example.net:5061>
From: <sips:fluffy@example.com:15001>;tag=1a93430b
Call-ID: OTZmMDE2OWNlYTVjNDkzYzBhMWRlMDU4NDExZmU4ZTQ.
CSeq: 4308 MESSAGE
<allOneLine>
Accept: multipart/signed, text/plain, application/pkcs7-mime,
        application/sdp, multipart/alternative
</allOneLine>
Content-Type: text/plain
Content-Length: 6

Hello!
```

When a User Agent (UA) goes to send a message to example.com, the UA can see if it already has a TLS connection to example.com and if it does, it may send the message over this connection. A UA should have some scheme for reusing connections as opening a new TLS connection for every message results in awful performance. Implementers are encouraged to read [RFC5923] and [RFC3263].

The response is sent from example.net to example.com over the same TLS connection. It is shown below.

```
SIP/2.0 200 OK
<allOneLine>
Via: SIP/2.0/TLS 192.0.2.2:15001;
     branch=z9hG4bK-d8754z-c785a077a9a8451b-1---d8754z-;
     rport=50738
</allOneLine>
To: <sips:kumiko@example.net:5061>;tag=0d075510
From: <sips:fluffy@example.com:15001>;tag=1a93430b
Call-ID: OTZmMDE2OWNlYTVjNDkzYzBhMWRlMDU4NDExZmU4ZTQ.
CSeq: 4308 MESSAGE
Content-Length: 0
```

4. Call Flow with S/MIME-Secured Message

4.1. MESSAGE Request with Signed Body

Below is an example of a signed message. The values on the Content-Type line (multipart/signed) and on the Content-Disposition line have been broken across lines to fit on the page, but they are not broken across lines in actual implementations.

```
MESSAGE sip:kumiko@example.net SIP/2.0
<allOneLine>
Via: SIP/2.0/TCP 192.0.2.2:15001;
     branch=z9hG4bK-d8754z-3a922b6dc0f0ff37-1---d8754z-;
     rport=50739
</allOneLine>
Max-Forwards: 70
To: <sip:kumiko@example.net>
From: <sip:fluffy@example.com>;tag=ef6bad5e
Call-ID: N2NiZjI0NjRjNDQ0MTY1NDRjNWNmMGU1MDA2MDRhYmI.
CSeq: 8473 MESSAGE
<allOneLine>
Accept: multipart/signed, text/plain, application/pkcs7-mime,
        application/sdp, multipart/alternative
</allOneLine>
<allOneLine>
Content-Type: multipart/signed;boundary=3b515e121b43a911;
              micalg=sha1;protocol="application/pkcs7-signature"
</allOneLine>
Content-Length: 774

--3b515e121b43a911
Content-Type: text/plain
Content-Transfer-Encoding: binary

Hello!
--3b515e121b43a911
Content-Type: application/pkcs7-signature;name=smime.p7s
<allOneLine>
Content-Disposition: attachment;handling=required;
                     filename=smime.p7s
</allOneLine>
Content-Transfer-Encoding: binary

*****************
* BINARY BLOB 1 *
*****************
--3b515e121b43a911--
```

It is important to note that the signature ("BINARY BLOB 1") is computed over the MIME headers and body, but excludes the multipart boundary lines. The value on the Message-body line ends with CRLF. The CRLF is included in the boundary and is not part of the signature computation. To be clear, the signature is computed over data starting with the "C" in the "Content-Type" and ending with the "!" in the "Hello!".

```
Content-Type: text/plain
Content-Transfer-Encoding: binary

Hello!
```

Following is the ASN.1 parsing of encrypted contents referred to above as "BINARY BLOB 1". Note that at address 30, the hash for the signature is specified as SHA-1. Also note that the sender's certificate is not attached as it is optional in [RFC5652].

```
 0  472: SEQUENCE {
 4    9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
15  457:   [0] {
19  453:     SEQUENCE {
23    1:       INTEGER 1
26   11:       SET {
28    9:         SEQUENCE {
30    5:           OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
37    0:           NULL
       :           }
       :         }
39   11:       SEQUENCE {
41    9:         OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
       :         }
52  420:       SET {
56  416:         SEQUENCE {
60    1:           INTEGER 1
63  125:           SEQUENCE {
65  112:             SEQUENCE {
67   11:               SET {
69    9:                 SEQUENCE {
71    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
76    2:                   PrintableString 'US'
       :                   }
       :                 }
80   19:               SET {
82   17:                 SEQUENCE {
84    3:                   OBJECT IDENTIFIER
       :                     stateOrProvinceName (2 5 4 8)
89   10:                   UTF8String 'California'
```

```
      :                   }
      :                 }
  101  17:               SET {
  103  15:                 SEQUENCE {
  105   3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
  110   8:                   UTF8String 'San Jose'
      :                   }
      :                 }
  120  14:               SET {
  122  12:                 SEQUENCE {
  124   3:                   OBJECT IDENTIFIER
      :                     organizationName (2 5 4 10)
  129   5:                   UTF8String 'sipit'
      :                   }
      :                 }
  136  41:               SET {
  138  39:                 SEQUENCE {
  140   3:                   OBJECT IDENTIFIER
      :                     organizationalUnitName (2 5 4 11)
  145  32:                   UTF8String 'Sipit Test Certificate
                                        Authority'
      :                   }
      :                 }
      :               }
  179   9:             INTEGER 00 96 A3 84 17 4E EF 8A 4D
      :             }
  190   9:           SEQUENCE {
  192   5:             OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
  199   0:             NULL
      :             }
  201  13:           SEQUENCE {
  203   9:             OBJECT IDENTIFIER
      :               rsaEncryption (1 2 840 113549 1 1 1)
  214   0:             NULL
      :             }
  216 256:           OCTET STRING
      :             74 4D 21 39 D6 E2 E2 2C 30 5A AA BC 4E 60 8D 69
      :             A7 E5 79 50 1A B1 7D 4A D3 C1 03 9F 19 7D A2 76
      :             97 B3 CE 30 CD 62 4B 96 20 35 DB C1 64 D9 33 92
      :             96 CD 28 03 98 6E 2C 0C F6 8D 93 40 F2 88 DA 29
      :             AD 0B C2 0E F9 D3 6A 95 2C 79 6E C2 3D 62 E6 54
      :             A9 1B AC 66 DB 16 B7 44 6C 03 1B 71 9C EE C9 EC
      :             4D 93 B1 CF F5 17 79 C5 C8 BA 2F A7 6C 4B DC CF
      :             62 A3 F3 1A 1B 24 E4 40 66 3C 4F 87 86 BF 09 6A
      :             7A 43 60 2B FC D8 3D 2B 57 17 CB 81 03 2A 56 69
      :             81 82 FA 78 DE D2 3A 2F FA A3 C5 EA 8B E8 0C 36
      :             1B BC DC FD 1B 8C 2E 0F 01 AF D9 E1 04 0E 4E 50
      :             94 75 7C BD D9 0B DD AA FA 36 E3 EC E4 A5 35 46
```

```
        :             BE A2 97 1D AD BA 44 54 3A ED 94 DA 76 4A 51 BA
        :             A4 7D 7A 62 BF 2A 2F F2 5C 5A FE CA E6 B9 DC 5D
        :             EA 26 F2 35 17 19 20 CE 97 96 4E 72 9C 72 FD 1F
        :             68 C1 6A 5C 86 42 F2 ED F2 70 65 4C C7 44 C5 7C
        :           }
        :         }
        :       }
        :     }
        :   }
```

SHA-1 parameters may be omitted entirely, instead of being set to NULL, as mentioned in [RFC3370]. The above dump of Blob 1 has SHA-1 parameters set to NULL. Below are the same contents signed with the same key, but omitting the NULL according to [RFC3370]. This is the preferred encoding. This is covered in greater detail in Section 5.

```
 0  468: SEQUENCE {
 4    9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
15  453:   [0] {
19  449:     SEQUENCE {
23    1:       INTEGER 1
26    9:       SET {
28    7:         SEQUENCE {
30    5:           OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
       :           }
       :         }
37   11:       SEQUENCE {
39    9:         OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
       :         }
50  418:       SET {
54  414:         SEQUENCE {
58    1:           INTEGER 1
61  125:           SEQUENCE {
63  112:             SEQUENCE {
65   11:               SET {
67    9:                 SEQUENCE {
69    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
74    2:                   PrintableString 'US'
       :                   }
       :                 }
78   19:               SET {
80   17:                 SEQUENCE {
82    3:                   OBJECT IDENTIFIER
       :                     stateOrProvinceName (2 5 4 8)
87   10:                   UTF8String 'California'
       :                   }
       :                 }
99   17:               SET {
```

```
    101   15:                 SEQUENCE {
    103    3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
    108    8:                   UTF8String 'San Jose'
          :                   }
          :                 }
    118   14:                 SET {
    120   12:                   SEQUENCE {
    122    3:                     OBJECT IDENTIFIER
          :                     organizationName (2 5 4 10)
    127    5:                     UTF8String 'sipit'
          :                   }
          :                 }
    134   41:                 SET {
    136   39:                   SEQUENCE {
    138    3:                     OBJECT IDENTIFIER
          :                     organizationalUnitName (2 5 4 11)
    143   32:                     UTF8String 'Sipit Test Certificate
          :                                 Authority'
          :                   }
          :                 }
          :               }
    177    9:             INTEGER 00 96 A3 84 17 4E EF 8A 4D
          :             }
    188    7:           SEQUENCE {
    190    5:             OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
          :             }
    197   13:           SEQUENCE {
    199    9:             OBJECT IDENTIFIER
          :             rsaEncryption (1 2 840 113549 1 1 1)
    210    0:             NULL
          :             }
    212  256:           OCTET STRING
          :             74 4D 21 39 D6 E2 E2 2C 30 5A AA BC 4E 60 8D 69
          :             A7 E5 79 50 1A B1 7D 4A D3 C1 03 9F 19 7D A2 76
          :             97 B3 CE 30 CD 62 4B 96 20 35 DB C1 64 D9 33 92
          :             96 CD 28 03 98 6E 2C 0C F6 8D 93 40 F2 88 DA 29
          :             AD 0B C2 0E F9 D3 6A 95 2C 79 6E C2 3D 62 E6 54
          :             A9 1B AC 66 DB 16 B7 44 6C 03 1B 71 9C EE C9 EC
          :             4D 93 B1 CF F5 17 79 C5 C8 BA 2F A7 6C 4B DC CF
          :             62 A3 F3 1A 1B 24 E4 40 66 3C 4F 87 86 BF 09 6A
          :             7A 43 60 2B FC D8 3D 2B 57 17 CB 81 03 2A 56 69
          :             81 82 FA 78 DE D2 3A 2F FA A3 C5 EA 8B E8 0C 36
          :             1B BC DC FD 1B 8C 2E 0F 01 AF D9 E1 04 0E 4E 50
          :             94 75 7C BD D9 0B DD AA FA 36 E3 EC E4 A5 35 46
          :             BE A2 97 1D AD BA 44 54 3A ED 94 DA 76 4A 51 BA
          :             A4 7D 7A 62 BF 2A 2F F2 5C 5A FE CA E6 B9 DC 5D
          :             EA 26 F2 35 17 19 20 CE 97 96 4E 72 9C 72 FD 1F
          :             68 C1 6A 5C 86 42 F2 ED F2 70 65 4C C7 44 C5 7C
```

```
          :           }
          :         }
          :       }
          :     }
          :   }
```

4.2. MESSAGE Request with Encrypted Body

Below is an example of an encrypted text/plain message that says "Hello!". The binary encrypted contents have been replaced with the block "BINARY BLOB 2".

```
MESSAGE sip:kumiko@example.net SIP/2.0
<allOneLine>
Via: SIP/2.0/TCP 192.0.2.2:15001;
     branch=z9hG4bK-d8754z-c276232b541dd527-1---d8754z-;
     rport=50741
</allOneLine>
Max-Forwards: 70
To: <sip:kumiko@example.net>
From: <sip:fluffy@example.com>;tag=7a2e3025
Call-ID: MDYyMDhhODA3NWE2ZjEyYzAwOTZlMjExNWI2ZWQwZGM.
CSeq: 3260 MESSAGE
<allOneLine>
Accept: multipart/signed, text/plain, application/pkcs7-mime,
        application/sdp, multipart/alternative
</allOneLine>
<allOneLine>
Content-Disposition: attachment;handling=required;
                     filename=smime.p7
</allOneLine>
Content-Transfer-Encoding: binary
<allOneLine>
Content-Type: application/pkcs7-mime;smime-type=enveloped-data;
              name=smime.p7m
</allOneLine>
Content-Length: 565

*****************
* BINARY BLOB 2 *
*****************
```

Following is the ASN.1 parsing of "BINARY BLOB 2". Note that at address 454, the encryption is set to aes128-CBC.

```
 0  561: SEQUENCE {
 4    9:   OBJECT IDENTIFIER envelopedData (1 2 840 113549 1 7 3)
15  546:   [0] {
```

```
  19  542:     SEQUENCE {
  23    1:       INTEGER 0
  26  409:       SET {
  30  405:         SEQUENCE {
  34    1:           INTEGER 0
  37  125:           SEQUENCE {
  39  112:             SEQUENCE {
  41   11:               SET {
  43    9:                 SEQUENCE {
  45    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
  50    2:                   PrintableString 'US'
       :                   }
       :                 }
  54   19:               SET {
  56   17:                 SEQUENCE {
  58    3:                   OBJECT IDENTIFIER
       :                     stateOrProvinceName (2 5 4 8)
  63   10:                   UTF8String 'California'
       :                   }
       :                 }
  75   17:               SET {
  77   15:                 SEQUENCE {
  79    3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
  84    8:                   UTF8String 'San Jose'
       :                   }
       :                 }
  94   14:               SET {
  96   12:                 SEQUENCE {
  98    3:                   OBJECT IDENTIFIER
       :                     organizationName (2 5 4 10)
 103    5:                   UTF8String 'sipit'
       :                   }
       :                 }
 110   41:               SET {
 112   39:                 SEQUENCE {
 114    3:                   OBJECT IDENTIFIER
       :                     organizationalUnitName (2 5 4 11)
 119   32:                   UTF8String 'Sipit Test Certificate
       :                                 Authority'
       :                   }
       :                 }
       :               }
 153    9:             INTEGER 00 96 A3 84 17 4E EF 8A 4E
       :             }
 164   13:           SEQUENCE {
 166    9:             OBJECT IDENTIFIER
       :             rsaEncryption (1 2 840 113549 1 1 1)
 177    0:             NULL
```

```
      :             }
  179  256:          OCTET STRING
      :             B9 12 8F 32 AB 4A E2 38 C1 E0 53 69 88 D6 25 E7
      :             40 03 B1 DE 79 21 A3 E8 23 5A 1B CB FB 58 F4 97
      :             48 A7 C8 F0 3D DF 41 A3 5A 90 32 70 82 FA B0 DE
      :             D8 94 7C 6C 2E 01 FE 33 BD 62 CB 07 4F 58 DE 6F
      :             EA 3F EF B4 FB 46 72 58 9A 88 A0 85 BC 23 D7 C8
      :             09 0B 90 8D 4A 5F 3F 96 7C AC D4 E2 19 E8 02 B6
      :             0E F3 0D F2 91 4A 67 A9 EE 51 6A 97 D7 86 6D EC
      :             78 6E C6 E0 83 7C E1 00 1F 5A 40 59 60 0C D7 EB
      :             A3 FB 04 B3 C9 A5 EB 79 ED B3 56 F8 F6 51 B2 5E
      :             58 E2 D8 17 28 33 A6 B8 35 8C 0E 14 7F 90 D0 7B
      :             03 00 6C 3D 81 29 F5 D7 E5 AC 75 5E E0 F0 DD E3
      :             3E B2 06 97 D6 49 A9 CB 38 08 F1 84 05 F5 C0 BC
      :             55 A6 D4 C9 D8 FD A4 AC 40 9F 9D 51 5B F7 3A C3
      :             C3 CD 3A E7 6D 21 05 D0 50 75 4F 14 D8 77 76 C6
      :             13 A6 48 12 7B 25 CC 22 5D 73 BD 40 E4 15 02 A2
      :             39 4A CB D9 55 08 A4 EE 4E 8A 5E BA C4 4A 46 9C
      :           }
      :         }
  439  124:      SEQUENCE {
  441    9:        OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
  452   29:        SEQUENCE {
  454    9:          OBJECT IDENTIFIER
      :             aes128-CBC (2 16 840 1 101 3 4 1 2)
  465   16:          OCTET STRING
      :             CA 35 CA BD 1E 78 83 D9 20 6C 47 B9 9F DC 91 88
      :           }
  483   80:        [0]
      :           1B AE 12 C4 0E 55 96 AB 99 CC 1C 7F B5 98 A4 BF
      :           D2 D8 7F 94 BB B5 38 05 59 F2 38 A1 CD 29 75 17
      :           1D 63 1B 0B B0 2D 88 06 7F 78 80 F3 5A 3E DC 35
      :           BF 22 1E 03 32 59 98 DA FD 81 5F D9 41 63 3A 18
      :           FD B5 84 14 01 46 0B 40 EB 56 29 86 47 8B D1 EE
      :         }
      :       }
      :     }
      :   }
```

4.3. MESSAGE Request with Encrypted and Signed Body

In the example below, some of the header values have been split across multiple lines. Where the lines have been broken, the <allOneLine> convention has been used. This was only done to make it fit in the RFC format. Specifically, the application/pkcs7-mime Content-Type line is one line with no whitespace between the "mime;" and the "smime-type". The values are split across lines for formatting, but are not split in the real message. The binary

encrypted content has been replaced with "BINARY BLOB 3", and the binary signed content has been replaced with "BINARY BLOB 4".

