

Examples of S/MIME Messages

Status of This Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

This document gives examples of message bodies formatted using S/MIME. Specifically, it has examples of Cryptographic Message Syntax (CMS) objects and S/MIME messages (including the MIME formatting). It includes examples of many common CMS formats. The purpose of this document is to help increase interoperability for S/MIME and other protocols that rely on CMS.

Table of Contents

1.	Introduction	3
2.	Constants Used in the Examples	3
2.1.	Content of Documents	4
2.2.	Private Keys	4
2.3.	Certificates	13
2.4.	CRLs	33
3.	Trivial Examples	39
3.1.	ContentInfo with Data Type, BER	39
3.2.	ContentInfo with Data Type, DER	39
4.	Signed-data	39
4.1.	Basic Signed Content, DSS	39
4.2.	Basic Signed Content, RSA	44
4.3.	Fancier Signed Content, Detached Content	49
4.4.	Fancier Signed Content	53
4.5.	All RSA Signed Message	68
4.6.	Multiple Signers	75
4.7.	Signing Using SKI	83
4.8.	S/MIME multipart/signed Message	87
4.9.	S/MIME application/pkcs7-mime Signed Message	88
4.10.	SignedData with Attributes	89
4.11.	SignedData with Certificates Only	101
5.	Enveloped-data	109
5.1.	Basic Encrypted Content, TripleDES and RSA	109
5.2.	Basic Encrypted Content, RC2/128 and RSA	110
5.3.	S/MIME application/pkcs7-mime Encrypted Message	112
6.	Digested-data	112
7.	Encrypted-data	113
7.1.	Simple EncryptedData	113
7.2.	EncryptedData with Unprotected Attributes	114
8.	Security Considerations	115
9.	References	115
9.1.	Normative References	115
9.2.	Informative References	115
A.	Binaries of the Examples	116
A.1.	How the Binaries and Extractor Works	116
A.2.	Example Extraction Program	116
B.	Examples in Order of Appearance	118
C.	Acknowledgements	135

1. Introduction

The examples in this document show the structure and format of CMS message bodies, as described in [CMS]. They are useful to implementors who use protocols that rely on CMS, such as the S/MIME message format protocol. There are also examples of simple S/MIME messages [SMIME-MSG] (including the MIME headers).

Every example in this document has been checked by two different implementors. This strongly indicates (but does not assure) that the examples are correct. All CMS implementors must read the CMS document carefully before implementing from it. No one should use the examples in this document as stand-alone explanations of how to create CMS message bodies.

This document explicitly does not attempt to cover many PKIX [PKIX] examples. Documents with examples of that format may be forthcoming. Also, note that [DVCS], which covers PKIX Data Validation and Certification Server Protocols, has examples of formats for its protocol.

The examples shown here were created and validated by many different people over a long period of time. Because of this, some of the dates used in the examples are many years in the past. This, plus the fact that some of the certificates in the examples have very long lifespans, may cause problems in some test situations.

2. Constants Used in the Examples

This section defines the data used in the rest of the document. The names of the constants indicate their use. For example, AlicePrivDSSSign is the private part of Alice's DSS signing key.

- Alice is the creator of the message bodies in this document.
- Bob is the recipient of the messages.
- Carl is a CA.
- Diane sometimes gets involved with these folks.
- Erica also sometimes gets involved.

2.1. Content of Documents

ExContent is the following sentence:

This is some sample content.

That is, it is the string of characters starting with "T" up to and including the ".".

The hex for ExContent is

5468 6973 2069 7320 736f 6d65 2073 616d 706c 6520 636f 6e74 656e 742e

The MD5 hash of ExContent is

9898 cac8 fab7 691f f89d c207 24e7 4a04

The SHA-1 hash of ExContent is

406a ec08 5279 ba6e 1602 2d9e 0629 c022 9687 dd48

2.2. Private Keys

The following private keys are needed to create the samples. To find the public keys, see the certificates in the next section.

AlicePrivDSSSign =

```

0 30 331: SEQUENCE {
  4 02 1: INTEGER 0
  7 30 299: SEQUENCE {
11 06 7: OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
      : (ANSI X9.57 algorithm)
20 30 286: SEQUENCE {
24 02 129: INTEGER
      : 00 81 8D CD ED 83 EA 0A 9E 39 3E C2
      : 48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
      : 53 C5 AB 84 08 4F FF 94 E1 73 48 7E
      : 0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
      : 2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
      : DC 5F 69 8A E4 75 D0 37 0C 91 08 95
      : 9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
      : 8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
      : C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
      : 78 BD FF 9D B0 84 97 37 F2 E4 51 1B
      : B5 E4 09 96 5C F3 7E 5B DB
156 02 21: INTEGER
      : 00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
      : B8 37 21 2B 62 8B F7 93 CD

```

```

179 02 128:    INTEGER
:             26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:             4B 59 6A 4C 76 23 39 04 02 35 5C F2
:             CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:             AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:             7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:             3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:             E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:             01 7C 6D 49 89 11 89 36 44 BD F8 C8
:             95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:             1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:             D1 81 4A 60 39 BA 36 39
:             }
:         }
310 04 23:    OCTET STRING, encapsulates {
312 02 21:    INTEGER
:             00 BB 44 46 D1 A5 C9 46 07 2E D0 FE
:             7A D6 92 07 F0 9A 85 89 3F
:         }
:     }

AlicePrivRSASign =
  0 30 630: SEQUENCE {
    4 02 1:   INTEGER 0
    7 30 13:  SEQUENCE {
      9 06 9:   OBJECT IDENTIFIER
:             rsaEncryption (1 2 840 113549 1 1 1)
:             (PKCS #1)
    20 05 0:   NULL
:         }
    22 04 608: OCTET STRING, encapsulates {
    26 30 604:  SEQUENCE {
    30 02 1:   INTEGER 0
    33 02 129: INTEGER
:             00 E0 89 73 39 8D D8 F5 F5 E8 87 76
:             39 7F 4E B0 05 BB 53 83 DE 0F B7 AB
:             DC 7D C7 75 29 0D 05 2E 6D 12 DF A6
:             86 26 D4 D2 6F AA 58 29 FC 97 EC FA
:             82 51 0F 30 80 BE B1 50 9E 46 44 F1
:             2C BB D8 32 CF C6 68 6F 07 D9 B0 60
:             AC BE EE 34 09 6A 13 F5 F7 05 05 93
:             DF 5E BA 35 56 D9 61 FF 19 7F C9 81
:             E6 F8 6C EA 87 40 70 EF AC 6D 2C 74
:             9F 2D FA 55 3A B9 99 77 02 A6 48 52
:             8C 4E F3 57 38 57 74 57 5F
    165 02 3:   INTEGER 65537
    170 02 128: INTEGER
:             00 A4 03 C3 27 47 76 34 34 6C A6 86

```

```

:      B5 79 49 01 4B 2E 8A D2 C8 62 B2 C7
:      D7 48 09 6A 8B 91 F7 36 F2 75 D6 E8
:      CD 15 90 60 27 31 47 35 64 4D 95 CD
:      67 63 CE B4 9F 56 AC 2F 37 6E 1C EE
:      0E BF 28 2D F4 39 90 6F 34 D8 6E 08
:      5B D5 65 6A D8 41 F3 13 D7 2D 39 5E
:      FE 33 CB FF 29 E4 03 0B 3D 05 A2 8F
:      B7 F1 8E A2 76 37 B0 79 57 D3 2F 2B
:      DE 87 06 22 7D 04 66 5E C9 1B AF 8B
:      1A C3 EC 91 44 AB 7F 21
301 02 65:    INTEGER
:      00 F6 D6 E0 22 21 4C 5F 0A 70 FF 27
:      FC E5 B3 50 6A 9D E5 0F B5 85 96 C6
:      40 FA A8 0A B4 9B 9B 0C 55 C2 01 1D
:      F9 37 82 8A 14 C8 F2 93 0E 92 CD A5
:      66 21 B9 3C D2 06 BF B4 55 31 C9 DC
:      AD CA 98 2D D1
368 02 65:    INTEGER
:      00 E8 DE B0 11 25 09 D2 02 51 01 DE
:      8A E8 98 50 F5 77 77 61 A4 45 93 6B
:      08 55 96 73 5D F4 C8 5B 12 93 22 73
:      8B 7F D3 70 7F F5 A4 AA BB 74 FD 3C
:      22 6A DA 38 91 2A 86 5B 6C 14 E8 AE
:      4C 9E FA 8E 2F
435 02 65:    INTEGER
:      00 97 4C F0 87 9B 17 7F EE 1B 83 1B
:      14 B6 0B 6A 90 5F 86 27 51 E1 B7 A0
:      7F F5 E4 88 E3 59 B9 F9 1E 9B D3 29
:      77 38 22 48 D7 22 B1 25 98 BA 3D 59
:      53 B7 FA 1E 20 B2 C8 51 16 23 75 93
:      51 E7 AB CD F1
502 02 64:    INTEGER
:      2C F0 24 5B FA A0 CD 85 22 EA D0 6E
:      4F FA 6C CD 21 D3 C8 E4 F1 84 44 48
:      64 73 D7 29 8F 7E 46 8C EC 15 DE E4
:      51 B3 94 E7 2C 99 2D 55 65 7B 24 EA
:      A3 62 1F 3E 6C 4D 67 41 11 3B E1 BE
:      E9 83 02 83
568 02 64:    INTEGER
:      58 88 D9 A1 50 38 84 6A AB 03 BC BB
:      DF 4B F4 9C 6F B8 B4 2A 25 FB F6 E4
:      05 2F 6E E2 88 89 21 6F 4B 25 9E D0
:      AB 50 93 CA BF 40 71 EC 21 25 C5 7F
:      FB 02 E9 21 96 B8 33 CD E2 C6 95 EE
:      6F 8D 5F 28
:      }
:    }
:  }