```
MESSAGE sip:kumiko@example.net SIP/2.0
<allOneLine>
Via: SIP/2.0/TCP 192.0.2.2:15001;
     branch=z9hG4bK-d8754z-97a26e59b7262b34-1---d8754z-;
     rport=50742
</allOneLine>
Max-Forwards: 70
To: <sip:kumiko@example.net>
From: <sip:fluffy@example.com>;tag=379f5b27
Call-ID: MjYwMzdjYTY3YWRkYzgzMjU0MGI4Mzc2Njk1YzJlNzE.
CSeq: 5449 MESSAGE
<allOneLine>
Accept: multipart/signed, text/plain, application/pkcs7-mime,
        application/sdp, multipart/alternative
</allOneLine>
<allOneLine>
Content-Type: multipart/signed;boundary=e8df6e1ce5d1e864;
              micalg=sha1;protocol="application/pkcs7-signature"
</allOneLine>
Content-Length: 1455

--e8df6e1ce5d1e864
<allOneLine>
Content-Type: application/pkcs7-mime;smime-type=enveloped-data;
              name=smime.p7m
</allOneLine>
<allOneLine>
Content-Disposition: attachment;handling=required;
                     filename=smime.p7
</allOneLine>
Content-Transfer-Encoding: binary

*****************
* BINARY BLOB 3 *
*****************
--e8df6e1ce5d1e864
Content-Type: application/pkcs7-signature;name=smime.p7s
<allOneLine>
Content-Disposition: attachment;handling=required;
                     filename=smime.p7s
</allOneLine>
Content-Transfer-Encoding: binary

*****************
* BINARY BLOB 4 *
```

```
   *****************
   --e8df6e1ce5d1e864--
```

Below is the ASN.1 parsing of "BINARY BLOB 3".

```
    0  561: SEQUENCE {
    4    9:   OBJECT IDENTIFIER envelopedData (1 2 840 113549 1 7 3)
   15  546:   [0] {
   19  542:     SEQUENCE {
   23    1:       INTEGER 0
   26  409:       SET {
   30  405:         SEQUENCE {
   34    1:           INTEGER 0
   37  125:           SEQUENCE {
   39  112:             SEQUENCE {
   41   11:               SET {
   43    9:                 SEQUENCE {
   45    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
   50    2:                   PrintableString 'US'
          :                   }
          :                 }
   54   19:               SET {
   56   17:                 SEQUENCE {
   58    3:                   OBJECT IDENTIFIER
          :                     stateOrProvinceName (2 5 4 8)
   63   10:                   UTF8String 'California'
          :                   }
          :                 }
   75   17:               SET {
   77   15:                 SEQUENCE {
   79    3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
   84    8:                   UTF8String 'San Jose'
          :                   }
          :                 }
   94   14:               SET {
   96   12:                 SEQUENCE {
   98    3:                   OBJECT IDENTIFIER
          :                     organizationName (2 5 4 10)
  103    5:                   UTF8String 'sipit'
          :                   }
          :                 }
  110   41:               SET {
  112   39:                 SEQUENCE {
  114    3:                   OBJECT IDENTIFIER
          :                   organizationalUnitName (2 5 4 11)
  119   32:                   UTF8String 'Sipit Test Certificate
          :                               Authority'
          :                   }
```

```
      :                 }
      :               }
  153  9:              INTEGER 00 96 A3 84 17 4E EF 8A 4E
      :             }
  164  13:            SEQUENCE {
  166   9:              OBJECT IDENTIFIER
      :                 rsaEncryption (1 2 840 113549 1 1 1)
  177   0:              NULL
      :             }
  179 256:            OCTET STRING
      :               49 11 0B 11 52 A9 9D E3 AA FB 86 CB EB 12 CC 8E
      :               96 9D 85 3E 80 D2 7C C4 9B B7 81 4B B5 FA 13 80
      :               6A 6A B2 34 72 D8 C0 82 60 DA B3 43 F8 51 8C 32
      :               8B DD D0 76 6D 9C 46 73 C1 44 A0 10 FF 16 A4 83
      :               74 85 21 74 7D E0 FD 42 C0 97 00 82 A2 80 81 22
      :               9C A2 82 0A 85 F0 68 EF 9A D7 6D 1D 24 2B A9 5E
      :               B3 9A A0 3E A7 D9 1D 1C D7 42 CB 6F A5 81 66 23
      :               28 00 7C 99 6A B6 03 3F 7E F6 48 EA 91 49 35 F1
      :               FD 40 54 5D AC F7 84 EA 3F 27 43 FD DE E2 10 DD
      :               63 C4 35 4A 13 63 0B 6D 0D 9A D5 AB 72 39 69 8C
      :               65 4C 44 C4 A3 31 60 79 B9 A8 A3 A1 03 FD 41 25
      :               12 E5 F3 F8 47 CE 8C 42 D9 26 77 A5 57 AF 1A 95
      :               BF 05 A5 E9 47 F2 D1 AE DC 13 7E 1B 83 5C 8C C4
      :               1F 31 BC 59 E6 FD 6E 9A B0 91 EC 71 A6 7F 28 3E
      :               23 1B 40 E2 C0 60 CF 5E 5B 86 08 06 82 B4 B7 DB
      :               00 DD AC 3A 39 27 E2 7C 96 AD 8A E9 C3 B8 06 5E
      :             }
      :           }
  439 124:        SEQUENCE {
  441   9:          OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
  452  29:          SEQUENCE {
  454   9:            OBJECT IDENTIFIER
      :               aes128-CBC (2 16 840 1 101 3 4 1 2)
  465  16:            OCTET STRING
      :               88 9B 13 75 A7 66 14 C3 CF CD C6 FF D2 91 5D A0
      :             }
  483  80:          [0]
      :             80 0B A3 B7 57 89 B4 F4 70 AE 1D 14 A9 35 DD F9
      :             1D 66 29 46 52 40 13 E1 3B 4A 23 E5 EC AB F9 35
      :             A6 B6 A4 BE C0 02 31 06 19 C4 39 22 7D 10 4C 0D
      :             F4 96 04 78 11 85 4E 7E E3 C3 BC B2 DF 55 17 79
      :             5F F2 4E E5 25 42 37 45 39 5D F6 DA 57 9A 4E 0B
      :           }
      :         }
      :       }
      :     }
```

Below is the ASN.1 parsing of "BINARY BLOB 4".

```
 0  472: SEQUENCE {
 4    9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
15  457:   [0] {
19  453:     SEQUENCE {
23    1:       INTEGER 1
26   11:       SET {
28    9:         SEQUENCE {
30    5:           OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
37    0:           NULL
       :           }
       :         }
39   11:       SEQUENCE {
41    9:         OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
       :         }
52  420:       SET {
56  416:         SEQUENCE {
60    1:           INTEGER 1
63  125:           SEQUENCE {
65  112:             SEQUENCE {
67   11:               SET {
69    9:                 SEQUENCE {
71    3:                   OBJECT IDENTIFIER countryName (2 5 4 6)
76    2:                   PrintableString 'US'
       :                   }
       :                 }
80   19:               SET {
82   17:                 SEQUENCE {
84    3:                   OBJECT IDENTIFIER
       :                     stateOrProvinceName (2 5 4 8)
89   10:                   UTF8String 'California'
       :                   }
       :                 }
 101   17:               SET {
 103   15:                 SEQUENCE {
 105    3:                   OBJECT IDENTIFIER localityName (2 5 4 7)
 110    8:                   UTF8String 'San Jose'
       :                   }
       :                 }
 120   14:               SET {
 122   12:                 SEQUENCE {
 124    3:                   OBJECT IDENTIFIER
       :                     organizationName (2 5 4 10)
 129    5:                   UTF8String 'sipit'
       :                   }
       :                 }
 136   41:               SET {
```

```
    138   39:                 SEQUENCE {
    140    3:                   OBJECT IDENTIFIER
          :                     organizationalUnitName (2 5 4 11)
    145   32:                   UTF8String 'Sipit Test Certificate
          :                               Authority'
          :                 }
          :               }
          :             }
    179    9:           INTEGER 00 96 A3 84 17 4E EF 8A 4D
          :           }
    190    9:           SEQUENCE {
    192    5:             OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
    199    0:             NULL
          :           }
    201   13:           SEQUENCE {
    203    9:             OBJECT IDENTIFIER
          :               rsaEncryption (1 2 840 113549 1 1 1)
    214    0:             NULL
          :           }
    216  256:           OCTET STRING
          :           6E 51 AC 24 2E BA 7C A1 EE 80 A8 55 BC D4 64 5D
          :           E5 29 09 5F B2 AF AA 6F 91 D2 97 79 32 5B AF CA
          :           FE A1 73 FC E5 57 4E C6 3B 67 35 AA E4 78 1E 59
          :           93 EE 67 63 77 1E 7A 82 BC 1E 26 0F 39 75 0C A6
          :           26 92 01 6A B7 5D F0 C0 2C 51 46 FB A7 36 44 E3
          :           64 C6 11 CB 0B 6B FD F3 6D 7C FD 3E AE 2E 91 BB
          :           78 9E F4 1B A1 20 68 B9 DE D3 E3 0C FC F7 14 9A
          :           2C 64 AB 27 52 BD 52 EC 27 88 14 BD DB C3 54 C7
          :           EA 48 DB 07 E9 9B 2E C8 BE 62 A2 76 83 53 37 E8
          :           02 4B D1 86 E9 DF 2E BD 93 39 EC 2F 01 53 A0 7F
          :           1A B9 A6 31 FC E7 91 1C DB 22 4A 67 83 94 B2 4E
          :           28 A9 CD DE 4A 04 6A E0 86 90 7B 58 5F DB 7A 96
          :           96 A0 25 61 C2 58 A2 28 E5 B3 B2 F1 6D 51 06 9C
          :           78 61 0D D8 3A A7 9F A3 B5 87 0B 80 11 C2 A9 1A
          :           E5 17 1C EB 82 55 AB CD 04 E7 D9 5B 11 E8 B7 47
          :           FE FD CC B7 DB 47 6F 77 85 9E 24 D8 11 E1 E4 7D
          :         }
          :       }
          :     }
          :   }
          : }
```

5. Observed Interoperability Issues

This section describes some common interoperability problems. These were observed by the authors at SIPit interoperability events. Implementers should be careful to verify that their systems do not introduce these common problems, and, when possible, make their

clients forgiving in what they receive. Implementations should take extra care to produce reasonable error messages when interacting with software that has these problems.

Some SIP clients incorrectly only do SSLv3 and do not support TLS. See Section 26.2.1 of [RFC3261].

Many SIP clients were found to accept expired certificates with no warning or error. See Section 4.1.2.5 of [RFC5280].

When used with SIP, TLS and S/MIME provide the identity of the peer that a client is communicating with in the Subject Alternative Name in the certificate. The software checks that this name corresponds to the identity the server is trying to contact. Normative text describing path validation can be found in Section 7 of [RFC5922] and Section 6 of [RFC5280]. If a client is trying to set up a TLS connection to good.example.com and it gets a TLS connection set up with a server that presents a valid certificate but with the name evil.example.com, it will typically generate an error or warning of some type. Similarly with S/MIME, if a user is trying to communicate with sip:fluffy@example.com, one of the items in the Subject Alternate Name set in the certificate will need to match according to the certificate validation rules in Section 23 of [RFC3261] and Section 6 of [RFC5280].

Some implementations used binary MIME encodings while others used base64. It is advisable that implementations send only binary and are prepared to receive either. See Section 3.2 of [RFC5621].

In several places in this document, the messages contain the encoding for the SHA-1 digest algorithm identifier. The preferred form for encoding as set out in Section 2 of [RFC3370] is the form in which the optional AlgorithmIdentifier parameter field is omitted. However, [RFC3370] also says the recipients need to be able to receive the form in which the AlgorithmIdentifier parameter field is present and set to NULL. Examples of the form using NULL can be found in Section 4.2 of [RFC4134]. Receivers really do need to be able to receive the form that includes the NULL because the NULL form, while not preferred, is what was observed as being generated by most implementations. Implementers should also note that if the algorithm is MD5 instead of SHA-1, then the form that omits the AlgorithmIdentifier parameters field is not allowed and the sender has to use the form where the NULL is included.

The preferred encryption algorithm for S/MIME in SIP is AES as defined in [RFC3853].

Observed S/MIME interoperability has been better when UAs did not attach the senders' certificates. Attaching the certificates significantly increases the size of the messages, which should be considered when sending over UDP. Furthermore, the receiver cannot rely on the sender to always send the certificate, so it does not turn out to be useful in most situations.

Please note that the certificate path validation algorithm described in Section 6 of [RFC5280] is a complex algorithm for which all of the details matter. There are numerous ways in which failing to precisely implement the algorithm as specified in Section 6 of [RFC5280] can create a security flaw, a simple example of which is the failure to check the expiration date that is already mentioned above. It is important for developers to ensure that this validation is performed and that the results are verified by their applications or any libraries that they use.

6. Additional Test Scenarios

This section provides a non-exhaustive list of tests that implementations should perform while developing systems that use S/MIME and TLS for SIP.

Much of the required behavior for inspecting certificates when using S/MIME and TLS with SIP is currently underspecified. The non-normative recommendations in this document capture the current folklore around that required behavior, guided by both related normative works such as [RFC4474] (particularly, Section 13.4 Domain Names and Subordination) and informative works such as [RFC2818], Section 3.1. To summarize, test plans should:

- For S/MIME secured bodies, ensure that the peer's URI (address-of-record, as per [RFC3261], Section 23.3) appears in the subjectAltName of the peer's certificate as a uniformResourceIdentifier field.

- For TLS, ensure that the peer's hostname appears as described in [RFC5922]. Also:

  * ensure an exact match in a dNSName entry in the subjectAltName if there are any dNSNames in the subjectAltName. Wildcard matching is not allowed against these dNSName entries. See Section 7.1 of [RFC5922].

  * ensure that the most specific CommonName in the Subject field matches if there are no dNSName entries in the subjectAltName at all (which is not the same as there being no matching

dNSName entries). This match can be either exact, or against an entry that uses the wildcard matching character '*'.

The peer's hostname is discovered from the initial DNS query in the server location process [RFC3263].

- IP addresses can appear in subjectAltName ([RFC5280]) of the peer's certificate, e.g., "IP:192.168.0.1". Note that if IP addresses are used in subjectAltName, there are important ramifications regarding the use of Record-Route headers that also need to be considered. See Section 7.5 of [RFC5922]. Use of IP addresses instead of domain names is inadvisable.

For each of these tests, an implementation will proceed past the verification point only if the certificate is "good". S/MIME protected requests presenting bad certificate data will be rejected. S/MIME protected responses presenting bad certificate information will be ignored. TLS connections involving bad certificate data will not be completed.

1. S/MIME : Good peer certificate

2. S/MIME : Bad peer certificate (peer URI does not appear in subjectAltName)

3. S/MIME : Bad peer certificate (valid authority chain does not end at a trusted CA)

4. S/MIME : Bad peer certificate (incomplete authority chain)

5. S/MIME : Bad peer certificate (the current time does not fall within the period of validity)

6. S/MIME : Bad peer certificate (certificate, or certificate in authority chain, has been revoked)

7. S/MIME : Bad peer certificate ("Digital Signature" is not specified as an X509v3 Key Usage)

8. TLS : Good peer certificate (hostname appears in dNSName in subjectAltName)

9. TLS : Good peer certificate (no dNSNames in subjectAltName, hostname appears in Common Name (CN) of Subject)

10. TLS : Good peer certificate (CN of Subject empty, and subjectAltName extension contains an iPAddress stored in the octet string in network byte order form as specified in RFC 791 [RFC0791])

11. TLS : Bad peer certificate (no match in dNSNames or in the Subject CN)

12. TLS : Bad peer certificate (valid authority chain does not end at a trusted CA)

13. TLS : Bad peer certificate (incomplete authority chain)

14. TLS : Bad peer certificate (the current time does not fall within the period of validity)

15. TLS : Bad peer certificate (certificate, or certificate in authority chain, has been revoked)

16. TLS : Bad peer certificate ("TLS Web Server Authentication" is not specified as an X509v3 Key Usage)

17. TLS : Bad peer certificate (Neither "SIP Domain" nor "Any Extended Key Usage" specified as an X509v3 Extended Key Usage, and X509v3 Extended Key Usage is present)

7. Acknowledgments

Many thanks to the developers of all the open source software used to create these call flows. This includes the underlying crypto and TLS software used from openssl.org, the SIP stack from www.resiprocate.org, and the SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) Instant Messaging and Presence Protocol (IMPP) agent from www.sipimp.org. The TLS flow dumps were done with SSLDump from http://www.rtfm.com/ssldump. The book "SSL and TLS" [EKR-TLS] was a huge help in developing the code for these flows. It's sad there is no second edition.

Thanks to Jim Schaad, Russ Housley, Eric Rescorla, Dan Wing, Tat Chan, and Lyndsay Campbell, who all helped find and correct mistakes in this document.

Vijay Gurbani and Alan Jeffrey contributed much of the additional test scenario content.

8. Security Considerations

Implementers must never use any of the certificates provided in this document in anything but a test environment. Installing the CA root certificates used in this document as a trusted root in operational software would completely destroy the security of the system while giving the user the impression that the system was operating securely.

This document recommends some things that implementers might test or verify to improve the security of their implementations. It is impossible to make a comprehensive list of these, and this document only suggests some of the most common mistakes that have been seen at the SIPit interoperability events. Just because an implementation does everything this document recommends does not make it secure.

This document does not show any messages to check certificate revocation status (see Sections 3.3 and 6.3 of [RFC5280]) as that is not part of the SIP call flow. The expectation is that revocation status is checked regularly to protect against the possibility of certificate compromise or repudiation. For more information on how certificate revocation status can be checked, see [RFC2560] (Online Certificate Status Protocol) and [RFC5055] (Server-Based Certificate Validation Protocol).

9. References

9.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2560] Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 2560, June 1999.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3263] Rosenberg, J. and H. Schulzrinne, "Session Initiation Protocol (SIP): Locating SIP Servers", RFC 3263, June 2002.

[RFC3370] Housley, R., "Cryptographic Message Syntax (CMS) Algorithms", RFC 3370, August 2002.

[RFC3428] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[RFC3853] Peterson, J., "S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP)", RFC 3853, July 2004.

[RFC4474] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", RFC 4474, August 2006.

[RFC5055] Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, "Server-Based Certificate Validation Protocol (SCVP)", RFC 5055, December 2007.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC5621] Camarillo, G., "Message Body Handling in the Session Initiation Protocol (SIP)", RFC 5621, September 2009.

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, September 2009.

[RFC5751] Ramsdell, B. and S. Turner, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010.

[RFC5922] Gurbani, V., Lawrence, S., and A. Jeffrey, "Domain Certificates in the Session Initiation Protocol (SIP)", RFC 5922, June 2010.

[RFC5923] Gurbani, V., Mahy, R., and B. Tate, "Connection Reuse in the Session Initiation Protocol (SIP)", RFC 5923, June 2010.

[RFC5924] Lawrence, S. and V. Gurbani, "Extended Key Usage (EKU) for Session Initiation Protocol (SIP) X.509 Certificates", RFC 5924, June 2010.

[X.509] International Telecommunications Union, "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks", ITU-T Recommendation X.509 (2005), ISO/ IEC 9594-8:2005.

[X.683] International Telecommunications Union, "Information technology - Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications", ITU-T Recommendation X.683 (2002), ISO/IEC 8824-4:2002, 2002.

9.2. Informative References

[EKR-TLS] Rescorla, E., "SSL and TLS - Designing and Building Secure Systems", Addison-Wesley, ISBN 0-201-61598-3, 2001.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC4134] Hoffman, P., "Examples of S/MIME Messages", RFC 4134, July 2005.

[RFC4475] Sparks, R., Hawrylyshen, A., Johnston, A., Rosenberg, J., and H. Schulzrinne, "Session Initiation Protocol (SIP) Torture Test Messages", RFC 4475, May 2006.

[RFC4514] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names", RFC 4514, June 2006.

[ssldump-manpage] Rescorla, E., "SSLDump manpage", <http://www.rtfm.com/ssldump/Ssldump.html>.

Appendix A. Making Test Certificates

These scripts allow you to make certificates for test purposes. The certificates will all share a common CA root so that everyone running these scripts can have interoperable certificates. WARNING - these certificates are totally insecure and are for test purposes only. All the CAs created by this script share the same private key to facilitate interoperability testing, but this totally breaks the security since the private key of the CA is well known.

The instructions assume a Unix-like environment with openssl installed, but openssl does work in Windows too. OpenSSL version 0.9.8j was used to generate the certificates used in this document. Make sure you have openssl installed by trying to run "openssl". Run the makeCA script found in Appendix A.1; this creates a subdirectory called demoCA. If the makeCA script cannot find where your openssl is installed you will have to set an environment variable called OPENSSLDIR to whatever directory contains the file openssl.cnf. You can find this with a "locate openssl.cnf". You are now ready to make certificates.

To create certificates for use with TLS, run the makeCert script found in Appendix A.2 with the fully qualified domain name of the proxy you are making the certificate for, e.g., "makeCert host.example.net domain eku". This will generate a private key and a certificate. The private key will be left in a file named domain_key_example.net.pem in Privacy Enhanced Mail (PEM) format. The certificate will be in domain_cert_example.net.pem. Some programs expect both the certificate and private key combined together in a Public-Key Cryptography Standards (PKCS) #12 format file. This is created by the script and left in a file named example.net.p12. Some programs expect this file to have a .pfx extension instead of .p12 -- just rename the file if needed. A file with a certificate signing request, called example.net.csr, is also created and can be used to get the certificate signed by another CA.

A second argument indicating the number of days for which the certificate should be valid can be passed to the makeCert script. It is possible to make an expired certificate using the command "makeCert host.example.net 0".

Anywhere that a password is used to protect a certificate, the password is set to the string "password".

The root certificate for the CA is in the file root_cert_fluffyCA.pem.

For things that need DER format certificates, a certificate can be converted from PEM to DER with "openssl x509 -in cert.pem -inform PEM -out cert.der -outform DER".

Some programs expect certificates in PKCS #7 format (with a file extension of .p7c). You can convert these from PEM format to PKCS #7 with "openssl crl2pkcs7 -nocrl -certfile cert.pem -certfile demoCA/cacert.pem -outform DER -out cert.p7c".

IE (version 8), Outlook Express (version 6), and Firefox (version 3.5) can import and export .p12 files and .p7c files. You can convert a PKCS #7 certificate to PEM format with "openssl pkcs7 -in cert.p7c -inform DER -outform PEM -out cert.pem".

The private key can be converted to PKCS #8 format with "openssl pkcs8 -in a_key.pem -topk8 -outform DER -out a_key.p8c".

In general, a TLS client will just need the root certificate of the CA. A TLS server will need its private key and its certificate. These could be in two PEM files, a single file with both certificate and private key PEM sections, or a single .p12 file. An S/MIME program will need its private key and certificate, the root certificate of the CA, and the certificate for every other user it communicates with.

A.1. makeCA script

```
#!/bin/sh
set -x

rm -rf demoCA

mkdir demoCA
mkdir demoCA/certs
mkdir demoCA/crl
mkdir demoCA/newcerts
mkdir demoCA/private
# This is done to generate the exact serial number used for the RFC
echo "4902110184015C" > demoCA/serial
touch demoCA/index.txt

# You may need to modify this for where your default file is
# you can find where yours in by typing "openssl ca"
for D in /etc/ssl /usr/local/ssl /sw/etc/ssl /sw/share/ssl; do
  CONF=${OPENSSLDIR:=$D}/openssl.cnf
  [ -f ${CONF} ] && break
done
```

```
CONF=${OPENSSLDIR}/openssl.cnf


if [ ! -f $CONF  ]; then
    echo "Can not find file $CONF - set your OPENSSLDIR variable"
    exit
fi

cp $CONF openssl.cnf

cat >> openssl.cnf  <<EOF
[ sipdomain_cert ]
subjectAltName=\${ENV::ALTNAME}
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
keyUsage = nonRepudiation,digitalSignature,keyEncipherment
extendedKeyUsage=serverAuth,1.3.6.1.5.5.7.3.20

[ sipdomain_req ]
basicConstraints = CA:FALSE
subjectAltName=\${ENV::ALTNAME}
subjectKeyIdentifier=hash

[ sipuser_cert ]
subjectAltName=\${ENV::ALTNAME}
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
keyUsage = nonRepudiation,digitalSignature,keyEncipherment
extendedKeyUsage=emailProtection,1.3.6.1.5.5.7.3.20

[ sipuser_req ]
basicConstraints = CA:FALSE
subjectAltName=\${ENV::ALTNAME}
subjectKeyIdentifier=hash

[ sipdomain_noeku_cert ]
subjectAltName=\${ENV::ALTNAME}
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
keyUsage = nonRepudiation,digitalSignature,keyEncipherment

[ sipdomain_noeku_req ]
basicConstraints = CA:FALSE
subjectAltName=\${ENV::ALTNAME}
subjectKeyIdentifier=hash
```

```
[ sipuser_noeku_cert ]
subjectAltName=\${ENV::ALTNAME}
basicConstraints=CA:FALSE
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer
keyUsage = nonRepudiation,digitalSignature,keyEncipherment

[ sipuser_noeku_req ]
basicConstraints = CA:FALSE
subjectAltName=\${ENV::ALTNAME}
subjectKeyIdentifier=hash

EOF


cat > demoCA/private/cakey.pem <<EOF
-----BEGIN ENCRYPTED PRIVATE KEY-----
MIIFDjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIlwtc771DlNUCAggA
MBQGCCqGSIb3DQMHBAhRD3Z1i2TavwSCBMgXoXo0H/dTplHwnqfW7Uh1dr776z7B
lsNxlenMA6lYmALF/4E1tqOE2/aEbr8W3wTVjNpew9r5TBsbA1I9/FMMe+USc1ra
5pIdDLx7ynzHvxcUWJ1xbWGeLcEmXGOvzkwW/oOg49Yq1ce1GtlLSV2L7Wi93TUQ
Q8i5l0X0xjx7cB7kaHTOTyaN0sxUE3qlQ2sXTbbHWUfIaNpEZUI5ITrDUflfMnxb
RogQGv+5owsM7zwzfyGz3QocM9WaZwKFOEOqBvEfGaaZ9ml+cn1Rz/1Id7tSBlRH
3ucN2mGdEVIUvzSACZ9LPuIO7WBGM56enDRsqZji4WfqDHdXa4gkJKqPEJeBnLVA
jxCmLJSyikM25kHDm8LWuOckO/Rk+7999h13Qv1Ynm7yCincorqdlTrAdmq1Z8Tj
QPgXioTlx6++6yxiDCV7Mwkydox31K9y/Tf2cZ//dWuf/lfMaaq8HfpSNl4RKqsz
ufL41K5sCzPRIugUdooUQSGPC0JgcskPcifT6zvrI62KLPFVrwG5HT9PdevQvC6O
VgglxbEGJ7I4vllzmY62/0LtQKIA6bh8pszvvmHjGo9s+f+p7KJVYygEHNEmRTm+
8M2owk67033sV6IClDOAdRL8siTHmcmM+r1x9VVIppsDrzjqQqYVGYBbjEJW8eQp
t7kAjuN48tDD1mS8E6DstPv/6S0AjzAqCbjkuPJ0WU5fD1cY+iTpo9vcunohcj+i
KVXsM34wOsBpMBjFQ+Aww5bsIkEV1liOYLav1F7/BvP2s0gc3puM5W35y1cbKLu2
ThJV7mIWoV770aQYpJba0UAk9OzBVEvPNahrDI1NucbEkFrhN2pfnOs7k4UvrjiK
uknKrm3gocDOdstyMZX81Beyj06NhpcJH+bOSvROk/d68aAsapy6qS9hLijNNbcd
itQ/fo+1o9MDujT/huj7ZFqdzNM3KA6vxf0kmmVM+GJbYke+cjXk6WB80lF9lYcB
0pWPd+fgwFL252FUoFcjvUWFXkvbR1+IMkv6sNdKcXHHazAE6nl6yPl9bVwCaS1I
WNqEfHntblNZbeW+3qH8ov1ZXVCqEmaHkajSAhFJKXCgpSXaIx2FSntzpVFbRpnw
Yd9eml9xwgE3l9aRuvR6p61fd051LzCh7KjvorV1CemPUT6YRBamFNCBoT7cqjhE
kqMQfowKkMEY0p2dzMnGzsSPKk10nI53RgPyD/8FT5dPuq073SyjxTKhAbvl+kVl
lrfZ6b7P/UKwLBCT3bLG6uU/Es84euWN+U2JXIADPoCcVeWrUqkf4j368c2Z8Zdd
A27X4ZJ+q+YfsFNiOA7vshHi3Am3gBzQhEEGsRdzgkf8qmtlRGhq/823GEexoUfu
8SiOOjoU08HGAkTtPWjV5+0C6Q6RW9SmNMwz7msZHoKTQ8kz2LKXUwb6DBwWcw6/
UTUgzVXqhA8HmjsnVe9ftDKL66v9zlp4RVRdDzm4TYUybYh5uigFbjJFLlnJnJho
TcnusHO80Cxgs64khLRzM46Oi+JSEPv7o7zHcfWNOVtNW908EKCubtEDZtnQn9VC
0Sky9R/WzunaLlG3LZ3BRUhWpyyvdNxlNq3ie4tcRMlXIEe14UZN0sPCKZY//NEn
BEc=
-----END ENCRYPTED PRIVATE KEY-----
EOF
```

```
cat > demoCA/cacert.pem <<EOF
-----BEGIN CERTIFICATE-----
MIIDtTCCAp2gAwIBAgIJAJajhBdO74pMMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDEyNzE4MzYwNVoYDzIxMTEwMTAzMTgzNjA1WjBwMQsw
CQYDVQQGEwJVUzETMBEGA1UECAwKQ2FsaWZvcm5pYTERMA8GA1UEBwwIU2FuIEpv
c2UxDjAMBgNVBAoMBXNpcGl0MSkwJwYDVQQLDCBTaXBpdCBUZXN0IENlcnRpZmlj
YXRlIEF1dGhvcml0eTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKsf
kWHxHMXNpnsWm7cUeeQwnpjQ7Ae3vXfX0fVbLOLu5rGw8IX6pbzLzM9pLE/8UO+d
MSvAWer7ZG8fVac9/XDSVtsUmReScKwm+DRBcNnAA5FqutERj6wSMd65GXCNXad9
ixnMQD+u/94f25SzRndsrq7/PtaEW8LeCyZl0JHHcEvHCkq/x5cE3bpYR8vgKyN2
h2XFVTQQqycfHPgwPbCbyqKBcky9YP73If4L2wvb6VsBNtQoFWt569CRGyFZuA6q
v9WxbHA3oz+lfQ6VRvb2WGeDdUI3GAukQTmyL2yALHjspQ++nBD4wAsNc5meDdeX
UMvMRTQjSUGFIiStKcMCAwEAAaNQME4wHQYDVR0OBBYEFJVFfl8r6mWYEpEE82PH
aJpYFncnMB8GA1UdIwQYMBaAFJVFfl8r6mWYEpEE82PHaJpYFncnMAwGA1UdEwQF
MAMBAf8wDQYJKoZIhvcNAQEFBQADggEBAAZfnq6gmry1uVt+lzPM32OYmJTLDWap
g+iqWCpZoZ5HMaavXD+iJYb43wWSt9tpoWlyh2bFqzWJATcZyXTrCdE/iHskE0LK
LftF5sxL+CF48/WX7AmSJKLw5pSNl0oAlAC9JbgXLFJTXcxcSKShHS32UFUTpNOy
ovTxuW1IXlzz3uD8WQmh2RRhZb/YP7m6LnztXCSba8qqX/HBHrCo2oIP+0xxO017
OMjjiioZNEQmC+rwRzhGKGUE4gFS3ew95fVTdHd0dW3G2cIKrDu4mFxVUzR0Uqgm
sS8wItCLt/Og3WgHM9Wut4GylFhyTnzGci+9bGn7tReoKo3XLJEGyAw=
-----END CERTIFICATE-----

EOF


# uncomment the following lines to generate your own key pair


# openssl req -newkey rsa:2048 -passin pass:password \
#     -passout pass:password -set_serial 0x96a384174eef8a4c \
#     -sha1 -x509 -keyout demoCA/private/cakey.pem \
#     -out demoCA/cacert.pem -days 36500 -config ${CONF} <<EOF
# US
# California
# San Jose
# sipit
# Sipit Test Certificate Authority
#
#
# EOF


# either randomly generate a serial number, or set it manually
# hexdump -n 4 -e '4/1 "%04u"' /dev/random > demoCA/serial
echo 96a384174eef8a4d > demoCA/serial
```

```
   openssl crl2pkcs7 -nocrl -certfile demoCA/cacert.pem \
           -outform DER -out demoCA/cacert.p7c

   cp demoCA/cacert.pem root_cert_fluffyCA.pem
```

A.2. makeCert script

```
  #!/bin/sh
  set -x

  # Make a symbolic link to this file called "makeUserCert"
  # if you wish to use it to make certs for users.

  # ExecName=$(basename $0)
  #
  # if [ ${ExecName} == "makeUserCert" ]; then
  #   ExtPrefix="sipuser"
  # elif [ ${ExecName} == "makeEkuUserCert" ]; then
  #   ExtPrefix="sipuser_eku"
  # elif [ ${ExecName} == "makeEkuCert" ]; then
  #   ExtPrefix="sipdomain_eku"
  # else
  #   ExtPrefix="sipdomain"
  # fi


  if [  $# == 3  ]; then
    DAYS=36500
  elif [ $# == 4 ]; then
    DAYS=$4
  else
    echo "Usage: makeCert test.example.org user|domain eku|noeku [days]"
    echo "       makeCert alice@example.org [days]"
    echo "days is how long the certificate is valid"
    echo "days set to 0 generates an invalid certificate"
    exit 0
  fi

  ExtPrefix="sip"${2}

  if [ $3 == "noeku" ]; then
    ExtPrefix=${ExtPrefix}"_noeku"
  fi

  DOMAIN=‘echo $1 | perl -ne ’{print "$1\n" if (/(\w+\..*)$/)}’   ‘
  USER=‘echo $1 | perl -ne ’{print "$1\n" if (/(\w+)\@(\w+\..*)$/)}’   ‘
  ADDR=$1
  echo "making cert for $DOMAIN ${ADDR}"
```

```
  if [ $2 == "user" ]; then
    CNVALUE=$USER
  else
    CNVALUE=$DOMAIN
  fi

  rm -f ${ADDR}_*.pem
  rm -f ${ADDR}.p12

  case ${ADDR} in
  *:*) ALTNAME="URI:${ADDR}" ;;
  *@*) ALTNAME="URI:sip:${ADDR},URI:im:${ADDR},URI:pres:${ADDR}" ;;
  *)   ALTNAME="DNS:${DOMAIN},URI:sip:${ADDR}" ;;
  esac

  rm -f demoCA/index.txt
  touch demoCA/index.txt
  rm -f demoCA/newcerts/*

  export ALTNAME

  openssl genrsa  -out ${ADDR}_key.pem 2048
  openssl req -new  -config openssl.cnf -reqexts ${ExtPrefix}_req \
          -sha1 -key ${ADDR}_key.pem \
          -out ${ADDR}.csr -days ${DAYS} <<EOF
  US
  California
  San Jose
  sipit

  ${CNVALUE}


  EOF

  if [ $DAYS == 0 ]; then
  openssl ca -extensions ${ExtPrefix}_cert -config openssl.cnf \
      -passin pass:password -policy policy_anything \
      -md sha1 -batch -notext -out ${ADDR}_cert.pem \
      -startdate 990101000000Z \
      -enddate 000101000000Z \
       -infiles ${ADDR}.csr
  else
  openssl ca -extensions ${ExtPrefix}_cert -config openssl.cnf \
      -passin pass:password -policy policy_anything \
      -md sha1 -days ${DAYS} -batch -notext -out ${ADDR}_cert.pem \
       -infiles ${ADDR}.csr
  fi
```

```
openssl pkcs12 -passin pass:password \
    -passout pass:password -export \
    -out ${ADDR}.p12 -in ${ADDR}_cert.pem \
    -inkey ${ADDR}_key.pem -name ${ADDR} -certfile demoCA/cacert.pem

openssl x509 -in ${ADDR}_cert.pem -noout -text

case ${ADDR} in
*@*) mv ${ADDR}_key.pem user_key_${ADDR}.pem; \
     mv ${ADDR}_cert.pem user_cert_${ADDR}.pem ;;
*)   mv ${ADDR}_key.pem domain_key_${ADDR}.pem; \
     mv ${ADDR}_cert.pem domain_cert_${ADDR}.pem ;;
esac
```

Appendix B. Certificates for Testing

This section contains various certificates used for testing in PEM format.

B.1. Certificates Using EKU

These certificates make use of the EKU specification described in [RFC5924].

Fluffy’s user certificate for example.com:

```
-----BEGIN CERTIFICATE-----
MIIEGTCCAwGgAwIBAgIJAJajhBdO74pNMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxN1oYDzIxMTEwMTE0MTkzMjE3WjBWMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MQ8wDQYDVQQDEwZmbHVmZnkwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQCjLFkM6bzk7NOe+5kC7LE2OrfTHU3DOrauUL1f
VQh3jH6k6fBoMSiPIzJWGcMil6dt/aciKgG1r2G9X37BFOWYKbQ0TjiKJu4N2tsn
uXjWrKwEeDKYwnXnarctszzj65el74tZQlAXe/6nga83p+fjH7CN0HIvbBRCxgFo
4Y/9Vk19zxbcqgVhCwrKyuxR7FNuPSsAgP41GwYKYROIC0TzzP0rDrSiC6CXhBQu
7ivjL8EanoaaeGqiTFeT5wEm01YNvbAv+NrHPAHcyy0xjGzGXLRj6LKiQBG/rfht
EgGXHUf4ahWL+yeWc0RGNNckHM9WjdS+1pRb8KZn493PtPLVAgMBAAGjgc0wgcow
UQYDVR0RBEowSIYWc2lwOmZsdWZmeUBleGFtcGxlLmNvbYYVaW06Zmx1ZmZ5QGV4
YW1wbGUuY29thhdwcmVzOmZsdWZmeUBleGFtcGxlLmNvbTAJBgNVHRMEAjAAMB0G
A1UdDgQWBBSFlwm401U3JIrc3uORcuQiz5iHUjAfBgNVHSMEGDAWgBSVRX5fK+pl
mBKRBPNjx2iaWBZ3JzALBgNVHQ8EBAMCBeAwHQYDVR0lBBYwFAYIKwYBBQUHAwQG
CCsGAQUFBwMUMA0GCSqGSIb3DQEBBQUAA4IBAQCoqY/YiguI7f9Pv+XNj557uOby
LKrjI1uacV79IKPd2dPB8ujwvnfbM8yKe0+RK43W9xTDjeBg0zRQvL5nIs31dHv0
mmiiUiuBL0bTCZ8lwyDoENXvOHvRF9Tx11RnVvETzy/8i4P8FOcBglmDzLGN8Mfa
TrHczFTPbDtHR1mH2Rbsr6/hEhMjHgrb9bX/XasVDuMlkQAOkNvYBxGQgQE6SKiq
nrBi0zbwDLcvpxeSUjYpFArWZYZnc3RuqjzuRzgeyG4GgYUcLvC2BH1sONuBnLgH
4we+9S8JaGMEa4cONrmho/vIMAygY41tqwr4RLB4GRo4fvpqodRLS3V1v28J
-----END CERTIFICATE-----
```

Fluffy’s private key for user certificate for example.com:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAoyxZDOm85OzTnvuZAuyxNjq30x1Nwzq2rlC9X1UId4x+pOnw
aDEojyMyVhnDIpenbf2nIioBta9hvV9+wRTlmCm0NE44iibuDdrbJ7l41qysBHgy
mMJ152q3LbM84+uXpe+LWUJQF3v+p4GvN6fn4x+wjdByL2wUQsYBaOGP/VZNfc8W
3KoFYQsKysrsUexTbj0rAID+NRsGCmETiAtE88z9Kw60ogugl4QULu4r4y/BGp6G
mnhqokxXk+cBJtNWDb2wL/jaxzwB3MstMYxsxly0Y+iyokARv634bRIBlx1H+GoV
i/snlnNERjTXJBzPVo3UvtaUW/CmZ+Pdz7Ty1QIDAQABAoIBAH+bSvjiQir1WnnW
YM78s4mpWeDr5chrvjmMQsyu/zQe11u4551T9FgcOl1DQGtpFjLaTz5Ug4nGYjVq
3QG6ieL5mkfddDH2R+zl3sWuMmYQG2ZTaZ41VWdo+V/v8Ap+T9YhA2UGiwQSoA/3
R0PLN3lTaws8nE+hwiaGGsweujBvcaIJu4RQrGHRHaeEplU+tfjcHHElfzUAmKyM
cMgF8IpdUcA1pyHe3Pyc0oGnLyEVnv291xGWQfWT7nqf7K0QDLA6+TvbG3fGEYIw
WK4DMraUbZ66Jlnj1XfADoxWOTsygV+KYhZcbwjBWAUSOSduAtfwa6b72OnWd28J
8KYvrXECgYEA1eCJZZSavxhlfxqsWC/WdQ8S3SimI62KSLrN3bI0RO/60KiU2ap3
16ZhNLq8t3DjpkWiZrukixs2odsU7k3z6q+qm++P0TUwL7z3Bri0FimqUeVSYgAf
ZmFgGz7wLAM29zhv0hTZjGrrwMlNSyJ2tjyqpiO1XqkbdBpPBxKPrdcCgYEAw09f
4M2QKQBFzjecPeQpwJqnh8cuoHS+2CNLYGjlmjd/zAUgVF2+WPA1R1DmjAqJ9iwh
15Yx3CbknpKbfhfilmHkcGyA+fjQaisq/NzN3Ya0FP9Waht0FoBsAHt9X5xFwXH6
YBKUrqoPF5DAy427EL1nsIRa+LtoPaTdqpphFzMCgYEAlgSOO0s2FA43uyTpeF3t
rmQpVilaB7KFSaiGGBgUY7p0koF9DwRsVT4l9sd48a7kb09ur2K08sHe2z8BenoB
Oj+HiyNJHHSTXRjNqNBLuTP2fMU+uPDfFX/92n6WFjkXB+d1P8VSJxUkUjCg36/H
1uHMzQZFBKXXVOPTROG3GDcCgYEAoPFmq8QZOIA+BbnzqVi8QzfuN8geFyE9JrSm
55JpKdT0HbZXts3tDjMbZGI5KUuB9nbViGb/PVBbcoSTV6vtD0kpyq7O9a5gaCyc
ZvS5PARFn0vt9NAcsHIxDZC1drU7EjaPQN3u4aPHff7NsK9haGD78gyPPoqIUsvp
0i0XNtsCgYEAxIUikI+5wXIrnC1FUt0gt6+4T0zc7qEO0EpQRtktZ/1saNXEhA6N
EUqWLJMOnClhp72V5IvXsKgjxU8VpgIZeHIIt5jZb8XMmBiSQxiVTf6rp3s8PqlM
EtXfh7TdJzKuRP7d0g2uG4boJMFf590nqNjrxj9VeSxEWUrSK3YG/h8=
-----END RSA PRIVATE KEY-----
```

Kumiko’s user certificate for example.net:

```
-----BEGIN CERTIFICATE-----
MIIEGTCCAwGgAwIBAgIJAJajhBdO74pOMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxN1oYDzIxMTEwMTE0MTkzMjE3WjBWMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MQ8wDQYDVQQDEwZrdW1pa28wggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDL5odVdA3gFf/MuGIqbMY8Kl7g7kUfexWkpXbT
ptxlxf2D8hzUX8/PUn2XXcTbP019DqA+MkMiX4NNGpDZyeoIrcquKUXK7UQlRoKy
Q6Va11DijHTqdPTWFIrRhbRUhPjj0WvG1AFPYRRG/IZfRQcH8Aw1w8XSp614m1mY
9XwL5LuHNimAgjADHMrSk1obmHws0thU9nV0t1UG1SA1lA32JZX81bqKDg3Tq1Ho
fsKU3GwoBZG507lVG5bcV2ByA5HnCFpFeDTDYE23197USLhqRtIqrxxr64SFo9Dn
P0mYH6e3lRveAZhdKIbCHgGaKqIr7+SZDnLdCyKDrFSPC/lbAgMBAAGjgc0wgcow
UQYDVR0RBEowSIYWc2lwOmt1bWlrb0BleGFtcGxlLm5ldIYVaW06a3VtaWtvQGV4
YW1wbGUubmV0hhdwcmVzOmt1bWlrb0BleGFtcGxlLm5ldDAJBgNVHRMEAjAAMB0G
A1UdDgQWBBQ02bNX/rnbbYoEy6wU7oyst63WbDAfBgNVHSMEGDAWgBSVRX5fK+pl
mBKRBPNjx2iaWBZ3JzALBgNVHQ8EBAMCBeAwHQYDVR0lBBYwFAYIKwYBBQUHAwQG
CCsGAQUFBwMUMA0GCSqGSIb3DQEBBQUAA4IBAQCTN2SNTLUcvgtVnBi3RBRtD0+p
aiFPtWQ+YWbyCG/+NetesegCwi7xBOgSK+GxUWpTVuDW5smyTTZyvrMQhpkckcyO
KvuUVz0/yK67oSume1vo75KY8BvgfeZXZG4PjqqelJ3czB0XLfeb6KFmtoiHQ/R7
4i/O9+MhB3Zoeg5bm5f2g9ljYwRbD1Uav/aH9WeGEX992d9XJ/bpGGPrAdgmV3jo
KDFKh8ys1yfmM3xVdU0qPtos2nlzGNaqoceeFZoYaMf8uTzoaan6KZkQDTiMDRpt
YKxyS721re/840FwDvt67w+GIfFf7ISrAlkHwroYt0NMnLv610rka8qnVvaQ
-----END CERTIFICATE-----
```

Kumiko’s private key for user certificate for example.net:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAy+aHVXQN4BX/zLhiKmzGPCpe4O5FH3sVpKV206bcZcX9g/Ic
1F/Pz1J9l13E2z9NfQ6gPjJDIl+DTRqQ2cnqCK3KrilFyu1EJUaCskOlWtdQ4ox0
6nT01hSK0YW0VIT449FrxtQBT2EURvyGX0UHB/AMNcPF0qeteJtZmPV8C+S7hzYp
gIIwAxzK0pNaG5h8LNLYVPZ1dLdVBtUgNZQN9iWV/NW6ig4N06tR6H7ClNxsKAWR
udO5VRuW3FdgcgOR5whaRXg0w2BNt9fe1Ei4akbSKq8ca+uEhaPQ5z9JmB+nt5Ub
3gGYXSiGwh4BmiqiK+/kmQ5y3Qsig6xUjwv5WwIDAQABAoIBAHCXmrGgRS0xWLBW
PLbKm+iLSRsR14+bqwbg663SHTAB1Yzvu+W2Bo2oMnvMJrEe0o407l2J6bJoZZvF
CKmKqrYiKaJkXgrBW/jtZ6xCWGPCNAL1pnX1IWG5tDIgj8SALOO4N7hyR0rrA4Rz
W0vuVQSYFFX4BhvdxZesyRwCqn3x0pPSff95Ad+vuJd5CYuFZCuyGkszQ3fi+Nia
Gqs01EuyolEv72rsw2E5+wtx3qXB8Z4HXr+Yq9NbE8lp2CWd1Uh1qIHl8kwWmnIG
V3oLKiIowV+M6Zx/uzwAMF0Rdn5kET+b5DOlIksUAAa8LZsf95rOvkLgw7aZaj5e
sXhAdGECgYEA8930YqU2+AcEkjC5hygw1M/X5k/IcvZp0a8/in2hJW7iZgGh0AFE
jjxuoIVXbxSf9cZ+M6g76Svww9ecmovLArqbhFaLfbZCsrLeEAhQtGcu3wv7o6px
N0EbbF5FmOK7qaQ1Sgqj0NF5zP2JsrxGNoRmgFFwVdcpP/3Jp/IlZEsCgYEA1guI
/7I8h9og1dmTPzMpvpnANdRF/iuMX9AE4LNRp09Hjx0B7Vuat1ABtx09/ZN1hLhZ
BTZ5R2R2RjbzSHXZ3FdoMgSx9Q3qa+xuPel4RcppHNjdYkPDhPLnOUwQBqFL6kyU
nTEF+k6VIZvNsmGbB6wpHU1cjDAZUx71p6W49TECgYAMHpa7pExUDT076rH9tpCe
sume544lsHtX0WbOAipVCuqzeRdKmBWJIBW7YoUS3yqH82JoPM8lamqfwQJmZ9Yh
/5YlAIwUJk+wQ9VnZJJmNM6OhTDvVFQmE9VCEHlS/Mmox6FiWZ8EjLSJ7HvAZzzy
Dqhtbh6wFW5WYM15zD3xewKBgQCRmIkY/QGFm0+Ih5ZMgB3eI7GGLB1sNe0nY1Ve
Dzv0pc3UQHQGI7CLDuYLy91V9o8St17+V76JXIHDYy97U4bdBau/kkgGm++gd9PJ
U11Xg8aaM73rUJLXhW7ZH68rA16jQnI4tpcNW5S/pr51n0UYI/hXkT7psPIZA08w
OV8lkQKBgQDaGzCYC/6WumGJUerVCzZd/H6+E3ntZmtz273c8+wV89oRtZzUoJY4
bVNrYFs9iKFxLtNGRECEU2VzDXHUAguqe05rbzPudAZ4wSsrNchUyw8lkIXHDckt
pVLs0vhRK2gW/W2I+p2exSPQPt3Uy8tT6IsB9ZbNg/H4D160heHkuQ==
-----END RSA PRIVATE KEY-----
```

Domain certificate for example.com:

```
-----BEGIN CERTIFICATE-----
MIID9DCCAtygAwIBAgIJAJajhBdO74pPMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxN1oYDzIxMTEwMTE0MTkzMjE3WjBbMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MRQwEgYDVQQDEwtleGFtcGxlLmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAN10BgIQwucEH7yMtiTnm5SjSDeFnm2D
EoRQGo5IsfqGjKeAub5S7KbKY0eErfZ0hYIWfk42QAp0LCCpag5qfzXPcHFjfelD
Z4FM6rUet0yjNQh7IQ0qcwdjnY11vx/UjuZnYHX36gp6bJCvkkXgYgWaihCY3HxU
i+RhlTsE/BBQ74BFul6El3bBICXBkh2JCvdVYmT66GmiYkxn0wjZYbU9FlS2t0SN
WSuQ1On7x32HWMMSrDN4AFC6BwWzuQEaY1Vs4XrsoweuOwKDoWngw9wtYemy47Nx
yKbP2vs+mcflcbnJF9TtvKBHVAmMbm1TmizJaMZv8T2RGiRDd32RaUsCAwEAAaOB
ozCBoDAnBgNVHREEIDAeggtleGFtcGxlLmNvbYYPc2lwOmV4YW1wbGUuY29tMAkG
A1UdEwQCMAAwHQYDVR0OBBYEFMwGWVuLXtYN8gVNG2hUHvz5QxkXMB8GA1UdIwQY
MBaAFJVFfl8r6mWYEpEE82PHaJpYFncnMAsGA1UdDwQEAwIF4DAdBgNVHSUEFjAU
BggrBgEFBQcDAQYIKwYBBQUHAxQwDQYJKoZIhvcNAQEFBQADggEBAGqa0dsAS5CG
sFPqbzAxiR6bCRS9b7kCqm9Y7jADuKH9s0Fy/7MNy3anF8ZXOAYT5fPkMBdN95e1
83Tpgfj0VaMN9YI4w5hDUh+EzRq0o0WfPeIx/cuire1gffrSqkkvQamAAbvttnXJ
l2l/DJFg8cRaNuhcrOGo55pV5eDNAfTek/Q4bMFx0v3NG10l65B7MUHnNw7lwAFI
kfc03cYfdOY0NObNkw8/zpStkdnicrGfHdOlfV7ipFbFsXFNEApdplbmVx9IpVx1
Z+qrNT72tvrB84rBgHEyGGwztfoWWhbhoWwZZ/VFaGRvsjHc41oastSHiZb9h7o4
TgoZBwNLm7E=
-----END CERTIFICATE-----
```

Private key for domain certificate for example.com:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEA3XQGAhDC5wQfvIy2JOeblKNIN4WebYMShFAajkix+oaMp4C5
vlLspspjR4St9nSFghZ+TjZACnQsIKlqDmp/Nc9wcWN96UNngUzqtR63TKM1CHsh
DSpzB2OdjXW/H9SO5mdgdffqCnpskK+SReBiBZqKEJjcfFSL5GGVOwT8EFDvgEW6
XoSXdsEgJcGSHYkK91ViZProaaJiTGfTCNlhtT0WVLa3RI1ZK5DU6fvHfYdYwxKs
M3gAULoHBbO5ARpjVWzheuyjB647AoOhaeDD3C1h6bLjs3HIps/a+z6Zx+VxuckX
1O28oEdUCYxubVOaLMloxm/xPZEaJEN3fZFpSwIDAQABAoIBAB9s231ni4Dk4OwM
u7w48acCFLlsSLMZqoMEKwCN6FO4zDTo23LaqaJxje0UMuuKVXfEYWAP6r6RBcIM
yHQLQMoOCdLNX4y+d+2tUJErLq+9aUUu093ebDxcMntkfh6yNyUS/mk/KQMbpFRT
1dn8oWxSJc19I6yxArkB7/9UEcDut6vzdbz+agXpHZH4Tje5OWZQXkHzsYobM8Y8
c2XwudP1zdQtvOrrOeirexxpOQf4CBQnBxoGmbae9Wf27Kw2bBm5+blZFgdqNxoh
6Q3rJ9EDyWkrVMAq9a67a59wST1ymyC0c6FmfToCMGlgoMPHcEdvuNYPWd2322oK
ZdfsawECgYEA+AewMiTdhAE+9TId2qilLQV+y8bdTHQ9rSqW9SF+q5ShOpZa79ER
asuDuqxU+TiewS0ircrkIyzQmCc1fnfBJh5y6GukpUk8HdLLkA29fV3ZJe+Y4ZbL
b4TEy/RxEECQREgtnQiaw08yOlT1dobNwxzVsi3mrhtOpfbPBERZUSsCgYEA5JG2
aGRCkyzASGAnZmqqXCP/pImU+tJb2OCgQ6/3gsxi/l91LwtRhFgx/ptYCgZWlpbz
+mpnDqexKtowldbjorrUADw84zG4u9d+uWOCXEpCVIEu4DZsRURdy3OzpK1vJaUm
NLgBiDj8JkUFrXTi4Rzx1Xysf6ndWAxDPDdI+GECgYEAoyFrYY+dohSvs9UijY4e
FV5n5t8E7iQF7L72SoOdLHy1DjOV2+VF71erbDusJ751q9hj1qp7Iid3ips/M87P
2qJsMTGbOJrST0s1V6mx16LCD5Fmm/jyFIbeaMZ9FpNgT4ipd38RSyPrhTIbv7kp
3Ao7AtXtwtVzBPUvcz8A/8ECgYEAw2ps2F13qdql3nsO1Ho3gqVoaGUUUUlOK2MI
wjYM1/AkZrR4PKthm1PIEpT/tTpsBz2yBBO6XoYya5+10DWz0yoGHNljeR7GgRqh
hqC0EHGQuizkRd9hu+rSgiI+oXmCQF4tBv+Wl7+YnKOAUidP3gTgIZUA6fjxe9io
FzBxG6ECgYEAyAHvSeqqwmdotdpWgR3Fk1CmtH7ZPnF2rsuRBaBoYnWtU619ote+
+Bmd4fBUB9tQOzUC9desRtoK3+wlJKHEPjm/0FxtQQi9ogHEn4e6P9jOwXJNkSsa
GjGUfzQ3Vm2baeNMg7sH8C5mQ9nskDuCzdlVAB2bMp23oPl6cvPIb0E=
-----END RSA PRIVATE KEY-----
```