```

```

BobPrivRSAEncrypt =
  0 30 645: SEQUENCE {
    4 02 1: INTEGER 0
    7 30 13: SEQUENCE {
      9 06 9: OBJECT IDENTIFIER
        : rsaEncryption (1 2 840 113549 1 1 1)
        : (PKCS #1)
    20 05 0: NULL
      : }
    22 04 608: OCTET STRING, encapsulates {
    26 30 604: SEQUENCE {
      30 02 1: INTEGER 0
      33 02 129: INTEGER
        : 00 A9 E1 67 98 3F 39 D5 5F F2 A0 93
        : 41 5E A6 79 89 85 C8 35 5D 9A 91 5B
        : FB 1D 01 DA 19 70 26 17 0F BD A5 22
        : D0 35 85 6D 7A 98 66 14 41 5C CF B7
        : B7 08 3B 09 C9 91 B8 19 69 37 6D F9
        : 65 1E 7B D9 A9 33 24 A3 7F 3B BB AF
        : 46 01 86 36 34 32 CB 07 03 59 52 FC
        : 85 8B 31 04 B8 CC 18 08 14 48 E6 4F
        : 1C FB 5D 60 C4 E0 5C 1F 53 D3 7F 53
        : D8 69 01 F1 05 F8 7A 70 D1 BE 83 C6
        : 5F 38 CF 1C 2C AA 6A A7 EB
    165 02 3: INTEGER 65537
    170 02 128: INTEGER
      : 67 CD 48 4C 9A 0D 8F 98 C2 1B 65 FF
      : 22 83 9C 6D F0 A6 06 1D BC ED A7 03
      : 88 94 F2 1C 6B 0F 8B 35 DE 0E 82 78
      : 30 CB E7 BA 6A 56 AD 77 C6 EB 51 79
      : 70 79 0A A0 F4 FE 45 E0 A9 B2 F4 19
      : DA 87 98 D6 30 84 74 E4 FC 59 6C C1
      : C6 77 DC A9 91 D0 7C 30 A0 A2 C5 08
      : 5E 21 71 43 FC 0D 07 3D F0 FA 6D 14
      : 9E 4E 63 F0 17 58 79 1C 4B 98 1C 3D
      : 3D B0 1B DF FA 25 3B A3 C0 2C 98 05
      : F6 10 09 D8 87 DB 03 19
    301 02 65: INTEGER
      : 00 D0 C3 22 C6 DE A2 99 18 76 8F 8D
      : BC A6 75 D6 66 3F D4 8D 45 52 8C 76
      : F5 72 C4 EB F0 46 9A F1 3E 5C AA 55
      : 0B 9B DA DD 6B 6D F8 FC 3B 3C 08 43
      : 93 B5 5B FE CE EA FD 68 84 23 62 AF
      : F3 31 C2 B9 E5
    368 02 65: INTEGER
      : 00 D0 51 FC 1E 22 B7 5B ED B5 8E 01
      : C8 D7 AB F2 58 D4 F7 82 94 F3 53 A8
      : 19 45 CB 66 CA 28 19 5F E2 10 2B F3

```

```

      :      8F EC 6A 30 74 F8 4D 11 F4 A7 C4 20
      :      B5 47 21 DC 49 01 F9 0A 20 29 F0 24
      :      08 84 60 7D 8F
435 02 64:      INTEGER
      :      34 BA 64 C9 48 28 57 74 D7 55 50 DE
      :      6A 48 EF 1B 2A 5A 1C 48 7B 1E 21 59
      :      C3 60 3B 9B 97 A9 C0 EF 18 66 A9 4E
      :      62 52 38 84 CE E5 09 88 48 94 69 C5
      :      20 14 99 5A 57 FE 23 6C E4 A7 23 7B
      :      D0 80 B7 85
501 02 65:      INTEGER
      :      00 9E 2F B3 37 9A FB 0B 06 5D 57 E1
      :      09 06 A4 5D D9 90 96 06 05 5F 24 06
      :      40 72 9C 3A 88 85 9C 87 0F 9D 62 12
      :      88 16 68 A8 35 1A 1B 43 E8 38 C0 98
      :      69 AF 03 0A 48 32 04 4E E9 0F 8F 77
      :      7D 34 30 25 07
568 02 64:      INTEGER
      :      57 18 67 D6 0A D2 B5 AB C2 BA 7A E7
      :      54 DA 9C 05 4F 81 D4 EF 01 89 1E 32
      :      3D 69 CB 31 C4 52 C8 54 55 25 00 3B
      :      1C 2A 7C 26 50 D5 E9 A6 D7 77 CB CF
      :      15 F5 EE 0B D5 8D EE B3 AF 4C A1 7C
      :      63 46 41 F6
      :      }
      :      }
634 A0 13:  [0] {
636 30 11:    SEQUENCE {
638 06 3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
      :      (X.509 id-ce (2 5 29))
643 31 4:      SET {
645 03 2:        BIT STRING 0 unused bits
      :        '00001000'B (bit 3)
      :        Error: Spurious zero bits in bitstring.
      :      }
      :    }
      :  }
      : }

```

CarlPrivDSSSign =

```

 0 30 330: SEQUENCE {
 4 02 1:   INTEGER 0
 7 30 299: SEQUENCE {
11 06 7:   OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
      :   (ANSI X9.57 algorithm)
20 30 286: SEQUENCE {
24 02 129:   INTEGER
      :   00 B6 49 18 3E 8A 44 C1 29 71 94 4C

```



```

:      01 C4 12 C1 7A 79 CB 54 4D AB 1E 81
:      FB C6 4C B3 0E 94 09 06 EB 01 D4 B1
:      C8 71 4B C7 45 C0 50 25 5D 9C FC DA
:      E4 6D D3 E2 86 48 84 82 7D BA 15 95
:      4A 16 F6 46 ED DD F6 98 D2 BB 7E 8A
:      0A 8A BA 16 7B B9 50 01 48 93 8B EB
:      25 15 51 97 55 DC 8F 53 0E 10 A9 50
:      FC 70 B7 CD 30 54 FD DA DE A8 AA 22
:      B5 A1 AF 8B CC 02 88 E7 8B 70 5F B9
:      AD E1 08 D4 6D 29 2D D6 E9
156 02 21:    INTEGER
:      00 DD C1 2F DF 53 CE 0B 34 60 77 3E
:      02 A4 BF 8A 5D 98 B9 10 D5
179 02 128:   INTEGER
:      0C EE 57 9B 4B BD DA B6 07 6A 74 37
:      4F 55 7F 9D ED BC 61 0D EB 46 59 3C
:      56 0B 2B 5B 0C 91 CE A5 62 52 69 CA
:      E1 6D 3E BD BF FE E1 B7 B9 2B 61 3C
:      AD CB AE 45 E3 06 AC 8C 22 9D 9C 44
:      87 0B C7 CD F0 1C D9 B5 4E 5D 73 DE
:      AF 0E C9 1D 5A 51 F5 4F 44 79 35 5A
:      73 AA 7F 46 51 1F A9 42 16 9C 48 EB
:      8A 79 61 B4 D5 2F 53 22 44 63 1F 86
:      B8 A3 58 06 25 F8 29 C0 EF BA E0 75
:      F0 42 C4 63 65 52 9B 0A
:      }
:    }
310 04 22:    OCTET STRING, encapsulates {
312 02 20:      INTEGER
:      19 B3 38 A5 21 62 31 50 E5 7F B9 3E
:      08 46 78 D1 3E B5 E5 72
:    }
:  }

CarlPrivRSASign =
  0 30 630: SEQUENCE {
    4 02 1:   INTEGER 0
    7 30 13:  SEQUENCE {
      9 06 9:   OBJECT IDENTIFIER
:         rsaEncryption (1 2 840 113549 1 1 1)
:         (PKCS #1)
    20 05 0:   NULL
:   }
  22 04 608: OCTET STRING, encapsulates {
  26 30 604: SEQUENCE {
    30 02 1:   INTEGER 0
    33 02 129: INTEGER
:      00 E4 4B FF 18 B8 24 57 F4 77 FF 6E

```

```

:      73 7B 93 71 5C BC 33 1A 92 92 72 23
:      D8 41 46 D0 CD 11 3A 04 B3 8E AF 82
:      9D BD 51 1E 17 7A F2 76 2C 2B 86 39
:      A7 BD D7 8D 1A 53 EC E4 00 D5 E8 EC
:      A2 36 B1 ED E2 50 E2 32 09 8A 3F 9F
:      99 25 8F B8 4E AB B9 7D D5 96 65 DA
:      16 A0 C5 BE 0E AE 44 5B EF 5E F4 A7
:      29 CB 82 DD AC 44 E9 AA 93 94 29 0E
:      F8 18 D6 C8 57 5E F2 76 C4 F2 11 60
:      38 B9 1B 3C 1D 97 C9 6A F1
165 02      3:      INTEGER 65537
170 02    129:      INTEGER
:      00 AE 73 E4 5B 5F 5B 66 5A C9 D7 C6
:      EF 38 5F 53 21 2A 2F 62 FE DE 29 9A
:      7A 86 67 36 E7 7D 62 78 75 3D 73 A0
:      BC 29 0E F3 8F BD C3 C9 C9 B6 F8 BA
:      D6 13 9B C3 97 7A CA 6A F0 B8 85 65
:      4E 0F BD A7 A8 F7 54 06 41 BD EB DC
:      20 77 90 DF 61 9B 9A 6F 74 DE EA 3B
:      D4 9C 87 60 ED 76 84 F1 6A 30 37 D5
:      E0 90 16 F8 80 47 C3 19 6B ED 75 77
:      BA 4A ED 39 B6 5D 02 47 3B 5F 1B C8
:      1C AB CB E8 F5 26 3F A4 81
302 02    65:      INTEGER
:      00 FF DF 09 A0 56 0B 42 52 9E C4 4D
:      93 B3 B0 49 BB DE E7 81 7D 28 99 D0
:      B1 48 BA 0B 39 E1 1C 7B 22 18 33 B6
:      40 F6 BF DC AE 1D D0 A1 AD 04 71 5A
:      61 0A 6E 3B CE 30 DA 36 9F 65 25 29
:      BB A7 0E 7F 0B
369 02    65:      INTEGER
:      00 E4 69 68 18 5F F9 57 D0 7C 66 89
:      0F BA 63 1D 72 CB 20 A4 81 76 64 89
:      CD 7D D1 C2 27 A9 2E AC 7A 56 9A 85
:      07 D9 30 03 A3 03 AB 7F 88 92 50 24
:      01 AA 1B 07 1F 20 4C B7 C9 7B 56 F7
:      B6 C2 7E AB 73
436 02    64:      INTEGER
:      57 36 6C 8F 8C 04 76 6C B6 D4 EE 24
:      44 00 F8 80 E2 AF 42 01 A9 0F 14 84
:      F8 E7 00 E0 8F 8C 27 A4 2D 5F A2 E5
:      6D B5 63 C0 AD 44 E9 76 91 A7 19 49
:      2E 46 F8 77 85 4B 3B 87 04 F0 AF D2
:      D8 54 26 95
502 02    64:      INTEGER
:      64 A1 0F AC 55 74 1B BD 0D 61 7B 17
:      03 CD B0 E6 A7 19 1D 80 AF F1 41 48
:      D8 1A B6 88 14 A0 2C 7A C5 76 D4 0F

```

```

:          0E 1F 7A 2A B2 6E 37 04 AB 39 45 73
:          BA 46 A8 0F 8D 82 5F 22 14 05 CF A2
:          A3 F3 7C 83
568 02    64:      INTEGER
:          26 1E 1D 1C A1 98 2B E4 DB 38 E8 57
:          6E 6B 73 19 88 61 3A FA 74 4A 36 8B
:          47 68 5D 50 EB 26 E3 EA 7D 9B 4E 65
:          A9 AF 7B AB 4B 2E 76 51 3D A8 D0 11
:          AB A3 D6 A8 C0 27 36 1D 54 0B AA A7
:          D1 6D 8D FA
:          }
:      }
:  }