Domain certificate for example.net:

```
-----BEGIN CERTIFICATE-----
MIID9DCCAtygAwIBAgIJAJajhBdO74pQMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxOFoYDzIxMTEwMTE0MTkzMjE4WjBbMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MRQwEgYDVQQDEwtleGFtcGxlLm5ldDCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAOwsdgPVSPMweLWsBDHUSXJS6Vk6pu6K
sVg8IWMf1g0TWTPc5jUAQlWlLNtmN4gcSzq5z1ecvf3rLMomJPZaWbektTTg1KZl
2wQgyP+vx/Hf1BByj3s2DE/KZoLnQjFQawHHMc+kCtSa6dCFTmD9nA5cYDVxNmKG
Kz/+5HYxe6ByI6NZGNlSB8ADPULcFg6UchO06JvrGFt1n9tAtMf5C31+YYGpqXBl
qZOV8Wo0Gp6Vlnd4LrvDZkwjpQ/o7EuFbiK34Gvh3cuh9EkMbk+IPgVv7ohjWPDl
6WygTkE2VXHDhhdN4MXPKyenXX35sB52fNytN+2qM8bo4QPfTZlGrx0CAwEAAaOB
ozCBoDAnBgNVHREEIDAeggtleGFtcGxlLm5ldIYPc2lwOmV4YW1wbGUubmV0MAkG
A1UdEwQCMAAwHQYDVR0OBBYEFNiNYjKOu6f046JHy28GDRVMeR7sMB8GA1UdIwQY
MBaAFJVFfl8r6mWYEpEE82PHaJpYFncnMAsGA1UdDwQEAwIF4DAdBgNVHSUEFjAU
BggrBgEFBQcDAQYIKwYBBQUHAxQwDQYJKoZIhvcNAQEFBQADggEBAHUzR2H2IWrQ
ls3iqNlG7815mOjm9mgQX6WP2ILwBOTOqtPJ9uE2XZU9qw6d9vdcbAgLpp4Em4T7
Whcs0zVTrgKpWjDlho/boRS1gP2Qu9I86zJzf2R3mhTHUsbpxIwMCcHQg/fdIIeP
5Ar8R5DZXx/Q9zdQLE+cjMSjxo7q7uOV8DRkgMpYtp7BURg5ZXhnkAhEHxa3/SbU
YGfy3PzRoAMQmRZieAXArsIxEfkaC4Dtox/D4XLvY7njBFv8H6wqlvQyDsKXWlUH
8dS9i/3wFEpQtymUUeXwk8gzf2ytT6hgrX70s6BLy/IeRU+wLJ3k5YZpopQZjDm1
fNQG/O8TJlQ=
-----END CERTIFICATE-----
```

Private key for domain certificate for example.net:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEA7Cx2A9VI8zB4tawEMdRJclLpWTqm7oqxWDwhYx/WDRNZM9zm
NQBCVaUs22Y3iBxLOrnPV5y9/essyiYk9lpZt6S1NODUpmXbBCDI/6/H8d/UEHKP
ezYMT8pmgudCMVBrAccxz6QK1Jrp0IVOYP2cDlxgNXE2YoYrP/7kdjF7oHIjo1kY
2VIHwAM9QtwWDpRyE7Tom+sYW3Wf20C0x/kLfX5hgampcGWpk5XxajQanpWWd3gu
u8NmTCOlD+jsS4VuIrfga+Hdy6H0SQxuT4g+BW/uiGNY8OXpbKBOQTZVccOGF03g
xc8rJ6ddffmwHnZ83K037aozxujhA99NmUavHQIDAQABAoIBABfBYR2BlpTfi0S6
yLE6aSjWriILhD76NFxrr/AIg79M8uwEjCNIo2N5+ckXvv4x2l9N0U0+tt2Tii3L
KGyfKecO6isncjxKgn0nzw/o3nO1z97Xpxb9mL9t3GHOYRoUvK6xGpGILo60BlCz
F+8pk0jegc7eVFoUpMULHm/FCmpY30N5cvCHcAE/ncW49bZmH3gQ+cmr5UcKKDUY
baJyLd8Q1f+uSmtrfYZzRT5c+4wmrBUjv3w9poMJuEo4slRaDnyeKJPSNR/6/LJk
tqnqgNif9cj9wqF6hWA23dDmmU/kSRtn1KOz5XmV9Jbo4Fu64Fvn/m/hj5Og4CP9
hZUWIQECgYEA+nV2pzspCfS7jSebVnvjChvqJ0nJAilSqCmrSQIT5PRmO+GQs6UT
PVN4GE0Ms8TTJyvxVkpoagQ36VLw/Wr0jUm+Z+dv1TIlFWTas8RNmdZHMv0LvfEe
Qu2fTI68l2d/L9GBMUCYa/sucX5E9q+3LC+Qo9jw8ehWjQZsWYER4dsCgYEA8WYX
AqDdKjHRqu2h248gZsuogiZq05iuzXhk2VTQoiM92mu8m1Htak+eov3/3wojqxuw
TAQbf/t8EfQ7LIGjaKqAua7mgG/aNB6MGGwdpBAPUZDL+DuKfbDbzTOL/IuaW0Fp
40RC0Up5nTU9wzIKB7a6n5S5R0KXxiGUIphfcGcCgYA6IYdPmziUOfxJ79ZrBUgV
8ZKwWbzQxpyLsVgzEsthSaRs45a9S2QiyLvIECIRm25S2i0ilRSU/rOncPvEJc3q
+SG7Zgkb146p34WvUbGdMhHGcNsH0+3tJM/jagG1tmzbwWmV7+MwtNT7vI3vH6uJ
EuUkUlbiHsXv53zAbWekHwKBgBy5HwfLCEXbA62o9NdhImPY28YQuClRQ4tjReyu
MNz6AIQayahZiTxbGO8f9fAeDrxvYPzKiFMkI1EnlFrpWf48O3DcpMSninklIVpO
kwBQgOIdrods3j+yaZTzCzcTjVxKXkUSfDjW+b2A9kZhj9v3HCGc2qbl/5Utraio
JMMFAoGAHb+k+C4e8WrW+jXbbG/DgAkSokK5vZwZLHeWBig9bEi626xN/oFEQVXp
zqwyNo6zQaofmS6anT6P2M7NClSGJxh27eBTiTLp1NCXlGTWAQEtXmYtvnAZNzXC
5Ur0wvS5bLx0nbhJwN8ZBwzJhYup0kU3pn99GcF+vkj5Eg7Zftg=
-----END RSA PRIVATE KEY-----
```

B.2. Certificates NOT Using EKU

These certificates do not make use of the EKU specification described in [RFC5924]. Most existing certificates fall in this category.

Fluffy's user certificate for example.com:

```
-----BEGIN CERTIFICATE-----
MIID+jCCAuKgAwIBAgIJAJajhBdO74pRMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxOFoYDzIxMTEwMTE0MTkzMjE4WjBWMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MQ8wDQYDVQQDEwZmbHVmZnkwggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQC6VyOIP6UANXy766KHiYDxyOpYEFboLJv6SEtw
UWQoZS3hQurFidOu4gkCspblzaMoty7lnUexbFxUKdbJOWGMcB2hrezJ+6rwJPK/
bF5YDijVtVqMRd5lv/Ni5yzteHfrMszWnz3t+ojgak4XTjBJmP2RO0T67GUpEbFV
sDeYtWi+G1ebDAR6bf6Jdba2K6DnmkxT5Rr6oYJHIApYbubk28asBQN6EGBBgPEO
RReJYrjoJR/rBDDe1bxK+ONdFXPlwjI/TRPMpvUYraWgTjJ18tXISgF1htaa/Y1K
YP79Yun2Nl/3UQcPIc/C6CXBs3yAUK3qQO1G6C5pXH9KMMlNAgMBAAGjga4wgasw
UQYDVR0RBEowSIYWc2lwOmZsdWZmeUBleGFtcGxlLmNvbYYVaW06Zmx1ZmZ5QGV4
YW1wbGUuY29thhdwcmVzOmZsdWZmeUBleGFtcGxlLmNvbTAJBgNVHRMEAjAAMB0G
A1UdDgQWBBT7CTXlQ5GKWvxGZNY24mmmVuEnRDAfBgNVHSMEGDAWgBSVRX5fK+pl
mBKRBPNjx2iaWBZ3JzALBgNVHQ8EBAMCBeAwDQYJKoZIhvcNAQEFBQADggEBAKL9
wUWGRhCQdhjzY4bx0R5Kwz+NHvsb8rjlPqfdcbNujBCw+rD+/uux0G3HwW+Mraj5
U2tUehwz87k6SgdqADzL/CP2mjzCJo5uDhi+tzjeg6ZklTSZYQrL3FSv/AgcUfFI
9HuCGkix/htaoEMy2zNZnZOjdtFME9w7wb3GxxqWTUzl9TToloCXYmLeQo/jwuad
40ybun1P5CWkO5Md2Y5zuNfCsRRz5lLYtAVfANtLBfeFV+S87AwrrdeITT+iyB7H
Jj+t24U4IMC8MttcHBlPPBuRVc2kmhNEQuTzelCsldXgY2+kn8ItnLdv1mvLpXA2
2Y41CPLCSj9AlqqZL9I=
-----END CERTIFICATE-----
```

Fluffy’s private key for user certificate for example.com:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAulcjiD+lADV8u+uih4mA8cjqWBBW6Cyb+khLcFFkKGUt4ULq
xYnTruIJArKW5c2jKLcu5Z1HsWxcVCnWyTlhjHAdoa3syfuq8CTyv2xeWA4o1bVa
jEXeZb/zYucs7Xh36zLM1p897fqI4GpOF04wSZj9kTtE+uxlKRGxVbA3mLVovhtX
mwwEem3+iXW2tiug55pMU+Ua+qGCRyAKWG7m5NvGrAUDehBgQYDxDkUXiWK46CUf
6wQw3tW8SvjjXRVz5cIyP00TzKb1GK2loE4ydfLVyEoBdYbWmv2NSmD+/WLp9jZf
91EHDyHPwuglwbN8gFCt6kDtRuguaVx/SjDJTQIDAQABAoIBABtIBLi+8K5eJlvw
/MOxOwKrMrwf8ElftppnGTxhfjN31MbFIFA5hJd3GnCdqwAMIlYks6YEZ+mu/rmH
wp2FXCXOiFgSebd8tCMilbO27v0fXZUkTxR4aj4lY0HYrLg7yfrSXjER8WQ1KPMK
PVKmLOWpk34+2jOOhqUDpR3xhcJClQ81fC1hKe2JoixNDoPdfM3azTq8QUPLQD2I
mjww1IH1677G5o/6qMloOM0Feqv/3cUWiRmvPv4eyGHdNtuFXKFpB4DQQMQL7TD8
FoOHBymHIOzSSF+gYgBFOb0YNgu2CqZrfED9cf0rRotrbXf6tM+akclxfHhkfKaa
JPZosbUCgYEA4MaetKsa7azhEYMc4TK0xhhV5Hi6lj1xR/6h++uYF0OIOBjM9yU3
5n6vLpyghNbW2bK08OIWPO0F4syvyKYR2elmUDraH29DKAtRLEkU9K82RG4AmXmk
G6ZsWOfx6Jf35OnAKVj/7aN9jc4K1v6EFyQGYEXbp4I0fhFfbJBAe28CgYEA1Dmx
iKJD+jWW9ypHk51YJ3r+a5qPPNVmjGKQQje3Y6+rSlxmW0hMwXoCBOYRwhHBRA//
SxH93PZ8rECjNkhxp6Ao87X2Gcol5U6kH+rwfd/3+SsHqPrugaDIwNlgkcu8VRrP
8uP2CgJoDBi5UY2UR97GVK98x8k2Sf6kDT32mQMCgYB/KH3R8VY7jOiKcqTc1UWl
J1E3/gB4S+wQ8YELth0FVCP0sDsLuZdlItfRw7OfUraa01k/SHeSIfiJdIghN6mz
oDFMQ+7vh47zUWurZPCg95n4nk5ihIkNR1nV9elJTudjLcWS3pFyC2JU3XIObE+n
k66zufFoUuWFSCi2juibqwKBgCT6RHe1JjkDe2FniX8r7D88y/W9wXVtDWgqiE4x
XQ/OfP8A6IjBKTaQ5qcp2zBAXbdZPjc7VEta21A8FvQPXVZCrsAAFXha4413zVsO
WYblLlTI7ZXA2yvU8wW/Gnds00zU1iTRGX6W+sAY0rll/M8k/tOknA5HfeEYsEbq
Y/w3AoGASjoC9Fjy2aBvH8SQaimn/Rx3hOFR4myOGWtHxrXmezoO2YdcMO1d8rlz
A/sQRvVofHRwyoaIkZkALprEGyxEqCdMmEs1h9xYAcxfW23RfqC39DYb9RTrRkwa
ArJmcEdRESOsIYhhXGfE1QMGiwj1UXMWeYcLtqQKWiLLDTYYfQE=
-----END RSA PRIVATE KEY-----
```

Kumiko’s user certificate for example.net:

```
-----BEGIN CERTIFICATE-----
MIID+jCCAuKgAwIBAgIJAJajhBdO74pSMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxOFoYDzIxMTEwMTE0MTkzMjE4WjBWMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MQ8wDQYDVQQDEwZrdW1pa28wggEiMA0GCSqGSIb3
DQEBAQUAA4IBDwAwggEKAoIBAQDE/QVN7nxDDu5ov6b0cmHIFH93KhNbTEyCisir
i4OeUBiCv9dgRgPBXffrIIVQdIlCoDeLDusHdsC9EfFWvg+pRlKVEDgwccO0F5AV
bq3MK2Njma5I0lwpIa0RXYQ0K//oX/+jZeakhFty/R9yer0KaXWdLRd6KtncISui
z9rFhlTB9lHg6vNJUN9+Xonbcs7siXbj3qZdhb7oipI4PoQlXVetyu+SzAVe6MsU
5lwLmpQpIzQdSsJyxaAsW+AsyxunhWWiPZ888UM4vXjacZuj8GvJ8w2XjgJilQvV
s8ojWMKnAGLaR7grTBmGQ90e6+cg7hWuoGBlQA0R0h8zWQz5AgMBAAGjga4wgasw
UQYDVR0RBEowSIYWc2lwOmt1bWlrb0BleGFtcGxlLm5ldIYVaW06a3VtaWtvQGV4
YW1wbGUubmV0hhdwcmVzOmt1bWlrb0BleGFtcGxlLm5ldDAJBgNVHRMEAjAAMB0G
A1UdDgQWBBR6WwH61Ul7BIWeiKM35fMAiE9xazAfBgNVHSMEGDAWgBSVRX5fK+pl
mBKRBPNjx2iaWBZ3JzALBgNVHQ8EBAMCBeAwDQYJKoZIhvcNAQEFBQADggEBAKE8
y9YyoZlkFw4WxPalK087sSEveKBfzh4TuYQf5YcSIPw0coZGj/gNxn1juiYhE93G
F+Si/hJM0M6cc7SLB5Spq06Tt3PyPBIOZOWk9koh92kDI3axSr6II9Plsvp+Xsrl
bz5Zy8njy/YZrk/qOaHqQ5J6nPNp5qwF+ns2t+5Zl88Lli5nkBgOXFOuE0RIkcdF
CUFRUj026GxAILR6wUThOzfq55Azwl5Y9Y9QmEjFhkbYLls00HxcJdnt+6Sdm/vN
MeMJZdTzp1x+8pfPhJgHoyz7nkAxhgzC9RT33ra33BNkMQ6esRlQONJ+ZRsRLhHP
O7+kvXvmj9AAsA291wY=
-----END CERTIFICATE-----
```