```

DianePrivDSSSign =

```

0 30    331: SEQUENCE {
4 02      1:  INTEGER 0
7 30    299: SEQUENCE {
11 06      7:  OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
:          (ANSI X9.57 algorithm)
20 30    286: SEQUENCE {
24 02    129: INTEGER
:          00 B6 49 18 3E 8A 44 C1 29 71 94 4C
:          01 C4 12 C1 7A 79 CB 54 4D AB 1E 81
:          FB C6 4C B3 0E 94 09 06 EB 01 D4 B1
:          C8 71 4B C7 45 C0 50 25 5D 9C FC DA
:          E4 6D D3 E2 86 48 84 82 7D BA 15 95
:          4A 16 F6 46 ED DD F6 98 D2 BB 7E 8A
:          0A 8A BA 16 7B B9 50 01 48 93 8B EB
:          25 15 51 97 55 DC 8F 53 0E 10 A9 50
:          FC 70 B7 CD 30 54 FD DA DE A8 AA 22
:          B5 A1 AF 8B CC 02 88 E7 8B 70 5F B9
:          AD E1 08 D4 6D 29 2D D6 E9
156 02    21: INTEGER
:          00 DD C1 2F DF 53 CE 0B 34 60 77 3E
:          02 A4 BF 8A 5D 98 B9 10 D5
179 02    128: INTEGER
:          0C EE 57 9B 4B BD DA B6 07 6A 74 37
:          4F 55 7F 9D ED BC 61 0D EB 46 59 3C
:          56 0B 2B 5B 0C 91 CE A5 62 52 69 CA
:          E1 6D 3E BD BF FE E1 B7 B9 2B 61 3C
:          ADCB AE 45 E3 06 AC 8C 22 9D 9C 44
:          87 0B C7 CD F0 1C D9 B5 4E 5D 73 DE
:          AF 0E C9 1D 5A 51 F5 4F 44 79 35 5A
:          73 AA 7F 46 51 1F A9 42 16 9C 48 EB
:          8A 79 61 B4 D5 2F 53 22 44 63 1F 86
:          B8 A3 58 06 25 F8 29 C0 EF BA E0 75
:          F0 42 C4 63 65 52 9B 0A

```

```

:      }
:    }
310 04 23:  OCTET STRING, encapsulates {
312 02 21:    INTEGER
:          00 96 95 F9 E0 C1 E0 41 2D 32 0F 8B
:          42 52 93 2A E6 1E 0E 21 29
:        }
:    }

DianePrivRSASignEncrypt =
  0 30 631: SEQUENCE {
    4 02 1:  INTEGER 0
    7 30 13: SEQUENCE {
      9 06 9:  OBJECT IDENTIFIER
:          rsaEncryption (1 2 840 113549 1 1 1)
:          (PKCS #1)
    20 05 0:  NULL
:    }
    22 04 609: OCTET STRING, encapsulates {
    26 30 605: SEQUENCE {
      30 02 1:  INTEGER 0
      33 02 129: INTEGER
:            00 D6 FD B8 C0 70 C6 4C 25 EC EA CF
:            EA 7C BB A2 62 FA F0 E6 32 3A 53 FF
:            B1 92 5A 17 F4 20 E1 99 24 82 0A D0
:            F6 7C FB 44 CA 8B 27 06 F1 7E 26 03
:            A9 76 9D CF EC A0 2C 70 96 F2 83 42
:            F6 D4 B7 28 0A BB F8 BF 4A 4C 19 3F
:            07 DB A0 C1 60 1E B7 7E 67 F7 DE B1
:            C3 60 49 AC 45 D7 F8 C6 EF 08 37 21
:            93 47 EE F0 73 35 72 B0 02 C4 F3 11
:            C3 5E 47 E5 0A B7 83 F1 DB 74 69 64
:            8B 44 1D 95 5D CD 28 C0 85
    165 02 3:  INTEGER 65537
    170 02 128: INTEGER
:           3D BD CD C2 0E 61 14 5B 4B E7 BF 60
:           23 04 2B C5 6B 35 A5 96 45 23 FC 69
:           7D 93 3C 0F D3 25 96 BA 62 52 42 E2
:           96 CF FE 58 80 8F EB B1 8C BD D4 0D
:           65 D0 3A 77 45 24 9E 0C EB 86 80 C3
:           AC 21 11 71 44 E3 B2 A8 A9 2E AC 17
:           D2 A3 84 25 63 B5 BC 2F 1E DD F6 21
:           FF 15 20 24 5B F1 80 2F D5 41 0E 32
:           24 F7 D4 4A 32 9E B9 49 D8 19 8E 3F
:           39 8D 62 BD 80 FC 0C 24 92 93 E4 C3
:           D7 05 91 53 BB 96 B6 41
    301 02 65: INTEGER
:           00 F3 B8 3F 4A D1 94 B0 91 60 13 41

```

```

:          92 0D 8D 44 3F 77 1D FF 96 23 44 08
:          D4 0B 70 C9 1A AF E9 90 94 F2 B0 D5
:          5F 4F 19 85 50 A1 90 91 AE BD 05 76
:          52 B3 22 D8 A8 7C 8E 54 7F 00 72 4F
:          36 75 68 73 B5
368 02    65:      INTEGER
:          00 E1 D2 E7 11 57 06 AE 72 95 22 16
:          AA 02 B4 5A ED 4E 9D 82 11 4F 96 3C
:          86 C9 10 8D 56 7B 31 75 79 69 E7 75
:          68 38 00 4B 2E D2 26 32 DD B1 E2 E0
:          2C 54 80 0A 75 BA D1 66 96 1B B0 0E
:          A0 7E D2 BB 91
435 02    65:      INTEGER
:          00 AF B6 BC DB 22 73 43 41 EC B4 B5
:          67 A9 A1 99 FC EF D2 8E FD 1D FB E5
:          29 8B FE 0A DF D4 C8 5E 57 25 0A 5D
:          2B D4 09 A0 56 5B C5 B1 62 FC 20 BE
:          08 2D E3 07 B5 A1 E7 B3 FF C4 C0 A5
:          5F AC 12 5C A9
502 02    65:      INTEGER
:          00 B9 98 41 FC 08 50 1F 73 60 8A 01
:          A2 7C 52 8A 20 5A EA 2C 89 D9 A5 19
:          DD 94 C6 1B C3 25 C0 82 51 E4 EE 2B
:          9A 19 DC 73 ED E9 1D 27 D4 F8 6C 03
:          DD AB 1D 08 7B B5 AC 7F E9 82 9B F1
:          89 8A 71 DB 61
569 02    64:      INTEGER
:          01 07 21 97 5F 7A 60 A8 FD 5A 5C 07
:          DF A8 DE F7 E2 B1 34 7D FC EB 91 BD
:          B0 73 74 C8 C4 BE 3F 58 45 30 06 90
:          B3 AC 69 CC B3 F7 3F 7C AC C7 B8 1B
:          65 A1 16 39 39 B0 E3 74 7D CF CD C5
:          AC 6C BF E5
:          }
:      }
:  }
```

2.3. Certificates

```

AliceDSSSignByCarlNoInherit =
  0 30 732: SEQUENCE {
    4 30 667: SEQUENCE {
      8 A0 3: [0] {
        10 02 1: INTEGER 2
        :      }
        13 02 2: INTEGER 200
        17 30 9: SEQUENCE {
          19 06 7: OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
```

```

      :      (ANSI X9.57 algorithm)
      :      }
28 30 18: SEQUENCE {
30 31 16:   SET {
32 30 14:     SEQUENCE {
34 06 3:       OBJECT IDENTIFIER commonName (2 5 4 3)
      :       (X.520 id-at (2 5 4))
39 13 7:       PrintableString 'CarlDSS'
      :       }
      :     }
      :   }
      : SEQUENCE {
48 30 30:   UTCTime '990817011049Z'
50 17 13:   UTCTime '391231235959Z'
65 17 13:   }
      : SEQUENCE {
80 30 19:   SET {
82 31 17:     SEQUENCE {
84 30 15:       OBJECT IDENTIFIER commonName (2 5 4 3)
86 06 3:       (X.520 id-at (2 5 4))
91 13 8:       PrintableString 'AliceDSS'
      :       }
      :     }
      :   }
101 30 438: SEQUENCE {
105 30 299:   SEQUENCE {
109 06 7:     OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
      :     (ANSI X9.57 algorithm)
118 30 286:     SEQUENCE {
122 02 129:       INTEGER
      :       00 81 8D CD ED 83 EA 0A 9E 39 3E C2
      :       48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
      :       53 C5 AB 84 08 4F FF 94 E1 73 48 7E
      :       0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
      :       2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
      :       DC 5F 69 8A E4 75 D0 37 0C 91 08 95
      :       9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
      :       8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
      :       C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
      :       78 BD FF 9D B0 84 97 37 F2 E4 51 1B
      :       B5 E4 09 96 5C F3 7E 5B DB
254 02 21:       INTEGER
      :       00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
      :       B8 37 21 2B 62 8B F7 93 CD
277 02 128:       INTEGER
      :       26 38 D0 14 89 32 AA 39 FB 3E 6D D9
      :       4B 59 6A 4C 76 23 39 04 02 35 5C F2
      :       CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD

```

```

:          AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:          7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:          3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:          E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:          01 7C 6D 49 89 11 89 36 44 BD F8 C8
:          95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:          1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:          D1 81 4A 60 39 BA 36 39
:      }
:  }
408 03 132:  BIT STRING 0 unused bits, encapsulates {
412 02 128:  INTEGER
:          5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:          3F A9 EC AC 5E DC BD B7 13 11 34 A6
:          16 89 28 11 23 D9 34 86 67 75 75 13
:          12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:          1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:          A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:          7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:          08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:          F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:          32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:          F2 A5 E2 F4 F2 83 E5 B8
:      }
:  }
543 A3 129:  [3] {
546 30 127:  SEQUENCE {
548 30 12:  SEQUENCE {
550 06 3:  OBJECT IDENTIFIER
:          basicConstraints (2 5 29 19)
:          (X.509 id-ce (2 5 29))
555 01 1:  BOOLEAN TRUE
558 04 2:  OCTET STRING, encapsulates {
560 30 0:  SEQUENCE {}
:  }
:  }
562 30 14:  SEQUENCE {
564 06 3:  OBJECT IDENTIFIER keyUsage (2 5 29 15)
:          (X.509 id-ce (2 5 29))
569 01 1:  BOOLEAN TRUE
572 04 4:  OCTET STRING, encapsulates {
574 03 2:  BIT STRING 6 unused bits
:          '11'B
:  }
:  }
578 30 31:  SEQUENCE {
580 06 3:  OBJECT IDENTIFIER
:          authorityKeyIdentifier (2 5 29 35)

```

```

:      (X.509 id-ce (2 5 29))
585 04 24:      OCTET STRING, encapsulates {
587 30 22:      SEQUENCE {
589 80 20:      [0]
:      70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:      3D 20 BC 43 2B 93 F1 1F
:      }
:      }
:      }
611 30 29:      SEQUENCE {
613 06 3:      OBJECT IDENTIFIER
:      subjectKeyIdentifier (2 5 29 14)
:      (X.509 id-ce (2 5 29))
618 04 22:      OCTET STRING, encapsulates {
620 04 20:      OCTET STRING
:      BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
:      13 01 E2 FD E3 97 FE CD
:      }
:      }
642 30 31:      SEQUENCE {
644 06 3:      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
:      (X.509 id-ce (2 5 29))
649 04 24:      OCTET STRING, encapsulates {
651 30 22:      SEQUENCE {
653 81 20:      [1] 'AliceDSS@example.com'
:      }
:      }
:      }
:      }
:      }
675 30 9:      SEQUENCE {
677 06 7:      OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
:      (ANSI X9.57 algorithm)
:      }
686 03 48:      BIT STRING 0 unused bits, encapsulates {
689 30 45:      SEQUENCE {
691 02 20:      INTEGER
:      55 0C A4 19 1F 42 2B 89 71 22 33 8D
:      83 6A B5 3D 67 6B BF 45
713 02 21:      INTEGER
:      00 9F 61 53 52 54 0B 5C B2 DD DA E7
:      76 1D E2 10 52 5B 43 5E BD
:      }
:      }
:      }

```

AliceRSASignByCarl =


```

0 30 556: SEQUENCE {
4 30 405:   SEQUENCE {
8 A0 3:     [0] {
10 02 1:      INTEGER 2
      :      }
13 02 16:    INTEGER
      :      46 34 6B C7 80 00 56 BC 11 D3 6E 2E
      :      C4 10 B3 B0
31 30 13:    SEQUENCE {
33 06 9:      OBJECT IDENTIFIER
      :      sha1withRSAEncryption (1 2 840 113549 1 1 5)
      :      (PKCS #1)
44 05 0:      NULL
      :      }
46 30 18:    SEQUENCE {
48 31 16:      SET {
50 30 14:        SEQUENCE {
52 06 3:          OBJECT IDENTIFIER commonName (2 5 4 3)
      :          (X.520 id-at (2 5 4))
57 13 7:          PrintableString 'CarlRSA'
      :          }
      :      }
      :      }
66 30 30:    SEQUENCE {
68 17 13:      UTCTime '990919010847Z'
83 17 13:      UTCTime '391231235959Z'
      :      }
98 30 19:    SEQUENCE {
100 31 17:      SET {
102 30 15:        SEQUENCE {
104 06 3:          OBJECT IDENTIFIER commonName (2 5 4 3)
      :          (X.520 id-at (2 5 4))
109 13 8:          PrintableString 'AliceRSA'
      :          }
      :      }
      :      }
119 30 159:   SEQUENCE {
122 30 13:     SEQUENCE {
124 06 9:       OBJECT IDENTIFIER
      :       rsaEncryption (1 2 840 113549 1 1 1)
      :       (PKCS #1)
135 05 0:       NULL
      :       }
137 03 141:   BIT STRING 0 unused bits, encapsulates {
141 30 137:     SEQUENCE {
144 02 129:       INTEGER
      :       00 E0 89 73 39 8D D8 F5 F5 E8 87 76
      :       39 7F 4E B0 05 BB 53 83 DE 0F B7 AB

```

```

      :      DC 7D C7 75 29 0D 05 2E 6D 12 DF A6
      :      86 26 D4 D2 6F AA 58 29 FC 97 EC FA
      :      82 51 0F 30 80 BE B1 50 9E 46 44 F1
      :      2C BB D8 32 CF C6 68 6F 07 D9 B0 60
      :      AC BE EE 34 09 6A 13 F5 F7 05 05 93
      :      DF 5E BA 35 56 D9 61 FF 19 7F C9 81
      :      E6 F8 6C EA 87 40 70 EF AC 6D 2C 74
      :      9F 2D FA 55 3A B9 99 77 02 A6 48 52
      :      8C 4E F3 57 38 57 74 57 5F
276 02  3:      INTEGER 65537
      :      }
      :      }
      :      }
281 A3 129: [3] {
284 30 127:   SEQUENCE {
286 30 12:     SEQUENCE {
288 06  3:       OBJECT IDENTIFIER
      :         basicConstraints (2 5 29 19)
      :         (X.509 id-ce (2 5 29))
293 01  1:       BOOLEAN TRUE
296 04  2:       OCTET STRING, encapsulates {
298 30  0:         SEQUENCE {}
      :       }
      :     }
300 30 14:     SEQUENCE {
302 06  3:       OBJECT IDENTIFIER keyUsage (2 5 29 15)
      :       (X.509 id-ce (2 5 29))
307 01  1:       BOOLEAN TRUE
310 04  4:       OCTET STRING, encapsulates {
312 03  2:         BIT STRING 6 unused bits
      :         '11'B
      :       }
      :     }
316 30 31:     SEQUENCE {
318 06  3:       OBJECT IDENTIFIER
      :         authorityKeyIdentifier (2 5 29 35)
      :         (X.509 id-ce (2 5 29))
323 04 24:       OCTET STRING, encapsulates {
325 30 22:         SEQUENCE {
327 80 20:           [0]
      :             E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
      :             42 37 4E 22 AE 9E 38 BB
      :           }
      :         }
      :       }
349 30 29:     SEQUENCE {
351 06  3:       OBJECT IDENTIFIER
      :         subjectKeyIdentifier (2 5 29 14)

```

```

:      (X.509 id-ce (2 5 29))
356 04 22:      OCTET STRING, encapsulates {
358 04 20:      OCTET STRING
:      77 D2 B4 D1 B7 4C 8A 8A A3 CE 45 9D
:      CE EC 3C A0 3A E3 FF 50
:      }
:      }
380 30 31:      SEQUENCE {
382 06 3:      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
:      (X.509 id-ce (2 5 29))
387 04 24:      OCTET STRING, encapsulates {
389 30 22:      SEQUENCE {
391 81 20:      [1] 'AliceRSA@example.com'
:      }
:      }
:      }
:      }
:      }
413 30 13:      SEQUENCE {
415 06 9:      OBJECT IDENTIFIER
:      sha1withRSAEncryption (1 2 840 113549 1 1 5)
:      (PKCS #1)
426 05 0:      NULL
:      }
428 03 129:      BIT STRING 0 unused bits
:      3E 70 47 A8 48 CC 13 58 8F CA 51 71
:      6B 4E 36 18 5D 04 7E 80 B1 8D 4D CC
:      CA A3 8F CC 7D 56 C8 BC CF 6E B3 1C
:      59 A9 20 AA 05 81 A8 4E 25 AD A7 70
:      14 75 2F F5 C7 9B D1 0E E9 63 D2 64
:      B7 C6 66 6E 73 21 54 DF F4 BA 25 5D
:      7D 49 D3 94 6B 22 36 74 73 B8 4A EC
:      2F 64 ED D3 3D D2 A7 42 C5 E8 37 8A
:      B4 DB 9F 67 E4 BD 9F F9 FE 74 EF EA
:      F9 EE 63 6A D8 3F 4B 25 09 B5 D8 1A
:      76 AE EB 9B DB 49 B0 22
:      }

```

BobRSASignByCarl =

```

0 30 551: SEQUENCE {
4 30 400: SEQUENCE {
8 A0 3: [0] {
10 02 1: INTEGER 2
: }
13 02 16: INTEGER
: 46 34 6B C7 80 00 56 BC 11 D3 6E 2E
: CD 5D 71 D0

```

```

31 30 13: SEQUENCE {
33 06 9:   OBJECT IDENTIFIER
      :   sha1withRSAEncryption (1 2 840 113549 1 1 5)
      :   (PKCS #1)
44 05 0:   NULL
      :   }
46 30 18: SEQUENCE {
48 31 16:   SET {
50 30 14:     SEQUENCE {
52 06 3:       OBJECT IDENTIFIER commonName (2 5 4 3)
      :       (X.520 id-at (2 5 4))
57 13 7:       PrintableString 'CarlRSA'
      :       }
      :     }
      :   }
66 30 30: SEQUENCE {
68 17 13:   UTCTime '990919010902Z'
83 17 13:   UTCTime '391231235959Z'
      :   }
98 30 17: SEQUENCE {
100 31 15:   SET {
102 30 13:     SEQUENCE {
104 06 3:       OBJECT IDENTIFIER commonName (2 5 4 3)
      :       (X.520 id-at (2 5 4))
109 13 6:       PrintableString 'BobRSA'
      :       }
      :     }
      :   }
117 30 159: SEQUENCE {
120 30 13:   SEQUENCE {
122 06 9:     OBJECT IDENTIFIER
      :     rsaEncryption (1 2 840 113549 1 1 1)
      :     (PKCS #1)
133 05 0:     NULL
      :     }
135 03 141:   BIT STRING 0 unused bits, encapsulates {
139 30 137:     SEQUENCE {
142 02 129:       INTEGER
      :       00 A9 E1 67 98 3F 39 D5 5F F2 A0 93
      :       41 5E A6 79 89 85 C8 35 5D 9A 91 5B
      :       FB 1D 01 DA 19 70 26 17 0F BD A5 22
      :       D0 35 85 6D 7A 98 66 14 41 5C CF B7
      :       B7 08 3B 09 C9 91 B8 19 69 37 6D F9
      :       65 1E 7B D9 A9 33 24 A3 7F 3B BB AF
      :       46 01 86 36 34 32 CB 07 03 59 52 FC
      :       85 8B 31 04 B8 CC 18 08 14 48 E6 4F
      :       1C FB 5D 60 C4 E0 5C 1F 53 D3 7F 53
      :       D8 69 01 F1 05 F8 7A 70 D1 BE 83 C6

```

```

      :          5F 38 CF 1C 2C AA 6A A7 EB
274 02  3:      INTEGER 65537
      :      }
      :      }
      :      }
279 A3 127: [3] {
281 30 125:     SEQUENCE {
283 30 12:     SEQUENCE {
285 06  3:     OBJECT IDENTIFIER
      :         basicConstraints (2 5 29 19)
      :         (X.509 id-ce (2 5 29))
290 01  1:     BOOLEAN TRUE
293 04  2:     OCTET STRING, encapsulates {
295 30  0:     SEQUENCE {}
      :     }
      :     }
297 30 14:     SEQUENCE {
299 06  3:     OBJECT IDENTIFIER keyUsage (2 5 29 15)
      :         (X.509 id-ce (2 5 29))
304 01  1:     BOOLEAN TRUE
307 04  4:     OCTET STRING, encapsulates {
309 03  2:     BIT STRING 5 unused bits
      :         '100'B (bit 2)
      :     }
      :     }
313 30 31:     SEQUENCE {
315 06  3:     OBJECT IDENTIFIER
      :         authorityKeyIdentifier (2 5 29 35)
      :         (X.509 id-ce (2 5 29))
320 04 24:     OCTET STRING, encapsulates {
322 30 22:     SEQUENCE {
324 80 20:     [0]
      :         E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
      :         42 37 4E 22 AE 9E 38 BB
      :     }
      :     }
      :     }
346 30 29:     SEQUENCE {
348 06  3:     OBJECT IDENTIFIER
      :         subjectKeyIdentifier (2 5 29 14)
      :         (X.509 id-ce (2 5 29))
353 04 22:     OCTET STRING, encapsulates {
355 04 20:     OCTET STRING
      :         E8 F4 B8 67 D8 B3 96 A4 2A F3 11 AA
      :         29 D3 95 5A 86 16 B4 24
      :     }
      :     }
377 30 29:     SEQUENCE {