Kumiko’s private key for user certificate for example.net:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAxP0FTe58Qw7uaL+m9HJhyBR/dyoTW0xMgorIq4uDnlAYgr/X
YEYDwV336yCFUHSJQqA3iw7rB3bAvRHxVr4PqUZSlRA4MHHDtBeQFW6tzCtjY5mu
SNJcKSGtEV2ENCv/6F//o2XmpIRbcv0fcnq9Cml1nS0XeirZ3CEros/axYZUwfZR
4OrzSVDffl6J23LO7Il2496mXYW+6IqSOD6EJV1XrcrvkswFXujLFOZcC5qUKSM0
HUrCcsWgLFvgLMsbp4Vloj2fPPFDOL142nGbo/BryfMNl44CYpUL1bPKI1jCpwBi
2ke4K0wZhkPdHuvnIO4VrqBgZUANEdIfM1kM+QIDAQABAoIBADuLR+kwp3sVrlcX
Z34IfSofmBALNeKpA4+KJ/JCr7xQ9bfACXhecZAnuWLnZ6TUNRFgoKl2DvEookYE
gHD57n36dcf9KR7rpH5xiOoRlJNcoiRfNeFpRNZiCZBwNiAXFLnHGtznVnpwT7xI
axMNqsrU6epi0O/quAPkOu5x6e0+j+j3ZauI4EfD1w2R6moBMUtATauZEEyLuC9A
6bFz2AFDchPVLwSjNMu0tAJc8Fss8xKls9HUXGS22eUfHxWfkCGwChuW60obGmas
E7GS7h4g9QvvQ4hGSVy9/MmQ88GmT0LynOyzFBCpuwjOQTHwsD674ldMSL4kXYVK
jcnTAkkCgYEA4bjN2ILis3uWTjvTNnrmWn1QoZBZDhg1LuNs5o1XtOJ7CdkckUvs
nqqQYOzNk/9N8vUs12ds3csXHypuuGrJwAVf648RSPDUUQ2XOoPSL9NeuZt5V1fT
1VyVWanKCBZ5sztISNVPt7Pu8DtGLHch4S/7M+gEUQB1Ogz7fyJHvFsCgYEA32mE
6lN67aHkqMLa06ZI9JIk/3SsFIPpjwZ4tk+sQCqEzawPvkT7qF2+U8lVt0XXKJZL
aexsopsULCGS86TEAPoYtjjk91p6ZZj8mgRZLU55g+gRdTpAFhXMgIctU7U6cDIw
SPa6UxJp9XCa/Gf6YLfas9VBhc/8OC7I4ygjLDsCgYEAgAG7yuM/CSY3MRrARw8f
f4W9qkIgHtwfnP2gjobtjEk8GXOkvcle4QQ9aJoiY6HPZM8hpO6kUIuSCzyXGcKF
s33Yzc+Or9zTqzuX3blQA4tNFtlS0POf0En28KhXSIrmbXxbG+LMmJNUF6yluSW+
cuQxA1i6ye0Gjes63Phl0i0CgYEAuEcILGQpTGMyAYWgC93n5Vu6ir+IxO89sgyL
ew1irhakLiWTYsTxsyGHwQKb4i0IWOEHWVp7DPDPhcs3tCIezhN8WKm7KtAFj1HO
YZfemsFU99lutPwUKmNWqFlXqOkeR7cOHtDsRWM15Q45uKJnYmmkSptHjYFNsGXe
q4fK40sCgYBoAYtsLfMlqt7s3htx4hZSMFbLP/iMGW2DMMAzDW+Xxsvw86ibrcWY
8c3hbohuJBpyAzba4QoR2G+gtRmodLca+tQFMrObETHFglNCY+WoHRSNRImbCS8w
dsszPgHWf1nrxBLBiDFlHZwSqbZtLyBjPlHJ+fTiPNo6UTx8aDQ4Pw==
-----END RSA PRIVATE KEY-----
```

Domain certificate for example.com:

```
-----BEGIN CERTIFICATE-----
MIID1TCCAr2gAwIBAgIJAJajhBdO74pTMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxOVoYDzIxMTEwMTE0MTkzMjE5WjBbMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MRQwEgYDVQQDEwtleGFtcGxlLmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAKEVuYyZlaqfqks9u9yWQRp9WfI+VsQg
GpJH3vAfastElCdxlBV7+R2CaQ/GnXDnE0lAC5SiKRcvPHq5OLx1VnDADMWmcXBv
wK5n1zN+7MUCy/MISMr7E2Nd+py8Ft3XhjWDIuUljAh4HDO4fxS/BFy8zozADxvP
OfpE40EABF5aj7e+xjtkErdkMybAcSYyo53IHP3wDPxmMzCsOw/fi8bfy9j1GiUD
uz01F9qT/Opz9K1snxgT1IK6GRlktG4JawSiohW1QbARfj9//hR7ZgeB0gO6LLGX
cGXdl87JdA4ZHMZNinN4Cv8ctZYSQZ3dbt1pRRbGtq7elPskiinDuUkCAwEAAaOB
hDCBgTAnBgNVHREEIDAeggtleGFtcGxlLmNvbYYPc2lwOmV4YW1wbGUuY29tMAkG
A1UdEwQCMAAwHQYDVR0OBBYEFFNu6jHPsItA+vy/Jqv81MW7wLJpMB8GA1UdIwQY
MBaAFJVFfl8r6mWYEpEE82PHaJpYFncnMAsGA1UdDwQEAwIF4DANBgkqhkiG9w0B
AQUFAAOCAQEANH+wX56VJd0vVB9+Mef1xItWrSQUyNYZZCBq+y/5vIoOp6Chaupn
xjTjWf50zg6CK8yKBWq8pGlG45GTUx+uCx+nVIbHpyTT5+YDDUzlIhhAUzIOOB33
Fd/XI/1PK5p5ftuJIYXU0rGuaoH8ud/p2nhIf9mwicUHxViTX3PUwlFC7eMbevBo
8/dMYnHb2i40ug6hsiYggsmQDbhHLVLo/yqkpvgzPLSSlkXS4sv2oIoJ/ISuSjhP
QkQ7mh7h01ct/LOa53qWfbCVogQDhMEqPTVdPm+JzTrMlWeZdrk4KbnXGp64Jtpu
xTVI4GcVAGWUT0cmpspDmHbPOKm5kcltkg==
-----END CERTIFICATE-----
```

Private key for domain certificate for example.com:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAoRW5jJmVqp+qSz273JZBGn1Z8j5WxCAakkfe8B9qy0SUJ3GU
FXv5HYJpD8adcOcTSUALlKIpFy88erk4vHVWcMAMxaZxcG/ArmfXM37sxQLL8whI
yvsTY136nLwW3deGNYMi5SWMCHgcM7h/FL8EXLzOjMAPG885+kTjQQAEXlqPt77G
O2QSt2QzJsBxJjKjncgc/fAM/GYzMKw7D9+Lxt/L2PUaJQO7PTUX2pP86nP0rWyf
GBPUgroZGWS0bglrBKKiFbVBsBF+P3/+FHtmB4HSA7ossZdwZd2Xzsl0Dhkcxk2K
c3gK/xy1lhJBnd1u3WlFFsa2rt6U+ySKKcO5SQIDAQABAoIBABI9gIZAOedZLxJY
Cja/ON4EBbRdhLuumvOnecIc/J3JxTD2Nnt8T0gdJUJpDhjjwZZQzz7kYdzDN4j6
Akeszb30sT2MTFob/WiCT6cAH1VrrKZ3cK6zYY2l7aPj1H8IUaUrlT73UnT/DMp6
gMFbo+XQZ18evFc8zubc+BK7KsN4Nb6/zMhw+PXEiyg2EGDN1Fo4TMhxPD4wBIMU
8oLlE8A6GKimxAk3gMuIiS6Ruau2HpGkjkkHkAx/yzU1s8BCMoLDJjyyH19PRISr
n0VFfe0gM0aZpdZ/94ynFPdMnBXTq8BabT09eiycuLKlL0g/ERmj6jIImGSYRWED
GzlzX0UCgYEA0FDUek2uLhyltXwlzhDTldyuItiYZq/MeXaq2eA96zhJlD6aX+55
PQIxEEfhgTNf4e4cKjXQSD7aixy7jp/kFGowFRlB4pwbLDuhlniYSxa8Kv0OpJM4
DTAGue4QFZId5Z43KH755Ub7tjrCEIdQnij44DA3gPnjqXk973pdyVcCgYEAxfUx
/zMXgTp7HxW+QHZD7xXEs4Fp1xjzL5BaHoJnM7WbmkWvUvcMaEE/i9RqpyGlXRiN
jX6KBZ9UVgh/B0/AcYMa3DImTa0+Uie9kN7jTi5pzvIUAdFh+RyQ4tULWr5cgrzv
PjGG9tXMthuIbILSumVEwvC+P6Ksi1r4xp1ezl8CgYEArF51sk2clqM1qpnzXjMm
IJbdsA+w6ycD9m1uqaGXGo8UswmqCz70KrspheM0gQfVisjPnU2x7lWz1/AKcdVz
kEDdUFf54FxzT4J4Dl3zBg7l3FxQRXVbp+3ZYvfNb0vcWSc1VNjcRg8aMIsmES8m
UfhtFnRPOPWMn6qmyQVjnTkCgYB/3zlinkBKq9ooZEU3Iq4TXL5pLemOloFQcjCk
kJvVnTRcXTM5pngPSEaiLp6OQ3+sOVYG1nyV0SwLPwW/VVb8fDH3lzWC66vcKeuc
Dz5JnFWg5mLiIbzly/wTaochIOJlWWI5jIigHc9Uu0hOv9sbqJrYSea6+Hv4sNUO
h01chQKBgQCKLEH7vWQX8fkw+yKnmvAFoZ5H3IHUQw/WYsoCOVnWoY+vowcuuTTt
cbW1VkrtEjJPuYeEPa5NI2kmsNUZGrKCpx/3uq2JfMVopJzJN9biFM4ulcKqf9ie
hiVIFVVmxq+dVmXBgXCknhYK1Mnt9b3BK6mDqerQjK1TKryqAJ2QpQ==
-----END RSA PRIVATE KEY-----
```

Domain certificate for example.net:

```
-----BEGIN CERTIFICATE-----
MIID1TCCAr2gAwIBAgIJAJajhBdO74pUMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
QXV0aG9yaXR5MCAXDTExMDIwNzE5MzIxOVoYDzIxMTEwMTE0MTkzMjE5WjBbMQsw
CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
c2UxDjAMBgNVBAoTBXNpcGl0MRQwEgYDVQQDEwtleGFtcGxlLm5ldDCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAKoWx8g1KbnGX2YEOXrbod2pbR0fpkYW
V7O/tIWHddl+ACLlqqNPKSmIqwAFbZ2uf7S95OkXhkgRJGw3BugftUJS7zDhqVqi
dgPLMUPrdzpFazeh/AwBjc0wNBz/6tkUXrm7y/FwwzaCoKw+8Qm4Ibn2E3bNqWlm
iyKOXnYt4LGmy6J5e64hfQ3Vqe0ze5cfLKcpBbjF/TF75utbnH25zE0C/o1b+x1f
dwyDjsH0NN+A1ZFrI2NdleVAuH6F2vx4ctwZUzUJXyXezFmw5SRzhtWkb0iHO0ER
Ne7hCHLCv2Z6/GfIuHirCsGtNKSQIC6k74MyD7D75nltnLVgJ7Oxt28CAwEAAaOB
hDCBgTAnBgNVHREEIDAeggtleGFtcGxlLm5ldIYPc2lwOmV4YW1wbGUubmV0MAkG
A1UdEwQCMAAwHQYDVR0OBBYEFC1TKpLjuKa/dPumVbeFXEW4UR6EMB8GA1UdIwQY
MBaAFJVFfl8r6mWYEpEE82PHaJpYFncnMAsGA1UdDwQEAwIF4DANBgkqhkiG9w0B
AQUFAAOCAQEAJry8LukecUv4DUs5u/s6IymyqDLpeNvm94yrIIk/eRW72Jtr9rf5
6zF0Pd/+NzDXRYPe99HQgF3EKYndKIfnRUStJzIqiba2UszypDVRTQ6W9cH9e/1q
FdCjjeoVkRvnGo91S8DkgWM4boNRUgZtYwP+1I8hR+0717tp0f4fKjYX+NxPe30r
WzbLYXFDEiPndEgcxHc84Eeupit7VBQm7jxtF+XbaVGiLPGKCiYqdVS08h2ZakRK
8T3xL8Ecs4/rQn7PNPyEfS52R8hC70r66aAxZqLbKNpth/SZ3/hdeAyJ/NnFMW1J
uq3kB5YAJSwMYAUXaQhB1BvxKzXqstzJHQ==
-----END CERTIFICATE-----
```

Private key for domain certificate for example.net:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAqhbHyDUpucZfZgQ5etuh3altHR+mRhZXs7+0hYd12X4AIuWq
o08pKYirAAVtna5/tL3k6ReGSBEkbDcG6B+1QlLvMOGpWqJ2A8sxQ+t3OkVrN6H8
DAGNzTA0HP/q2RReubvL8XDDNoKgrD7xCbghufYTds2paWaLIo5edi3gsabLonl7
riF9DdWp7TN7lx8spykFuMX9MXvm61ucfbnMTQL+jVv7HV93DIOOwfQ034DVkWsj
Y12V5UC4foXa/Hhy3BlTNQlfJd7MWbDlJHOG1aRvSIc7QRE17uEIcsK/Znr8Z8i4
eKsKwa00pJAgLqTvgzIPsPvmeW2ctWAns7G3bwIDAQABAoIBAHIjpV+B5YVITL59
+UCr4JyKVLGlioQf/CygafjtZTVVa6v/aRn8Rkgb8XyrJ9sXvZVBlTqiUbdM4Z9I
8faVSKLAWsj3thkfSojTMzU77x+IdCG6LxSzekAGqAIJ7sRL+iEzl/Fm1WlgEYhl
GIWILgHH01n3O0eCy72dwmAV+2Hazn8eBggkWxMp0fblRC9pVh0FCo+jy1lHasjL
oOBkH51lbmZ4PUuUY072j2665gPm7i0nr25igef842JkbqAV8rAoNlQ26Y7tYLEw
6QyLv0odeb0rHZ8IEzahWAdmIPGCIUcFM7RmyInOatGA0dVEU3uYnkUQQVOi/JTx
46CCMbECgYEA4c1Dv/IVz9pdW1o/0MaJ94zfeg7Pgn5DRXnNMjCsSxVHSMINwlUl
BcYozs77vWbIuXiXO2xQe9mGA2ss3+vNxB0eu6EBQ/fK16cQQQH52nXdrV1sqnkN
5B5elFKcZKPfNVWrg0BC6csDndTcHp9STIKsxWkesLzC3Vz5UXZMsocCgYEAwNYV
+SsCIQGLT8ZZfKyE2nHqRUFknKc/tWQJop5gnE4ws3Lql3SNyCUQr/sDYelxQDE3
6COm197JcZ7jggDq7grigIxMznRxLMeG7bb7FfwPE/SKV0H5uagEB7ktFl8xIJKt
yOCK1ulillQjToSs4uetHLRXKCDSEpRiSw7wRdkCgYEAkDKBXYa/nykYDUqpDi57
1PbFkDD9G5x+YVPTUoX6wUgpabFjEANHzVQqo0dTRDTrYmY8Tdpx22WiS3SaB7WS
hfcCtVewczM++lDZ9GnKoVQ76IaM6qC72j36sEXBUhPEa072ZK8ZDCx1dsmEeJnN
+MZKhxcGXl9tIehJ31foyukCgYB9AUs1PwAeTVX13OrduyhUQ0xOoNmMA491Euh8
FpciPD2t1mzkyZWvjPeIXPwQWLglmMJZJeNeRPnpQcrR165zqXKzSj/wBePn12BM
cTXLRp6vnPKhJg+wno4eQ5hKzGKYbv1hHs5iCuDx+pD4sWExpmW+Gdn2FXCYwsAF
UCXJ4QKBgAKSrm8Y5xQhd8RAMg9JZLGUpPnmTKNU98f3fUFnX7jZEZETasnn18vd
65x04h58cohJJkNxqeL6k3lc3Mw0pzZrvsIha3ZMEoJPCgwBa8zLzrR13YQin6yf
+bAmfTDmhigpORB36ODY4B1kcwxKzQ0n3XAtlrL7NRV5wHr2ejkY
-----END RSA PRIVATE KEY-----
```

B.3. Certificate Chaining with a Non-Root CA

Following is a certificate for a non-root CA in example.net. The certificate was signed by the root CA shown in Section 2.1. As indicated in Sections 4.2.1.9 and 4.2.1.3 [RFC5280], "cA" is set in Basic Constraints, and "keyCertSign" is set in Key Usage. This identifies the certificate holder as a signing authority.

```
Version: 3 (0x2)
Serial Number:
    96:a3:84:17:4e:ef:8a:52
Signature Algorithm: sha1WithRSAEncryption
Issuer: C=US, ST=California, L=San Jose, O=sipit,
         OU=Sipit Test Certificate Authority
Validity
    Not Before: Feb  7 20:21:13 2011 GMT
    Not After : Jan 14 20:21:13 2111 GMT
Subject: C=US, ST=California, L=San Jose, O=sipit,
```

```
           OU=Test CA for example.net, CN=example.net
   Subject Public Key Info:
       Public Key Algorithm: rsaEncryption
       RSA Public Key: (2048 bit)
           Modulus (2048 bit):
               00:d4:46:65:51:f8:84:1c:b5:93:47:a5:15:14:06:
               ec:dc:2a:77:93:11:5e:75:14:d2:88:54:bd:16:50:
               dd:41:3f:7e:2a:e4:26:d5:a3:33:b0:5e:37:1d:e5:
               96:37:1c:1c:69:80:a4:ef:fd:22:78:d7:ce:d3:c3:
               de:96:fb:87:30:88:bc:06:14:80:5d:f3:ab:d7:64:
               3e:07:31:dc:97:c5:d6:19:26:bc:7d:0b:f8:de:5e:
               f9:0f:dc:9a:45:0f:28:8d:dd:fa:15:56:d5:35:17:
               28:80:d2:fc:1f:d6:95:95:42:0e:2c:47:38:53:ad:
               fd:0e:24:fd:a3:43:33:83:52:65:54:da:48:d8:dc:
               86:42:d5:26:ac:1d:52:54:08:52:e5:3f:4a:76:95:
               77:8d:c6:f2:33:f0:18:87:c8:fc:5b:54:5d:dd:65:
               f1:5c:f5:c8:f4:36:54:8a:b6:7b:6f:f8:55:f8:d8:
               d8:df:a9:7b:40:45:4c:92:0f:aa:b2:2c:a1:a8:64:
               d5:99:22:1e:28:78:a0:d8:e5:51:64:3f:03:14:a9:
               12:47:61:84:d6:b0:69:1a:6b:a3:6e:d8:ca:ce:43:
               50:ad:57:96:2b:87:15:d9:c2:11:03:b0:82:d4:f0:
               80:bf:dd:44:f4:f6:39:0a:2b:e3:4d:d3:f5:e7:aa:
               34:e5
           Exponent: 65537 (0x10001)
   X509v3 extensions:
       X509v3 Basic Constraints:
           CA:TRUE
       X509v3 Subject Key Identifier:
           72:70:CF:66:1E:23:A5:38:FC:6F:40:8F:86:8A:AF:E0:B9:6F:E9:C3
       X509v3 Authority Key Identifier:
           95:45:7E:5F:2B:EA:65:98:12:91:04:F3:63:C7:68:9A:58:16:77:27