```

```

379 06      3:      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
      :      (X.509 id-ce (2 5 29))
384 04      22:      OCTET STRING, encapsulates {
386 30      20:      SEQUENCE {
388 81      18:      [1] 'BobRSA@example.com'
      :      }
      :      }
      :      }
      :      }
      :      }
      :      }
      :      }
408 30      13:      SEQUENCE {
410 06      9:      OBJECT IDENTIFIER
      :      sha1withRSAEncryption (1 2 840 113549 1 1 5)
      :      (PKCS #1)
421 05      0:      NULL
      :      }
423 03      129:     BIT STRING 0 unused bits
      :      7B 8E 66 C5 F1 10 3F 10 20 4C 88 71
      :      AB 7B 40 6B 21 33 FA 4A 95 DE 9D 0E
      :      5B 6B 94 21 05 C0 F2 E1 7E 2A CD 9C
      :      93 88 87 FB 8B B7 7E 7D 41 61 E1 E4
      :      D6 6D F9 E2 04 55 61 45 BC 64 27 44
      :      C0 A1 BD 59 79 D9 1D 64 3C 21 D6 45
      :      B0 5D 68 33 92 EA AC F1 57 E5 81 7D
      :      98 E6 35 91 A3 39 DE 77 F4 E8 1C 3B
      :      29 DC 7F 51 07 97 F3 36 F0 50 0A DD
      :      9B DE B6 5E 38 11 2B FB 57 EA 89 6D
      :      AD C9 88 D8 8F CF 2B D3
      :      }

CarlDSSSelf =
  0 30      667: SEQUENCE {
    4 30      602: SEQUENCE {
      8 A0      3: [0] {
        10 02      1: INTEGER 2
          :      }
        13 02      1: INTEGER 1
        16 30      9: SEQUENCE {
          18 06      7: OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
            :      (ANSI X9.57 algorithm)
            :      }
          27 30      18: SEQUENCE {
            29 31      16: SET {
              31 30      14: SEQUENCE {
                33 06      3: OBJECT IDENTIFIER commonName (2 5 4 3)
                  :      (X.520 id-at (2 5 4))
                38 13      7: PrintableString 'CarlDSS'

```

```

:      }
:    }
:  }
47 30 30: SEQUENCE {
49 17 13:   UTCTime '990816225050Z'
64 17 13:   UTCTime '391231235959Z'
:   }
79 30 18: SEQUENCE {
81 31 16:   SET {
83 30 14:     SEQUENCE {
85 06 3:      OBJECT IDENTIFIER commonName (2 5 4 3)
:          (X.520 id-at (2 5 4))
90 13 7:      PrintableString 'CarlDSS'
:      }
:    }
:  }
99 30 439: SEQUENCE {
103 30 299:   SEQUENCE {
107 06 7:     OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
:         (ANSI X9.57 algorithm)
116 30 286:     SEQUENCE {
120 02 129:       INTEGER
:         00 B6 49 18 3E 8A 44 C1 29 71 94 4C
:         01 C4 12 C1 7A 79 CB 54 4D AB 1E 81
:         FB C6 4C B3 0E 94 09 06 EB 01 D4 B1
:         C8 71 4B C7 45 C0 50 25 5D 9C FC DA
:         E4 6D D3 E2 86 48 84 82 7D BA 15 95
:         4A 16 F6 46 ED DD F6 98 D2 BB 7E 8A
:         0A 8A BA 16 7B B9 50 01 48 93 8B EB
:         25 15 51 97 55 DC 8F 53 0E 10 A9 50
:         FC 70 B7 CD 30 54 FD DA DE A8 AA 22
:         B5 A1 AF 8B CC 02 88 E7 8B 70 5F B9
:         AD E1 08 D4 6D 29 2D D6 E9
252 02 21:       INTEGER
:         00 DD C1 2F DF 53 CE 0B 34 60 77 3E
:         02 A4 BF 8A 5D 98 B9 10 D5
275 02 128:       INTEGER
:         0C EE 57 9B 4B BD DA B6 07 6A 74 37
:         4F 55 7F 9D ED BC 61 0D EB 46 59 3C
:         56 0B 2B 5B 0C 91 CE A5 62 52 69 CA
:         E1 6D 3E BD BF FE E1 B7 B9 2B 61 3C
:         AD CB AE 45 E3 06 AC 8C 22 9D 9C 44
:         87 0B C7 CD F0 1C D9 B5 4E 5D 73 DE
:         AF 0E C9 1D 5A 51 F5 4F 44 79 35 5A
:         73 AA 7F 46 51 1F A9 42 16 9C 48 EB
:         8A 79 61 B4 D5 2F 53 22 44 63 1F 86
:         B8 A3 58 06 25 F8 29 C0 EF BA E0 75
:         F0 42 C4 63 65 52 9B 0A

```

```

:      }
:    }
406 03 133:    BIT STRING 0 unused bits, encapsulates {
410 02 129:    INTEGER
:      00 99 87 74 27 03 66 A0 B1 C0 AD DC
:      2C 75 BB E1 6C 44 9C DA 21 6D 4D 47
:      6D B1 62 09 E9 D8 AE 1E F2 3A B4 94
:      B1 A3 8E 7A 9B 71 4E 00 94 C9 B4 25
:      4E B9 60 96 19 24 01 F3 62 0C FE 75
:      C0 FB CE D8 68 00 E3 FD D5 70 4F DF
:      23 96 19 06 94 F4 B1 61 8F 3A 57 B1
:      08 11 A4 0B 26 25 F0 52 76 81 EA 0B
:      62 0D 95 2A E6 86 BA 72 B2 A7 50 83
:      0B AA 27 CD 1B A9 4D 89 9A D7 8D 18
:      39 84 3F 8B C5 56 4D 80 7A
:    }
:  }
542 A3 66: [3] {
544 30 64:   SEQUENCE {
546 30 15:     SEQUENCE {
548 06 3:       OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
:         (X.509 id-ce (2 5 29))
553 01 1:       BOOLEAN TRUE
556 04 5:       OCTET STRING, encapsulates {
558 30 3:         SEQUENCE {
560 01 1:           BOOLEAN TRUE
:         }
:       }
:     }
563 30 14:   SEQUENCE {
565 06 3:     OBJECT IDENTIFIER keyUsage (2 5 29 15)
:     (X.509 id-ce (2 5 29))
570 01 1:     BOOLEAN TRUE
573 04 4:     OCTET STRING, encapsulates {
575 03 2:       BIT STRING 1 unused bits
:       '1100001'B
:     }
:   }
579 30 29: SEQUENCE {
581 06 3:   OBJECT IDENTIFIER
:   subjectKeyIdentifier (2 5 29 14)
:   (X.509 id-ce (2 5 29))
586 04 22:   OCTET STRING, encapsulates {
588 04 20:     OCTET STRING
:       70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:       3D 20 BC 43 2B 93 F1 1F
:     }
:   }

```



```

:      }
:    }
:  }
:
610 30    9: SEQUENCE {
612 06    7:   OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
:         (ANSI X9.57 algorithm)
:   }
621 03   48: BIT STRING 0 unused bits, encapsulates {
624 30   45:   SEQUENCE {
626 02   20:     INTEGER
:           6B A9 F0 4E 7A 5A 79 E3 F9 BE 3D 2B
:           C9 06 37 E9 11 17 A1 13
648 02   21:     INTEGER
:           00 8F 34 69 2A 8B B1 3C 03 79 94 32
:           4D 12 1F CE 89 FB 46 B2 3B
:         }
:   }
: }

CarlRSASelf =
  0 30  491: SEQUENCE {
    4 30  340:   SEQUENCE {
      8 A0   3:     [0] {
        10 02  1:       INTEGER 2
        :         }
        13 02  16:       INTEGER
        :         46 34 6B C7 80 00 56 BC 11 D3 6E 2E
        :         9F F2 50 20
        31 30  13:       SEQUENCE {
        33 06   9:         OBJECT IDENTIFIER
        :           sha1withRSAEncryption (1 2 840 113549 1 1 5)
        :           (PKCS #1)
        44 05   0:         NULL
        :         }
        46 30  18:       SEQUENCE {
        48 31  16:         SET {
        50 30  14:           SEQUENCE {
        52 06   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
        :             (X.520 id-at (2 5 4))
        57 13   7:             PrintableString 'CarlRSA'
        :           }
        :         }
        :       }
        66 30  30:       SEQUENCE {
        68 17  13:         UTCTime '990818070000Z'
        83 17  13:         UTCTime '391231235959Z'
        :       }

```

```

 98 30 18: SEQUENCE {
100 31 16:   SET {
102 30 14:     SEQUENCE {
104 06 3:       OBJECT IDENTIFIER commonName (2 5 4 3)
      :       (X.520 id-at (2 5 4))
109 13 7:       PrintableString 'CarlRSA'
      :     }
      :   }
      : }
118 30 159: SEQUENCE {
121 30 13:   SEQUENCE {
123 06 9:     OBJECT IDENTIFIER
      :     rsaEncryption (1 2 840 113549 1 1 1)
      :     (PKCS #1)
134 05 0:     NULL
      :   }
136 03 141:   BIT STRING 0 unused bits, encapsulates {
140 30 137:     SEQUENCE {
143 02 129:       INTEGER
      :       00 E4 4B FF 18 B8 24 57 F4 77 FF 6E
      :       73 7B 93 71 5C BC 33 1A 92 92 72 23
      :       D8 41 46 D0 CD 11 3A 04 B3 8E AF 82
      :       9D BD 51 1E 17 7A F2 76 2C 2B 86 39
      :       A7 BD D7 8D 1A 53 EC E4 00 D5 E8 EC
      :       A2 36 B1 ED E2 50 E2 32 09 8A 3F 9F
      :       99 25 8F B8 4E AB B9 7D D5 96 65 DA
      :       16 A0 C5 BE 0E AE 44 5B EF 5E F4 A7
      :       29 CB 82 DD AC 44 E9 AA 93 94 29 0E
      :       F8 18 D6 C8 57 5E F2 76 C4 F2 11 60
      :       38 B9 1B 3C 1D 97 C9 6A F1
275 02 3:     INTEGER 65537
      :   }
      : }
      : }
280 A3 66: [3] {
282 30 64:   SEQUENCE {
284 30 15:     SEQUENCE {
286 06 3:       OBJECT IDENTIFIER
      :       basicConstraints (2 5 29 19)
      :       (X.509 id-ce (2 5 29))
291 01 1:       BOOLEAN TRUE
294 04 5:       OCTET STRING, encapsulates {
296 30 3:         SEQUENCE {
298 01 1:           BOOLEAN TRUE
      :         }
      :       }
      :     }
      :   }
301 30 14: SEQUENCE {

```

```

303 06      3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
              :      (X.509 id-ce (2 5 29))
308 01      1:      BOOLEAN TRUE
311 04      4:      OCTET STRING, encapsulates {
313 03      2:      BIT STRING 1 unused bits
              :      '1100001'B
              :      }
              :      }
317 30      29:     SEQUENCE {
319 06      3:      OBJECT IDENTIFIER
              :      subjectKeyIdentifier (2 5 29 14)
              :      (X.509 id-ce (2 5 29))
324 04      22:     OCTET STRING, encapsulates {
326 04      20:     OCTET STRING
              :      E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
              :      42 37 4E 22 AE 9E 38 BB
              :      }
              :      }
              :      }
              :      }
              :      }
348 30      13:     SEQUENCE {
350 06      9:      OBJECT IDENTIFIER
              :      sha1withRSAEncryption (1 2 840 113549 1 1 5)
              :      (PKCS #1)
361 05      0:      NULL
              :      }
363 03      129:    BIT STRING 0 unused bits
              :      B7 9E D4 04 D3 ED 29 E4 FF 89 89 15
              :      2E 4C DB 0C F0 48 0F 32 61 EE C4 04
              :      EC 12 5D 2D FF 0F 64 59 7E 0A C3 ED
              :      18 FD E3 56 40 37 A7 07 B5 F0 38 12
              :      61 50 ED EF DD 3F E3 0B B8 61 A5 A4
              :      9B 3C E6 9E 9C 54 9A B6 95 D6 DA 6C
              :      3B B5 2D 45 35 9D 49 01 76 FA B9 B9
              :      31 F9 F9 6B 12 53 A0 F5 14 60 9B 7D
              :      CA 3E F2 53 6B B0 37 6F AD E6 74 D7
              :      DB FA 5A EA 14 41 63 5D CD BE C8 0E
              :      C1 DA 6A 8D 53 34 18 02
              :      }

```

```

DianeDSSSignByCarlInherit =
  0 30 440: SEQUENCE {
    4 30 375: SEQUENCE {
      8 A0 3: [0] {
        10 02 1: INTEGER 2
        :      }
        13 02 2: INTEGER 210

```

```

17 30    9:    SEQUENCE {
19 06    7:      OBJECT IDENTIFIER dsaWithShal (1 2 840 10040 4 3)
      :      (ANSI X9.57 algorithm)
      :    }
28 30   18:    SEQUENCE {
30 31   16:      SET {
32 30   14:        SEQUENCE {
34 06    3:          OBJECT IDENTIFIER commonName (2 5 4 3)
      :          (X.520 id-at (2 5 4))
39 13    7:          PrintableString 'CarlDSS'
      :        }
      :      }
      :    }
48 30   30:    SEQUENCE {
50 17   13:      UTCTime '990817020810Z'
65 17   13:      UTCTime '391231235959Z'
      :    }
80 30   19:    SEQUENCE {
82 31   17:      SET {
84 30   15:        SEQUENCE {
86 06    3:          OBJECT IDENTIFIER commonName (2 5 4 3)
      :          (X.520 id-at (2 5 4))
91 13    8:          PrintableString 'DianeDSS'
      :        }
      :      }
      :    }
101 30  147:    SEQUENCE {
104 30    9:      SEQUENCE {
106 06    7:        OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
      :        (ANSI X9.57 algorithm)
      :      }
115 03  133:    BIT STRING 0 unused bits, encapsulates {
119 02  129:      INTEGER
      :      00 A0 00 17 78 2C EE 7E 81 53 2E 2E
      :      61 08 0F A1 9B 51 52 1A DA 59 A8 73
      :      2F 12 25 B6 08 CB CA EF 2A 44 76 8A
      :      52 09 EA BD 05 22 D5 0F F6 FD 46 D7
      :      AF 99 38 09 0E 13 CB 4F 2C DD 1C 34
      :      F7 1C BF 25 FF 23 D3 3B 59 E7 82 97
      :      37 BE 31 24 D8 18 C8 F3 49 39 5B B7
      :      E2 E5 27 7E FC 8C 45 72 5B 7E 3E 8F
      :      68 4D DD 46 7A 22 BE 8E FF CC DA 39
      :      29 A3 39 E5 9F 43 E9 55 C9 D7 5B A6
      :      81 67 CC C0 AA CD 2E C5 23
      :    }
      :  }
251 A3  129:    [3] {
254 30  127:      SEQUENCE {

```

```

256 30 12:      SEQUENCE {
258 06 3:      OBJECT IDENTIFIER
:              basicConstraints (2 5 29 19)
:              (X.509 id-ce (2 5 29))
263 01 1:      BOOLEAN TRUE
266 04 2:      OCTET STRING, encapsulates {
268 30 0:      SEQUENCE {}
:      }
:      }
270 30 14:     SEQUENCE {
272 06 3:      OBJECT IDENTIFIER keyUsage (2 5 29 15)
:      (X.509 id-ce (2 5 29))
277 01 1:      BOOLEAN TRUE
280 04 4:      OCTET STRING, encapsulates {
282 03 2:      BIT STRING 6 unused bits
:      '11'B
:      }
:      }
286 30 31:     SEQUENCE {
288 06 3:      OBJECT IDENTIFIER
:      authorityKeyIdentifier (2 5 29 35)
:      (X.509 id-ce (2 5 29))
293 04 24:     OCTET STRING, encapsulates {
295 30 22:     SEQUENCE {
297 80 20:     [0]
:             70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:             3D 20 BC 43 2B 93 F1 1F
:             }
:      }
:      }
319 30 29:     SEQUENCE {
321 06 3:      OBJECT IDENTIFIER
:      subjectKeyIdentifier (2 5 29 14)
:      (X.509 id-ce (2 5 29))
326 04 22:     OCTET STRING, encapsulates {
328 04 20:     OCTET STRING
:             64 30 99 7D 5C DC 45 0B 99 3A 52 2F
:             16 BF 58 50 DD CE 2B 18
:             }
:      }
350 30 31:     SEQUENCE {
352 06 3:      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
:      (X.509 id-ce (2 5 29))
357 04 24:     OCTET STRING, encapsulates {
359 30 22:     SEQUENCE {
361 81 20:     [1] 'DianeDSS@example.com'
:      }
:      }

```

```

:      }
:    }
:  }
:
383 30    9: SEQUENCE {
385 06    7:   OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
:         (ANSI X9.57 algorithm)
:   }
394 03   48: BIT STRING 0 unused bits, encapsulates {
397 30   45:   SEQUENCE {
399 02   21:     INTEGER
:           00 A1 1A F8 17 0E 3E 5D A8 8C F4 B6
:           55 33 1E 4B E3 2C AC B9 5F
422 02   20:     INTEGER
:           28 4B 10 45 58 D2 1C 9D 55 35 14 18
:           91 B2 3F 39 DF B5 6E D3
:         }
:   }
: }

```

DianeRSASignByCarl =

```

 0 30 556: SEQUENCE {
 4 30 405:   SEQUENCE {
 8 A0   3:     [0] {
10 02   1:       INTEGER 2
:         }
13 02  16:       INTEGER
:         46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:         D5 9A 30 90
31 30  13:       SEQUENCE {
33 06   9:         OBJECT IDENTIFIER
:             sha1withRSAEncryption (1 2 840 113549 1 1 5)
:             (PKCS #1)
44 05   0:         NULL
:       }
46 30  18:       SEQUENCE {
48 31  16:         SET {
50 30  14:           SEQUENCE {
52 06   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
:               (X.520 id-at (2 5 4))
57 13   7:             PrintableString 'CarlRSA'
:           }
:         }
:       }
66 30  30:       SEQUENCE {
68 17  13:         UTCTime '990819070000Z'
83 17  13:         UTCTime '391231235959Z'
:       }

```

```

 98 30 19: SEQUENCE {
100 31 17:   SET {
102 30 15:     SEQUENCE {
104 06 3:       OBJECT IDENTIFIER commonName (2 5 4 3)
      :       (X.520 id-at (2 5 4))
109 13 8:       PrintableString 'DianerSA'
      :     }
      :   }
      : }
119 30 159: SEQUENCE {
122 30 13:   SEQUENCE {
124 06 9:     OBJECT IDENTIFIER
      :     rsaEncryption (1 2 840 113549 1 1 1)
      :     (PKCS #1)
135 05 0:     NULL
      :   }
137 03 141:   BIT STRING 0 unused bits, encapsulates {
141 30 137:     SEQUENCE {
144 02 129:       INTEGER
      :       00 D6 FD B8 C0 70 C6 4C 25 EC EA CF
      :       EA 7C BB A2 62 FA F0 E6 32 3A 53 FF
      :       B1 92 5A 17 F4 20 E1 99 24 82 0A D0
      :       F6 7C FB 44 CA 8B 27 06 F1 7E 26 03
      :       A9 76 9D CF EC A0 2C 70 96 F2 83 42
      :       F6 D4 B7 28 0A BB F8 BF 4A 4C 19 3F
      :       07 DB A0 C1 60 1E B7 7E 67 F7 DE B1
      :       C3 60 49 AC 45 D7 F8 C6 EF 08 37 21
      :       93 47 EE F0 73 35 72 B0 02 C4 F3 11
      :       C3 5E 47 E5 0A B7 83 F1 DB 74 69 64
      :       8B 44 1D 95 5D CD 28 C0 85
276 02 3:       INTEGER 65537
      :     }
      :   }
      : }
281 A3 129: [3] {
284 30 127:   SEQUENCE {
286 30 12:     SEQUENCE {
288 06 3:       OBJECT IDENTIFIER
      :       basicConstraints (2 5 29 19)
      :       (X.509 id-ce (2 5 29))
293 01 1:       BOOLEAN TRUE
296 04 2:       OCTET STRING, encapsulates {
298 30 0:         SEQUENCE {}
      :       }
      :     }
300 30 14:   SEQUENCE {
302 06 3:     OBJECT IDENTIFIER keyUsage (2 5 29 15)
      :     (X.509 id-ce (2 5 29))