       X509v3 Key Usage:
           Certificate Sign
       Signature Algorithm: sha1WithRSAEncryption
   70:73:c0:65:9c:2f:09:39:39:d6:a4:5b:95:e7:7b:43:34:b5:
   b9:b2:5d:76:eb:ef:87:e0:25:b6:68:ab:ee:f8:f7:85:c4:21:
   47:bb:6c:68:62:ff:f8:84:1e:44:5a:30:4e:ce:97:91:cc:3d:
   43:4a:8b:b7:25:26:08:63:c6:71:4a:c1:94:35:81:66:de:23:
   9d:e3:37:de:31:80:ed:58:b7:07:a7:ea:87:d3:cc:da:1b:62:
   c9:82:c2:17:e6:2d:20:e4:b2:69:14:cb:05:43:34:6f:b5:2c:
   60:d8:44:43:f9:e6:e9:3d:7c:54:a2:b9:d9:1e:7d:67:bb:3f:
   32:31:0d:c1:88:78:a8:67:39:f5:d2:3e:08:f7:38:84:a6:8f:
   c2:3e:00:ce:5f:b4:c8:da:a1:b5:2f:c2:89:60:a4:3a:2b:be:
   98:e0:44:34:af:ec:7f:73:26:f1:94:5b:39:09:b9:9f:93:c2:
   9d:7a:96:2f:82:66:c8:4d:f6:db:87:00:8e:bc:2a:b9:51:73:
   6c:cc:ff:e5:31:25:b1:4a:d0:9a:a9:c3:65:35:21:89:76:3d:
   39:f8:84:42:a6:03:0e:b5:c9:2f:5d:18:bc:9d:b9:82:f6:83:
```

```
   dd:2b:29:6c:8d:2c:8c:47:d4:7d:be:de:32:13:85:92:32:bc:
   61:62:6b:e5

   Robert's certificate was signed by the non-root CA in example.net:

   Version: 3 (0x2)
   Serial Number:
       96:a3:84:17:4e:ef:8a:53
   Signature Algorithm: sha1WithRSAEncryption
   Issuer: C=US, ST=California, L=San Jose, O=sipit,
            OU=Test CA for example.net,
            CN=example.net
   Validity
       Not Before: Feb  7 20:21:13 2011 GMT
       Not After : Jan 14 20:21:13 2111 GMT
   Subject: C=US, ST=California, L=San Jose, O=sipit, CN=robert
   Subject Public Key Info:
       Public Key Algorithm: rsaEncryption
       RSA Public Key: (2048 bit)
           Modulus (2048 bit):
               00:d3:dc:14:69:6b:71:09:2c:0b:0f:9d:95:08:c1:
               64:20:66:ef:9f:9c:30:06:30:39:eb:14:16:da:19:
               cc:41:4d:b1:cf:f8:53:5b:a5:0d:76:ec:97:ba:16:
               10:9f:ed:57:b5:fb:6d:4b:9f:8f:d0:9f:0e:15:a7:
               3e:88:c4:e4:ef:35:d1:63:91:20:68:18:f4:8e:3b:
               b4:0f:03:3e:a0:00:d6:c3:26:e7:57:8e:21:92:a3:
               7a:2d:21:44:48:db:01:b9:54:e8:dc:d6:e3:d1:b3:
               f2:4b:26:0f:3f:d4:99:63:e4:7e:14:0a:b2:73:1c:
               5f:3b:41:36:e9:9a:70:be:f7:4f:08:6b:4a:db:44:
               02:e8:bb:50:66:2c:98:94:45:9e:7e:01:0e:9d:c3:
               a9:03:b7:28:15:28:c3:cd:a2:ad:ab:07:f6:ff:69:
               f4:ec:ba:7f:4b:bd:9b:28:8c:0d:87:e2:66:d1:24:
               34:e5:77:be:89:f1:c9:76:4c:37:34:3a:bc:d9:9c:
               36:f5:28:60:01:29:5c:f4:1e:7a:15:19:34:81:1c:
               cf:1a:06:5c:0f:f9:81:67:dc:50:09:e2:a8:d7:9d:
               9f:35:6e:ff:a6:a8:80:74:6c:f8:a1:0a:f3:bb:2b:
               b6:51:8c:21:bc:06:72:59:d0:95:42:d3:02:2c:ce:
               f9:23
           Exponent: 65537 (0x10001)
   X509v3 extensions:
       X509v3 Subject Alternative Name:
           URI:sip:robert@example.net, URI:im:robert@example.net,
              URI:pres:robert@example.net
       X509v3 Basic Constraints:
           CA:FALSE
       X509v3 Subject Key Identifier:
           A6:42:BD:62:0D:6B:BF:EE:67:D4:C7:BC:09:3F:0B:3A:12:AB:19:CE
       X509v3 Authority Key Identifier:
```

```
            72:70:CF:66:1E:23:A5:38:FC:6F:40:8F:86:8A:AF:E0:B9:6F:E9:C3

        X509v3 Key Usage:
            Digital Signature, Non Repudiation, Key Encipherment
        X509v3 Extended Key Usage:
            E-mail Protection, 1.3.6.1.5.5.7.3.20
        Signature Algorithm: sha1WithRSAEncryption
    25:99:ea:1a:1e:96:6d:4e:b1:9c:5a:43:77:ea:3a:a7:a1:b7:
    22:db:b9:d4:9a:1e:17:f7:13:2e:b2:ca:80:dd:c9:a5:db:61:
    41:c6:8b:65:ae:0e:fc:9a:46:77:16:e0:e2:3d:1d:20:3c:e5:
    d5:e0:b8:03:41:4f:e7:69:bf:e0:4c:dd:cc:c4:51:b1:da:2f:
    ad:58:e1:ed:c6:5b:04:ea:1e:af:9a:89:cd:be:60:3c:9a:30:
    51:7f:99:5a:6b:5c:8f:5a:d4:b8:ce:b5:8b:31:74:70:b3:cc:
    5c:04:90:d8:8d:b6:75:55:fb:c1:d8:e8:db:cf:3d:80:e4:8d:
    2f:7e:b9:2b:a2:9e:9f:1e:6f:d0:4e:6e:f7:f0:a6:61:3b:9e:
    9b:4b:78:6b:84:37:ad:93:19:0d:7f:46:5a:18:74:89:8b:a8:
    1a:75:bf:db:df:25:43:4b:57:ab:a1:19:2e:7c:7b:b9:b5:50:
    ef:2c:1f:5c:18:8f:6c:66:83:61:eb:25:a3:21:81:2c:61:3b:
    ee:8c:18:1a:89:9a:29:0d:5c:5b:38:f3:71:3d:61:f0:3f:80:
    33:90:f2:60:53:48:fb:7a:65:c9:5f:1f:a3:e8:75:42:42:f5:
    ad:db:60:29:c6:0f:3c:68:00:7a:2b:38:db:c7:17:b9:4e:d8:
    90:d8:52:bc

    Certificate for CA for example.net in PEM format:

    -----BEGIN CERTIFICATE-----
    MIIDzzCCAregAwIBAgIJAJajhBdO74pSMA0GCSqGSIb3DQEBBQUAMHAxCzAJBgNV
    BAYTAlVTMRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEO
    MAwGA1UECgwFc2lwaXQxKTAnBgNVBAsMIFNpcGl0IFRlc3QgQ2VydGlmaWNhdGUg
    QXV0aG9yaXR5MCAXDTExMDIwNzIwMjExM1oYDzIxMTEwMTE0MjAyMTEzWjB9MQsw
    CQYDVQQGEwJVUzETMBEGA1UECBMKQ2FsaWZvcm5pYTERMA8GA1UEBxMIU2FuIEpv
    c2UxDjAMBgNVBAoTBXNpcGl0MSAwHgYDVQQLExdUZXN0IENBIGZvciBleGFtcGxl
    Lm5ldDEUMBIGA1UEAxMLZXhhbXBsZS5uZXQwggEiMA0GCSqGSIb3DQEBAQUAA4IB
    DwAwggEKAoIBAQDURmVR+IQctZNHpRUUBuzcKneTEV51FNKIVL0WUN1BP34q5CbV
    ozOwXjcd5ZY3HBxpgKTv/SJ4187Tw96W+4cwiLwGFIBd86vXZD4HMdyXxdYZJrx9
    C/jeXvkP3JpFDyiN3foVVtU1FyiA0vwf1pWVQg4sRzhTrf0OJP2jQzODUmVU2kjY
    3IZC1SasHVJUCFLlP0p2lXeNxvIz8BiHyPxbVF3dZfFc9cj0NlSKtntv+FX42Njf
    qXtARUySD6qyLKGoZNWZIh4oeKDY5VFkPwMUqRJHYYTWsGkaa6Nu2MrOQ1CtV5Yr
    hxXZwhEDsILU8IC/3UT09jkKK+NN0/XnqjTlAgMBAAGjXTBbMAwGA1UdEwQFMAMB
    Af8wHQYDVR0OBBYEFHJwz2YeI6U4/G9Aj4aKr+C5b+nDMB8GA1UdIwQYMBaAFJVF
    fl8r6mWYEpEE82PHaJpYFncnMAsGA1UdDwQEAwICBDANBgkqhkiG9w0BAQUFAAOC
    AQEAcHPAZZwvCTk51qRbled7QzS1ubJdduvvh+Altmir7vj3hcQhR7tsaGL/+IQe
    RFowTs6Xkcw9Q0qLtyUmCGPGcUrBlDWBZt4jneM33jGA7Vi3B6fqh9PM2htiyYLC
    F+YtIOSyaRTLBUM0b7UsYNhEQ/nm6T18VKK52R59Z7s/MjENwYh4qGc59dI+CPc4
    hKaPwj4Azl+0yNqhtS/CiWCkOiu+mOBENK/sf3Mm8ZRbOQm5n5PCnXqWL4JmyE32
    24cAjrwquVFzbMz/5TElsUrQmqnDZTUhiXY9OfiEQqYDDrXJL10YvJ25gvaD3Ssp
    bI0sjEfUfb7eMhOFkjK8YWJr5Q==
    -----END CERTIFICATE-----
```

Private key for CA for example.net:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEA1EZlUfiEHLWTR6UVFAbs3Cp3kxFedRTSiFS9FlDdQT9+KuQm
1aMzsF43HeWWNxwcaYCk7/0ieNfO08PelvuHMIi8BhSAXfOr12Q+BzHcl8XWGSa8
fQv43l75D9yaRQ8ojd36FVbVNRcogNL8H9aVlUIOLEc4U639DiT9o0Mzg1JlVNpI
2NyGQtUmrB1SVAhS5T9KdpV3jcbyM/AYh8j8W1Rd3WXxXPXI9DZUirZ7b/hV+NjY
36l7QEVMkg+qsiyhqGTVmSIeKHig2OVRZD8DFKkSR2GE1rBpGmujbtjKzkNQrVeW
K4cV2cIRA7CC1PCAv91E9PY5CivjTdP156o05QIDAQABAoIBADp/7/pIH7h9vcn3
z7hGNE50kaGBHuPrSh3yJG4a+O67XbzaRW2I3XzUaiIeHGixoY7duha9Txu4dbJc
f2JijR4uAIs4aSv7NDdW09VNw3o8NkWWLEnV288Eo2Tgqc8wXz/BleL9nCJWcH4Y
Jw1rKKwKmTdQpVBCWcPlI9UzduXQdZfBbrsL6+OZ+F3kbvUwYAVhhUuBS9sf4Xib
5GA2CDLPm433giOS3yr9KigpcLvbhAhMiPTXJ6i65m9xGGCcjhxP/drOH0cNczRD
yW0FCbaNRJUg9kEVu+n3uG1aVfOnU7RqcblFXgO7ea7G+mfp3Cfm744kvFEXz04k
8WLW6gECgYEA9lK9mKhMUeB1+xPJB4Za5QvrFc7nLt8ee7/aTNcyMI0l3uXyPDPj
TNEfgaRobptmwd2HVtXjlQ54fE+pE+qS8dOORh2VFoWi91zI4C8WnM/6j5P+QiXY
tcZDPF22bmsSW7uaQyaOhUfIMhzox1BbUH5q5YrcA5DmmQtaxcIZ+IECgYEA3J07
6DamIgy0eJO2GKHU/Hy8RvQZgauzCtmqmLQrWZeOmx9hORe1a71QU5F6Y3HQRcTD
RDDdJua9Y8BJ0WTkasbRgxjmHQlf4pUdT6ycfWgISbcCNFTosgPH+/OZPEh4DKlO
rbldUzHPuZdo2Q72KtSPMk+ikny2lCZ9cm2mKmUCgYEAsGoX4fJ/HpDMzrKf4qTG
Co8bojXZ+wbPVT/Vf/0LtBwTCG3VrGpZG5YWo4n1RWpFEQmwuW9cnE+N2TJQXLQ+
47Vpiyv6r/OsAM9SCsWOw2ZtBFGw4v0qFR3W37AaTUCgGFTnKbq+jhQX/FQaH02c
6KxxsM5fvqoTjX7FVycp5IECgYA4Tq1WpHQcpq99Qv4sJUnuM4v+dBj6fq9Q6qNf
HEUgNc2BDC5NWx7D4+rXmX7qWMc2t3S7N9mKL0RRbGeq2RxvoFUjJ7y71oOxmiuE
BWNfoqjS37HhV3aY0Nw/EzqeJ0T0vlXFg1Utgb4p+VoaZHYyElSGG8s7pjcXcwd7
qD7L/QKBgQCeDLKx5T1d/EqwW8KNK5qD/5lG/T0zu3MCDlzCjfs2BHMasv5RALd+
unMMANDElPHOFs7fSmCfspN8Y7+W15/k9WugpwQfST2Y8dSRVdPFp1FRt8u25yX2
mdRbU3vJSiAqPEEpKpBolXPxLOeLGvoTHFWSazgmCPIKKxq0wL+0+w==
-----END RSA PRIVATE KEY-----
```

Robert's certificate:

```
-----BEGIN CERTIFICATE-----
MIIEJjCCAw6gAwIBAgIJAJajhBdO74pTMA0GCSqGSIb3DQEBBQUAMH0xCzAJBgNV
BAYTAlVTMRMwEQYDVQQIEwpDYWxpZm9ybmlhMREwDwYDVQQHEwhTYW4gSm9zZTEO
MAwGA1UEChMFc2lwaXQxIDAeBgNVBAsTF1Rlc3QgQ0EgZm9yIGV4YW1wbGUubmV0
MRQwEgYDVQQDEwtleGFtcGxlLm5ldDAgFw0xMTAyMDcyMDIxMTNaGA8yMTExMDEx
NDIwMjExM1owVjELMAkGA1UEBhMCVVMxEzARBgNVBAgTCkNhbGlmb3JuaWExETAP
BgNVBAcTCFNhbiBKb3NlMQ4wDAYDVQQKEwVzaXBpdDEPMA0GA1UEAxMGcm9iZXJ0
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA09wUaWtxCSwLD52VCMFk
IGbvn5wwBjA56xQW2hnMQU2xz/hTW6UNduyXuhYQn+1XtfttS5+P0J8OFac+iMTk
7zXRY5EgaBj0jju0DwM+oADWwybnV44hkqN6LSFESNsBuVTo3Nbj0bPySyYPP9SZ
Y+R+FAqycxxfO0E26ZpwvvdPCGtK20QC6LtQZiyYlEWefgEOncOpA7coFSjDzaKt
qwf2/2n07Lp/S72bKIwNh+Jm0SQ05Xe+ifHJdkw3NDq82Zw29ShgASlc9B56FRk0
gRzPGgZcD/mBZ9xQCeKo152fNW7/pqiAdGz4oQrzuyu2UYwhvAZyWdCVQtMCLM75
IwIDAQABo4HNMIHKMFEGA1UdEQRKMEiGFnNpcDpyb2JlcnRAZXhhbXBsZS5uZXSG
FWltOnJvYmVydEBleGFtcGxlLm5ldIYXcHJlczpyb2JlcnRAZXhhbXBsZS5uZXQw
CQYDVR0TBAIwADAdBgNVHQ4EFgQUpkK9Yg1rv+5n1Me8CT8LOhKrGc4wHwYDVR0j
BBgwFoAUcnDPZh4jpTj8b0CPhoqv4Llv6cMwCwYDVR0PBAQDAgXgMB0GA1UdJQQW
MBQGCCsGAQUFBwMEBggrBgEFBQcDFDANBgkqhkiG9w0BAQUFAAOCAQEAJZnqGh6W
bU6xnFpDd+o6p6G3Itu51JoeF/cTLrLKgN3JpdthQcaLZa4O/JpGdxbg4j0dIDzl
1eC4A0FP52m/4EzdzMRRsdovrVjh7cZbBOoer5qJzb5gPJowUX+ZWmtcj1rUuM61
izF0cLPMXASQ2I22dVX7wdjo2889gOSNL365K6Kenx5v0E5u9/CmYTuem0t4a4Q3
rZMZDX9GWhh0iYuoGnW/298lQ0tXq6EZLnx7ubVQ7ywfXBiPbGaDYesloyGBLGE7
7owYGomaKQ1cWzjzcT1h8D+AM5DyYFNI+3plyV8fo+h1QkL1rdtgKcYPPGgAeis4
28cXuU7YkNhSvA==
-----END CERTIFICATE-----
```

Robert's private key:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA09wUaWtxCSwLD52VCMFkIGbvn5wwBjA56xQW2hnMQU2xz/hT
W6UNduyXuhYQn+1XtfttS5+P0J8OFac+iMTk7zXRY5EgaBj0jju0DwM+oADWwybn
V44hkqN6LSFESNsBuVTo3Nbj0bPySyYPP9SZY+R+FAqycxxfO0E26ZpwvvdPCGtK
20QC6LtQZiyYlEWefgEOncOpA7coFSjDzaKtqwf2/2n07Lp/S72bKIwNh+Jm0SQ0
5Xe+ifHJdkw3NDq82Zw29ShgASlc9B56FRk0gRzPGgZcD/mBZ9xQCeKo152fNW7/
pqiAdGz4oQrzuyu2UYwhvAZyWdCVQtMCLM75IwIDAQABAoIBAAv+Q3GMUYPRaHbj
1tH+EKr86MfCUb2n8T9rjbefCj8QJOa/CgkAGPkIf7ZbFWnYR8TXjOJhEAUhW+zB
4PphGwynoUjfqFP8RavfmVvYNS1dnsrBYwtD0oa4lmwDnBf7vec99Ui7KX5vj2HN
r8NPR7et8a00xdFaY9G46WDkC0nkH8AqMMymY/Vu2KpH0f01hTpFLmxS7We+d3Uq
mva15GUc8+EL079uphokchr4E0036Ce4luCnqQfOUAKcXCMYK27lG5uue620IXLE
CqeevZPEn8eqWhSNGl981CF15AEb0tApMcMwrfcbpnQMHQuyQHm2XVewgF0gQGLn
UA0i6NECgYEA9TrFg3Kuw1Vfi+kztX6IMjW07YgN443NtB/9+sXKoc0Iz6LoPbOT
VHSVqHHpjicicBUyUa77Kr61HAv7AV0s2FRHAb3M7wOVYGkT52+12o4FH6EMU42G
ISAcsS4vCfHhYq1T0hC91bIY1XXxuBrpo0yb1RkEaSALHN6arAEgWccCgYEA3Sod
gEcahQEnu5P8UY5j9yFaBRqVxdQKWnO2trkfLkyVgtvn7ES31EGojVHg23nr5IsK
IpwFgBiQvEGUgV3dR0Jc5sZTETOweWBLebC/CtZfnhBcCNx8jwX5m/CtTzMHuxVs
VJ1WpUDn+K7+G8KIK0+Kp5QdOCxXptHRLkGPBcUCgYAVgCulFL8B3VBdQfsIpKlo
TZEpak5dbydj7ZIlFIZpnUJyggP+tOnr87TTafliP0gjr5gT1VWsL8BNTzeYrQSr
iugW3P9EzXmhVFUsa3z0RpNobIRaJwRljx0046m4I37xWeUJe/JI9C59OLQSwjlN
2f+ntWPPm8GdrF6/SfH+LQKBgQCyDaf2kEf/cHCmiXuHxVUhrs4kccTGofE75RDi
hqNdyPZNhfFvu9srnTivnY2j5MJPGsksF+Qtvpk3lqySghkVt43HlT9nB/A5p5bb
/7muZexQ+ua9k5UMKElOjDNbIcBFk/fFH26UWG7pPSkC/FhYVg9Q3uOvR7PBcAYy
cUFN6QKBgBw2k5SDvun41wNV4wxGEli9ia+i4lzg8pwJ1DUxnOcDvlDGzAzCNtW9
wPoR+jvhK6V6X1mI0tqqcYZ07pC3CJBEtAckHj2Ik+ZAEjQMf+eH62Rcv6Sbozq0
5dFCBZwzIe2IQomg3J8+OyILSs/uzFkjGjloJIrP+OtPKSrfR+/Y
-----END RSA PRIVATE KEY-----
```

Appendix C. Message Dumps

This section contains a base64-encoded, gzipped, compressed tar file of various Cryptographic Message Syntax (CMS) messages used in this document. Saving the data in a file foo.tgz.b64 then running a command like "openssl base64 -d -in foo.tgz.b64 | tar xfz -" would recover the CMS messages and allow them to be used as test vectors.