```

```

307 01    1:      BOOLEAN TRUE
310 04    4:      OCTET STRING, encapsulates {
312 03    2:      BIT STRING 5 unused bits
           :      '111'B
           :      }
           :      }
316 30    31:     SEQUENCE {
318 06    3:      OBJECT IDENTIFIER
           :      authorityKeyIdentifier (2 5 29 35)
           :      (X.509 id-ce (2 5 29))
323 04    24:     OCTET STRING, encapsulates {
325 30    22:     SEQUENCE {
327 80    20:     [0]
           :      E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
           :      42 37 4E 22 AE 9E 38 BB
           :      }
           :      }
           :      }
349 30    29:     SEQUENCE {
351 06    3:      OBJECT IDENTIFIER
           :      subjectKeyIdentifier (2 5 29 14)
           :      (X.509 id-ce (2 5 29))
356 04    22:     OCTET STRING, encapsulates {
358 04    20:     OCTET STRING
           :      8C F3 CB 75 0E 8D 31 F6 D4 29 DA 44
           :      92 75 B8 FE ED 4F 39 0C
           :      }
           :      }
380 30    31:     SEQUENCE {
382 06    3:      OBJECT IDENTIFIER subjectAltName (2 5 29 17)
           :      (X.509 id-ce (2 5 29))
387 04    24:     OCTET STRING, encapsulates {
389 30    22:     SEQUENCE {
391 81    20:     [1] 'DianeRSA@example.com'
           :      }
           :      }
           :      }
           :      }
           :      }
413 30    13:     SEQUENCE {
415 06    9:      OBJECT IDENTIFIER
           :      sha1withRSAEncryption (1 2 840 113549 1 1 5)
           :      (PKCS #1)
426 05    0:      NULL
           :      }
428 03    129:    BIT STRING 0 unused bits
           :      7D A6 2C B5 78 42 D6 79 F3 31 FE F6

```



```

:      42 CA 0F 13 07 92 09 1B E0 6F B0 91
:      18 F6 BF 4A FB CC 63 79 FB 81 BF DD
:      97 C7 90 6B CB 0A 37 2B 41 6A 03 98
:      C5 1B 3E 32 C8 45 2B 86 01 9C 1C E2
:      36 EF 16 C1 1A 92 B8 BE 62 FB 53 3E
:      49 47 0B C4 B9 E4 2B 58 A6 06 83 F0
:      B2 A7 BB 85 7E D5 C6 DA CE 9C 7B 31
:      72 D7 A2 EA 41 AB 6A C0 DD 1F B9 14
:      44 18 CF 84 57 66 E8 C5 E6 B8 DC 2D
:      B3 1F 1B 28 43 36 75 7A
:    }

```

2.4. CRLs

```

CarlDSSCRLForAll =
  0 30 216: SEQUENCE {
    3 30 153: SEQUENCE {
      6 30 9: SEQUENCE {
        8 06 7: OBJECT IDENTIFIER dsaWithShal (1 2 840 10040 4 3)
          : (ANSI X9.57 algorithm)
          : }
      17 30 18: SEQUENCE {
        19 31 16: SET {
          21 30 14: SEQUENCE {
            23 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
              : (X.520 id-at (2 5 4))
            28 13 7: PrintableString 'CarlDSS'
              : }
            : }
          : }
        37 17 13: UTCTime '990827070000Z'
        52 30 105: SEQUENCE {
          54 30 19: SEQUENCE {
            56 02 2: INTEGER 200
            60 17 13: UTCTime '990822070000Z'
              : }
          75 30 19: SEQUENCE {
            77 02 2: INTEGER 201
            81 17 13: UTCTime '990822070000Z'
              : }
          96 30 19: SEQUENCE {
            98 02 2: INTEGER 211
            102 17 13: UTCTime '990822070000Z'
              : }
          117 30 19: SEQUENCE {
            119 02 2: INTEGER 210
            123 17 13: UTCTime '990822070000Z'
              : }

```

```

138 30 19: SEQUENCE {
140 02 2: INTEGER 212
144 17 13: UTCTime '990824070000Z'
      : }
      : }
      : }
159 30 9: SEQUENCE {
161 06 7: OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
      : (ANSI X9.57 algorithm)
      : }
170 03 47: BIT STRING 0 unused bits, encapsulates {
173 30 44: SEQUENCE {
175 02 20: INTEGER
      : 7E 65 52 76 33 FE 34 73 17 D1 F7 96
      : F9 A0 D4 D8 6D 5C 7D 3D
197 02 20: INTEGER
      : 02 7A 5B B7 D5 5B 18 C1 CF 87 EF 7E
      : DA 24 F3 2A 83 9C 35 A1
      : }
      : }
      : }

```

CarlDSSCRLForCarl =

```

0 30 131: SEQUENCE {
3 30 68: SEQUENCE {
5 30 9: SEQUENCE {
7 06 7: OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
      : (ANSI X9.57 algorithm)
      : }
16 30 18: SEQUENCE {
18 31 16: SET {
20 30 14: SEQUENCE {
22 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
      : (X.520 id-at (2 5 4))
27 13 7: PrintableString 'CarlDSS'
      : }
      : }
      : }
36 17 13: UTCTime '990825070000Z'
51 30 20: SEQUENCE {
53 30 18: SEQUENCE {
55 02 1: INTEGER 1
58 17 13: UTCTime '990822070000Z'
      : }
      : }
      : }
73 30 9: SEQUENCE {
75 06 7: OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)

```

```

      :      (ANSI X9.57 algorithm)
      :      }
84 03  48:  BIT STRING 0 unused bits, encapsulates {
87 30  45:      SEQUENCE {
89 02  21:          INTEGER
      :          00 B3 1F C5 4F 7A 3D EC 76 D5 60 F9
      :          DE 79 22 EC 4F B0 90 FE 97
112 02 20:          INTEGER
      :          5A 8B C3 84 BC 66 87 1B BF 79 82 5B
      :          0A 5D 07 F6 BA A9 05 29
      :      }
      :  }
      :  }

```

CarlDSSCRLEmpty =

```

 0 30 109: SEQUENCE {
 2 30  46:     SEQUENCE {
 4 30   9:         SEQUENCE {
 6 06   7:             OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
      :             (ANSI X9.57 algorithm)
      :         }
15 30 18:     SEQUENCE {
17 31 16:         SET {
19 30 14:             SEQUENCE {
21 06   3:                 OBJECT IDENTIFIER commonName (2 5 4 3)
      :                 (X.520 id-at (2 5 4))
26 13   7:                 PrintableString 'CarlDSS'
      :             }
      :         }
      :     }
35 17 13:     UTCTime '990820070000Z'
      : }
50 30   9: SEQUENCE {
52 06   7:     OBJECT IDENTIFIER dsaWithSha1 (1 2 840 10040 4 3)
      :     (ANSI X9.57 algorithm)
      : }
61 03  48:  BIT STRING 0 unused bits, encapsulates {
64 30  45:      SEQUENCE {
66 02  20:          INTEGER
      :          62 3F 36 17 31 58 2E 67 50 79 F5 09
      :          4B 8C AD D4 6B F4 64 9F
88 02  21:          INTEGER
      :          00 B5 3B 4E A1 4C 7B FD 0F C3 8D 9B
      :          B6 FE C3 5D 6F DE 65 28 7D
      :      }
      :  }
      :  }

```

```

CarlRSACRLForAll =
  0 30 307: SEQUENCE {
    4 30 157: SEQUENCE {
      7 30 13: SEQUENCE {
        9 06 9: OBJECT IDENTIFIER
          : md5withRSAEncryption (1 2 840 113549 1 1 4)
          : (PKCS #1)
      20 05 0: NULL
        :
      }
    22 30 18: SEQUENCE {
      24 31 16: SET {
        26 30 14: SEQUENCE {
          28 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
            : (X.520 id-at (2 5 4))
        33 13 7: PrintableString 'CarlRSA'
          :
          :
        }
      }
    42 17 13: UTCTime '990827070000Z'
    57 30 105: SEQUENCE {
      59 30 33: SEQUENCE {
        61 02 16: INTEGER
          : 46 34 6B C7 80 00 56 BC 11 D3 6E 2E
          : C4 10 B3 B0
      79 17 13: UTCTime '990822070000Z'
        :
        }
    94 30 33: SEQUENCE {
      96 02 16: INTEGER
        : 46 34 6B C7 80 00 56 BC 11 D3 6E 2E
        : D5 9A 30 90
    114 17 13: UTCTime '990822070000Z'
      :
      }
    129 30 33: SEQUENCE {
      131 02 16: INTEGER
        : 46 34 6B C7 80 00 56 BC 11 D3 6E 2E
        : CD 5D 71 D0
    149 17 13: UTCTime '990824070000Z'
      :
      }
    :
    :
  }
  164 30 13: SEQUENCE {
    166 06 9: OBJECT IDENTIFIER
      : md5withRSAEncryption (1 2 840 113549 1 1 4)
      : (PKCS #1)
    177 05 0: NULL
      :
    }
  179 03 129: BIT STRING 0 unused bits
    : BF B3 97 AA 53 F0 32 21 16 2B 77 92

```

```

:      7A 6B BB 97 C8 DC EA F1 FA 66 16 30
:      0E B5 9E 5C F0 81 D4 5E B3 6E C1 88
:      6B 8C D4 5E C5 4D FB 47 5E 66 F3 5D
:      AB E5 B4 18 36 60 A8 4D 9C 3C 89 EC
:      6F 27 BF 35 50 71 81 C2 B9 44 5B 62
:      89 19 12 31 A9 7B 9A D3 CC 66 CB 11
:      D9 0B 10 47 77 AD 4F 22 D9 E5 7F 30
:      F2 5B FC 94 51 A5 58 76 3B 1F A8 46
:      A6 1F F6 A1 DE 55 A1 ED 31 88 69 97
:      0F 08 D3 D4 0C 60 5B 1E
:      }

```

CarlRSACRLForCarl =

```

0 30 236: SEQUENCE {
3 30 87: SEQUENCE {
5 30 13: SEQUENCE {
7 06 9: OBJECT IDENTIFIER
:      md5withRSAEncryption (1 2 840 113549 1 1 4)
:      (PKCS #1)
18 05 0: NULL
:      }
20 30 18: SEQUENCE {
22 31 16: SET {
24 30 14: SEQUENCE {
26 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
:      (X.520 id-at (2 5 4))
31 13 7: PrintableString 'CarlRSA'
:      }
:      }
:      }
40 17 13: UTCTime '990825070000Z'
55 30 35: SEQUENCE {
57 30 33: SEQUENCE {
59 02 16: INTEGER
:      46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:      9F F2 50 20
77 17 13: UTCTime '990822070000Z'
:      }
:      }
:      }
92 30 13: SEQUENCE {
94 06 9: OBJECT IDENTIFIER
:      md5withRSAEncryption (1 2 840 113549 1 1 4)
:      (PKCS #1)
105 05 0: NULL
:      }
107 03 129: BIT STRING 0 unused bits
:      21 EF 21 D4 C1 1A 85 95 49 6B CA 45

```

```

: 62 DC D7 09 FF A9 51 2E 8E D9 47 18
: FA F8 E5 72 DD 4F ED 74 74 E3 F3 65
: 32 65 28 2C 9A 1D 57 E5 D5 26 06 EA
: D5 E6 23 95 84 8D 0E 89 9E EE 9B 0C
: 2F CE 07 F7 A3 D1 6B 85 4C 0F FF E6
: DD FC DC CD 73 2C 1E 7D DC B0 71 C5
: 4C FC 01 6E 52 57 69 1E 39 63 DF 12
: 22 30 C7 13 55 94 05 6E 2A 00 A9 5B
: C4 2A 66 94 62 CE 36 33 C2 2B 63 47
: 25 9D F3 DE 70 EE 00 56
: }