```
-- BEGIN MESSAGE ARCHIVE --
H4sIAIpaUE0CA+ybeUATxx7HCSCIHIpoqSIQvFECu5tsDhAEDATQhCsQExTZ
JBtIyGUSIERERERU8i1ZRqVYERVHUCqKiUBWP1vusXCJeeIv3LfpCaRUpSF8f
tJXH/JPdmd3fTjYz8/n+fr8JT6LEKSVCCYqTKCMd+YhKp/0LAABEAgHb8Eki
wp98NhSIQACxIAhDBACGIRDCAiCBQCTqYAGdv6HEKFWIQtsVrkKISD9zXVvt
jd8F++HzCyl0r+BgD5oXVimUO0fHSITRMndUjUjkYtRRiqqwwb4BTpAjYNoj
VIg4/37mxBwTgAUp2iNHyBFyBmEAAF24CkTKi3LVUKJoBO5YHJ9MggkaHAUi
CxASgSvAc3kwgQDgQBzu9zYXhVymULnCAImgfQAdUeO8ZYo4RMFXOmNJ2hqm
zBk7quV+uZn28FbIJL+1C8QxAkH8h3aeTOLmokIiXXkIWSAgEHimPcYgYjHO
l+qMZYui49gsdpw/ky9mM33V2mOAwWTDdCpPQ6eFSugsuppOjYbZIraj9rZg
```

```
dLIzlkwg4bG/vSfTHh48HipXOWMlMWKVUI4oVE5KYaQU5TtgVaha5SQXI0Kp
AxaRy8VCHqISyqRO8miekoRrmGOf1iv5cocmZhCxClVItU2xqPbJMqkKlapw
zHg5+sdnuXBlMVI+ooh3JQkAIoULAhRAwKMIINBForUujnRVRiGgi1whU8l4
MrHrwD92p8EQoopRoAM/PmwcKo1URWlffsPbN+2BwzW33rxfH79xkxbtOFAK
UAXOS8qT8YXSSGcsVyjV9rXBpA8qFsvs/ozpz/TYRYpIUNdfFy1HOUn58U6q
UCmXKYUN92gNqFQIL0qirXeJQqR8sbYnrgp0coxQoX1/AqEYbc3KZ78AkIw5
b2A0IsUn5YUpxlA3MxlzSFu1XxeDAY0AQ4NuI830dPsDxh8vwYDJmCztJd9r
LwGmAnLQGDAy0AvRN7DQDQkGLQDzhpPuJr8OUaFAppAKEdAc6NlQa2jSPRiR
Yv1kShQ0A0waqkx7mHTTjnihCrQHhjVUGJtggxvOsUxUqcKOQRUqoaDhtaFY
jxhVlEwhVMXrGumkZ8+0ZDyYS//YQ9MPPcRguunoJ2N0VHQ7yoWrVx0AzsZi
RsQC4fra+ID+26b6nduvt7rf1KzYZQUngRPcselYuGo/vwK/OP3EcL3lUgeT
5wu+dX+cWmm/2bjU7NU50VKHeGmpK/cGM9cqT1D11U6qWM9q8sq6I/fo3247
9cwy/tDPu53Wi8dePMXNftLfavB1d8Eo/9kpe41EmjERI9+Wu45kWR6brjci
VDg9+bX60llnp9fZh+7Mu2VCtCq+WG8137EnZmvFFX0zRsCSmISSCuPqja+J
1+5dXwd7/5i1zHrzbirT+f6Syli/wN1rp2q4e0c4PZ7AeXf0RtHFiXeGPoYt
+2FPLktnKFYq6m2j9ospmHg+vv9Yjo77iXoooaVBicOZ9tDp3EWCKpVIJIpT
iZU4BaqUd4QEaIP/AIEENOM/DAH4Lv7/HeU3pGMhbZ/9xzbn/LjgP8l5HokM
IwCJhFAQMgEGuS1zHk/+SHVlC1h3hgEi2MhugA+QYBgEmoJe2QLpG3vVeA+I
UPAEPMBtwnt/JkdCp3pB/iyGlvehIgY1WsPWeEbRWUFiOjWEwKB6qTmSEAKH
GfiB91oT5I+8bw7HX9HYeZaFJvO/g9R/2/Of9Ef9D+Hhrvn/N+v/Vmdka15A
B6wOrXsBra8X//gS8U+4BC1J79+XKGJTqd3G7y+VodExOF6HRgH+e/8fJgJd
878z+P8CroDHhfFkMiSABDwC2Ir/T+gQ/59PQUEKD0aaTHYG1VviT/OF2CJe
PFvjAdJpgXEMUWA8ncUmMCSh0QzIW0RniiUMmu8X4f8L8ESUC8E8MsCDiBCR
0L7+f3Pr7ej/t2W6y/9vf/8/8HP+v9NPV5eaR6TYVRfWLfhu+1lh9PEH1Yl7
3jz7fs3BmbkOBuG3w2pqtwGq9cbhC6OdH96zemPjG4apsSHkbVgcoUcehTFK
OWy7cOCpAU/puqGxdgMgZPODtBLNnvfvJ/vNw+utn/rowmayYapSsrV8Dykr
oeeNqsQnR8adKxop2bOKs3FLYdEZeeaIqUusqmMYP5nVzdYPybwytua2/eLE
H1KtuBdqepaMG+w9Fn8y8krfg0ZDhjr1PcK2W385634htWhFRL3aEne7xP2b
u4b1ewyC5s1GzZ/Pt/LaHLkhZNaNd2YF9k604RuOKkWaQTtOVP5UOGTKnAvB
MxPUO5e9HvBypdFIe7tcIT/uSkuD8v/A/2/kfyP7USlPES9Xofx2VgBt8R/C
k5rxn6it6uJ/J+A/SsaDFDJERPB8PpEIEFvhP9wh/Af5EB8gkD7hv8gXr5X7
AIMZDdI1gQCbFaLVAtEgg+YXxWCFxPmz6ABd5B3F0PA+8B8PEYF/iv//A9L+
FNHagm6DLZfG1UGlvcQVlcaiYpkc5eO0SwTyKYklf1QSMBFupKYu+BGJepnJ
ugO1VTa6GB0tHVdo6bhUe/hP0zGoBSjuo1ZeP9XrMm7+knrDUIfaOajutiR+
1V3a4n2njLBOoePccHmXneaWvBeZ59noD3vlpzFMfBpaqZd229hH1D1sCMOD
o7vxgaEUfRl33svcUzD95IYZc0PDjqzPej56ZblXwcnKhcJdgUOTVdhizi77
bUfNr48KjZ0gsN+jCs1aBizgpe9Q7xy1et+m11+dHXyROEVgrS80Of1457vt
tW/N3Q5gfpyvd9ku0U6j/7Vmh5GqICyaIApP8JwVysod4jd9p/skL/eTD49W
SZ2KU4vU5iWxo75POZTx3bDM5IlOg3fnw7OKlDdWzJb1DU3LNfd5GRYb/dB6
q+y8dkzcwoTffGNyxsvum+OjgIQmUgawNjCKSPHBoHr6GF39XrzBx9SKM6eD
a4oSrqXTD71KCoistLQITlMeYfRj+XKQKK/oVeHiy2nwiITFQZutH/DpQeqc
vbIj9dH1R+Zd35uQs2ZJfI11vQnV+q7sweLwNN7g0irbvoHyN18Pm7tpV/GI
rJudnr7/Lv531A6Atv1/uLn/D+DBLv53Av5rf08eAIJ8PAUPCAQAoRX+4zuE
/xREO6i4hE/9fzaBzvJV+7NC8P5MD5AhiobpLHocR8KOY0C+eAbkpWEwfWF/
Ju+L8P/xRATSyis+Hw8QBGSY3E7+P6HR/29uvR39/7ZM/3v9//JP/f8j2qpD
Df5/g3Rp9K676TSPAKzVXpT5r4gANOljVwygpRhA84HZyWMATfnfMMN+DwK0
pxBog/94EoRv7v8TiV3x/87AfxKfCwoIJBIIk0ABv9X4P7Fj+E+GIJBEIjXh
Px3yBTlMLzydGtWw/w/mMDlCjihKxKD6SdgSLzWHqm1j8sX+tI/7/2ACgfJv
```

```
5T+fCBJhlEQgg1wuiof57cJ/kADDjQKgufkOCxh0dJzjC4lBMFqAcvKdjFmr
L0ziPSuQYdfIDmX9vIJ7ro5zN3koOb1nZXDiqzQj2PxgRPfvt3692MPesDw3
H0mRFak32LoZXSW5mZVmmHEWzc6t9900ZeP9gYbHTscbvXB5Yuk6d7DnTupR
zS97Jtkejg3IeTE3/yvh5Ko6cXzQpnFhIJ9SYbN5dIplpR4F7337BfKy5v0I
zDy7YUxd/zmPbLdcnxc0VVBa+1w1Y0BGVC/r8WGZ5CdzcoQFugTSlKP97Yfd
t2TaztDc2oZRG848pK4SbVjCjb1sEjbbgDNGrCPKC/ZZ914Usqo/bXj/+OUX
PHUP6r6ca1RTeHPQnKiHZy3STN8T7+wvs31XNpGZbuJJ+1wIYgGsMP1lVUJ3
sWn+UVD3lmNSwPnbj/Z7mvZ4ek1i49fPd4PGduPzY/cLy0eNLY9VYZLKAiTB
K7aM74m3GMg/XX3D/RnboCgzWqWesPS0xb7C07Dt2bQhY0r5C48vzDPpttsi
gMka8temQZdYbY/tqp8Vq0rvxIKvIg7nF71/PmnMyyzd0mn6eVzNU+dvH2w4
c8XBuLdN0YSMHBfhvHnjYjg78aylKLrrTOlyN1qF+PRFl1SrGZNnmU+Wjk05
G+saWZdicn8BeVBe0g/IrbKKiVnCUFr2IltxryU+mccj+kgCvMfeHSVh95o2
ab7u01UQ5f4O5wr9Q1XXhsfVDPUoM4ms3lTHcGaUbinanG12t/ervoYlR5Kr
h0tLLdPDfYcrZxUnxkwetmuDXt7+3WXblV6S9L2mPfpaTl2+Zxt31HGR5UNe
6rSg8xWj7tNcsne/vbDVlTToHJmT3+v2pl599bIm6Cfu3mzn8F4Ve2XiNp9J
uum46AWJRNo1C3J9SyPzv1arHZv5+bP5H2Lz+A8IduV/OoP+40EkIoSHuDAB
5PNhiNSK/gM7RP+REAjFAxDcVP9R2fFa7RflT/XAM1heEEfk1bAPJM6fyRHT
RV5qBssX4rAC4zg0elf+5/8r/9OS9irq/Q20ye8qef/lYGHqhSE33fW2XYq3
y741iGN17M34p8t81v/80LXGI5uTBsmTX/9wqXxJgtgR8w5fwj1m6D/+kuzO
6Afb33grxmekZs4qHlT2s5Fx2gK/SaPTE/LOX+13S3eH2RPTx4v8InPrAkXL
ylIk99TSw5dnJFzRseW4syNMyu5mv9EvOLLubvz9gtCXzwPzw8dfLbccjs/Z
Bc836zMt7fQUPR2x63T7Z2W1eTHhlx9WX3PLN1h2wTf3GLn7o5ndnu0rDsk5
f6S8fm2e++pVgWEvnA8cOOF8U2LX7XRAjH+f8rjYwxY5Pr2nDDk+cKKyxP16
X90sit+xipDua+sYc8N3H/TzXvk57XUUPlpio55RgRXTilZfXJSaFGC1pfdB
s5D0TSuOD5hWuHzt3rPl05bsKSR3Yz8mrzlhH2NpzbMy/gGXajBNnfSE43YR
3jvQRg9iL6+snz6pwoPn/HV94cw+GG9j97uh9im0eWfqutI/HV8+ZH5wcVGI
ChcpU+Ea17N2VAJt7f8Ggeb5HyIJALr4/3eUdgnod02jL3/+d9i/P/5K/hdP
AqGu+d8J9D8eoUAQl8jnAQJAIMC3ov/xlA7R/6iAyEX4MNo0/wsxhByRL8AQ
BYkY1ECAzmSDDKr2mMWQ0GkhIJ3qAdGpQVFsyZex/xvPhUEYBSGQS9C+ahBs
5/xvM+vtmf9tw3RXSO0fCKl1/Qf8jwOzk+d/Ozbz+2fzv1Dz+B9M6Mr/dgb+
U0gIRERhCpcEESEuvrX9X1CH8B9PoghgLvRJ/lfEjqNr+CI2k41ns4Ki2ZpI
DV0UAtBpvgS6hgcxRNEgW+MnZmi8voj8L0rmC4goyENhPoiS2+n/Xx/zv83N
d+V///4YpK+5sXlQ7qprG9+kHLvb+/jC9FWz3JLOJhz8buf0sYWvLZJEonyC
onxfckRlwZiXgfOhedWnYyUrvZX7qZm93n+1doZqlp1q6uV6z33LdJKzkqYP
XJmVbNpj1sOoBxllEuvBI3PDCzIy3dZXWA8o8zwmWzddMGi4TsIK0Q690YnP
fe4s8oUf1bszJ+a9mHln9LAx9Zeu9qrmHYT9LHjGEtOMXzYpKML56DjqwWww
Ir5oQ/YavXqPIb1rn7yknZzvWTE0bh1ra/+le7utu017fGbLRYtEqxkT5h+0
BYvZN+qlGT8sujc5Z9pwt0FW7lf3RZwKD0vpbpC8fWeVTnWeM2XY1YT0zXNv
H9hlEP65IGTqdxYx6wV9Dpw6cfj92UUTM5MCkoyzd7LmbH8q32LdJxeufmUt
sPcOcre44uI3qPbepldwzo61P+7TDoV+BykDp/YaZ/o0XV9tPouReO1AcX5N
iGX8pMeM2iGeJC/KxOeVrAyG8V+bBl1itcPFqjQwb7Dj7oQ1dUkbQorP8yfW
2htNyt+6Ubbo7LJ4KGzr0XdrlG9rWYzDLpHwxutqG/a3dZG8OBtNcrHN0J6U
GJOcoYsxop0TH+5zCPR+s55IvcY/bH7MOLr+iSSh3m2L46I96u+fWq3BRhVd
OnfN5O2LPhkO/E3DgkqC7g1L7VNSdYD50x2fKsPb3zn+/CM3K3ZGMOmW7tgz
KbdrHEu+pdxzwgRnTutflAO+vbloQNVAv8gZS/IZw3NPXPLTF11OSZsyflKV
Jj09cwhSOj5reG1B/iNJoMFKNWJa7rx+dXbhbOMk89Lc/7RvxzQMAgEARRkw
```

```
   wNSEMOLlFDBVBGMnFhJsYAABZ4LuJUwkJZCgoQQDdcB7Gv768/VRHG01vNNt
   emZ7DOdvjHOoX11ffrLl2/wL8wbDIgAAAAAAAAAAAJBchjiJbgB4AAA=
   -- END MESSAGE ARCHIVE --
```

Authors' Addresses

Cullen Jennings
Cisco Systems
170 West Tasman Drive
Mailstop SJC-21/2
San Jose, CA 95134
USA

Phone: +1 408 421 9990
EMail: fluffy@cisco.com

Kumiko Ono
Columbia University
1214 Amsterdam Avenue
MC 0401
New York, NY 10027
USA

EMail: kumiko@cs.columbia.edu

Robert Sparks
Tekelec
17210 Campbell Road
Suite 250
Dallas, TX 75252
USA

EMail: Robert.Sparks@tekelec.com

Brian Hibbard (editor)
Tekelec
17210 Campbell Road
Suite 250
Dallas, TX 75252
USA

EMail: Brian.Hibbard@tekelec.com