```

CarlRSACRLEmpty =

```

0 30 199: SEQUENCE {
3 30 50: SEQUENCE {
5 30 13: SEQUENCE {
7 06 9: OBJECT IDENTIFIER
: md5withRSAEncryption (1 2 840 113549 1 1 4)
: (PKCS #1)
18 05 0: NULL
: }
20 30 18: SEQUENCE {
22 31 16: SET {
24 30 14: SEQUENCE {
26 06 3: OBJECT IDENTIFIER commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
31 13 7: PrintableString 'CarlRSA'
: }
: }
: }
40 17 13: UTCTime '990820070000Z'
: }
55 30 13: SEQUENCE {
57 06 9: OBJECT IDENTIFIER
: md5withRSAEncryption (1 2 840 113549 1 1 4)
: (PKCS #1)
68 05 0: NULL
: }
70 03 129: BIT STRING 0 unused bits
: A9 C5 21 B8 13 7C 74 F3 B5 11 EC 04
: F3 20 45 86 1E 0B 6E 7F 83 6D 5F F4
: 34 76 06 59 25 0E 04 3D 88 09 88 81
: 37 C4 DC 20 98 FA 17 81 0B 37 94 AC
: B4 8F 7B 51 89 14 A4 CB 72 73 14 07
: BC 22 9C 40 A1 07 FC 44 7C 85 0F 0B
: 88 D1 EE E1 0E AF F6 16 74 AD A1 AF
: C1 00 75 00 64 EA A5 9A F6 0B 08 A2
: DB 95 19 5F A6 A7 B9 39 45 25 0A 0E

```

```

:      F6 5E 84 E7 F8 B9 5A C9 18 C2 0E B8
:      A0 96 BE 81 3A 80 6D C9
:      }

```

3. Trivial Examples

This section covers examples of small CMS types.

3.1. ContentInfo with Data Type, BER

The object is a ContentInfo containing a Data object in BER format that is ExContent.

```

0 30 NDEF: SEQUENCE {
2 06 9: OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:      (PKCS #7)
13 A0 NDEF: [0] {
15 24 NDEF: OCTET STRING {
17 04 4: OCTET STRING 'This'
23 04 24: OCTET STRING ' is some sample content.'
:      }
:      }
:      }

```

3.2. ContentInfo with Data Type, DER

The object is a ContentInfo containing a Data object in DER format that is ExContent.

```

0 30 43: SEQUENCE {
2 06 9: OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:      (PKCS #7)
13 A0 30: [0] {
15 04 28: OCTET STRING 'This is some sample content.'
:      }
:      }

```

4. Signed-data

4.1. Basic Signed Content, DSS

A SignedData with no attribute certificates, signed by Alice using DSS, just her certificate (not Carl's root cert), no CRL. The message is ExContent, and is included in the eContent. There are no signed or unsigned attributes.

```

0 30 919: SEQUENCE {
4 06 9:  OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
      :  (PKCS #7)
15 A0 904:  [0] {
19 30 900:    SEQUENCE {
23 02 1:      INTEGER 1
26 31 9:      SET {
28 30 7:        SEQUENCE {
30 06 5:          OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
      :          (OIW)
      :        }
      :      }
37 30 43:    SEQUENCE {
39 06 9:      OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
      :      (PKCS #7)
50 A0 30:    [0] {
52 04 28:      OCTET STRING 'This is some sample content.'
      :    }
      :  }
82 A0 736:  [0] {
86 30 732:    SEQUENCE {
90 30 667:      SEQUENCE {
94 A0 3:        [0] {
96 02 1:          INTEGER 2
      :        }
99 02 2:          INTEGER 200
103 30 9:          SEQUENCE {
105 06 7:            OBJECT IDENTIFIER
      :            dsaWithSha1 (1 2 840 10040 4 3)
      :            (ANSI X9.57 algorithm)
      :          }
114 30 18:          SEQUENCE {
116 31 16:            SET {
118 30 14:              SEQUENCE {
120 06 3:                OBJECT IDENTIFIER
      :                commonName (2 5 4 3)
      :                (X.520 id-at (2 5 4))
125 13 7:                PrintableString 'CarlDSS'
      :              }
      :            }
      :          }
134 30 30:          SEQUENCE {
136 17 13:            UTCTime '990817011049Z'
151 17 13:            UTCTime '391231235959Z'
      :          }
166 30 19:          SEQUENCE {
168 31 17:            SET {
170 30 15:              SEQUENCE {

```



```

172 06      3:      OBJECT IDENTIFIER
                :      commonName (2 5 4 3)
                :      (X.520 id-at (2 5 4))
177 13      8:      PrintableString 'AliceDSS'
                :      }
                :      }
                :      }
187 30      438:    SEQUENCE {
191 30      299:    SEQUENCE {
195 06      7:      OBJECT IDENTIFIER
                :      dsa (1 2 840 10040 4 1)
                :      (ANSI X9.57 algorithm)
204 30      286:    SEQUENCE {
208 02      129:    INTEGER
                :      00 81 8D CD ED 83 EA 0A 9E 39 3E C2
                :      48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
                :      53 C5 AB 84 08 4F FF 94 E1 73 48 7E
                :      0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
                :      2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
                :      DC 5F 69 8A E4 75 D0 37 0C 91 08 95
                :      9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
                :      8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
                :      C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
                :      78 BD FF 9D B0 84 97 37 F2 E4 51 1B
                :      B5 E4 09 96 5C F3 7E 5B DB
340 02      21:    INTEGER
                :      00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
                :      B8 37 21 2B 62 8B F7 93 CD
363 02      128:    INTEGER
                :      26 38 D0 14 89 32 AA 39 FB 3E 6D D9
                :      4B 59 6A 4C 76 23 39 04 02 35 5C F2
                :      CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
                :      AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
                :      7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
                :      3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
                :      E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
                :      01 7C 6D 49 89 11 89 36 44 BD F8 C8
                :      95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
                :      1F 11 7F C2 BD ED D1 50 FF 98 74 C2
                :      D1 81 4A 60 39 BA 36 39
                :      }
                :      }
494 03      132:    BIT STRING 0 unused bits, encapsulates {
498 02      128:    INTEGER
                :      5C E3 B9 5A 75 14 96 0B A9 7A DD E3
                :      3F A9 EC AC 5E DC BD B7 13 11 34 A6
                :      16 89 28 11 23 D9 34 86 67 75 75 13
                :      12 3D 43 5B 6F E5 51 BF FA 89 F2 A2

```

```

:          1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:          A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:          7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:          08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:          F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:          32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:          F2 A5 E2 F4 F2 83 E5 B8
:          }
:      }
629 A3 129: [3] {
632 30 127: SEQUENCE {
634 30 12: SEQUENCE {
636 06 3: OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
:         (X.509 id-ce (2 5 29))
641 01 1: BOOLEAN TRUE
644 04 2: OCTET STRING, encapsulates {
646 30 0: SEQUENCE {}
:
:     }
648 30 14: SEQUENCE {
650 06 3: OBJECT IDENTIFIER
:         keyUsage (2 5 29 15)
:         (X.509 id-ce (2 5 29))
655 01 1: BOOLEAN TRUE
658 04 4: OCTET STRING, encapsulates {
660 03 2: BIT STRING 6 unused bits
:         '11'B
:         }
:     }
664 30 31: SEQUENCE {
666 06 3: OBJECT IDENTIFIER
:         authorityKeyIdentifier (2 5 29 35)
:         (X.509 id-ce (2 5 29))
671 04 24: OCTET STRING, encapsulates {
673 30 22: SEQUENCE {
675 80 20: [0]
:         70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:         3D 20 BC 43 2B 93 F1 1F
:         }
:     }
697 30 29: SEQUENCE {
699 06 3: OBJECT IDENTIFIER
:         subjectKeyIdentifier (2 5 29 14)
:         (X.509 id-ce (2 5 29))
704 04 22: OCTET STRING, encapsulates {
706 04 20: OCTET STRING

```

```

:          BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
:          13 01 E2 FD E3 97 FE CD
:          }
:      }
728 30 31: SEQUENCE {
730 06 3:   OBJECT IDENTIFIER
:       subjectAltName (2 5 29 17)
:       (X.509 id-ce (2 5 29))
735 04 24:   OCTET STRING, encapsulates {
737 30 22:       SEQUENCE {
739 81 20:         [1] 'AliceDSS@example.com'
:         }
:       }
:     }
:   }
: }
761 30 9: SEQUENCE {
763 06 7:   OBJECT IDENTIFIER
:       dsaWithShal (1 2 840 10040 4 3)
:       (ANSI X9.57 algorithm)
:     }
772 03 48:   BIT STRING 0 unused bits, encapsulates {
775 30 45:       SEQUENCE {
777 02 20:         INTEGER
:         55 0C A4 19 1F 42 2B 89 71 22 33 8D
:         83 6A B5 3D 67 6B BF 45
799 02 21:         INTEGER
:         00 9F 61 53 52 54 0B 5C B2 DD DA E7
:         76 1D E2 10 52 5B 43 5E BD
:       }
:     }
:   }
: }
822 31 99: SET {
824 30 97:   SEQUENCE {
826 02 1:     INTEGER 1
829 30 24:   SEQUENCE {
831 30 18:     SEQUENCE {
833 31 16:       SET {
835 30 14:         SEQUENCE {
837 06 3:           OBJECT IDENTIFIER
:           commonName (2 5 4 3)
:           (X.520 id-at (2 5 4))
842 13 7:           PrintableString 'CarlDSS'
:         }
:       }
:     }
:   }

```

```

851 02      2:          INTEGER 200
           :          }
855 30      7:          SEQUENCE {
857 06      5:              OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
           :              (OIW)
           :          }
864 30      9:          SEQUENCE {
866 06      7:              OBJECT IDENTIFIER
           :                  dsaWithSha1 (1 2 840 10040 4 3)
           :                  (ANSI X9.57 algorithm)
           :              }
875 04     46:          OCTET STRING, encapsulates {
877 30     44:              SEQUENCE {
879 02     20:                  INTEGER
           :                      09 91 FE EB D2 69 F5 18 B7 D7 CD 55
           :                      F4 81 EA 2A 42 6A AD 03
901 02     20:                  INTEGER
           :                      3A 07 CC C3 21 BE E1 1A 4B 7F 3E B5
           :                      0D DB BA 1C EA BC CD 89
           :                  }
           :              }
           :          }
           :      }
           :  }
           : }

```

4.2. Basic Signed Content, RSA

Same as 4.1, except using RSA signatures. A SignedData with no attribute certificates, signed by Alice using RSA, just her certificate (not Carl's root cert), no CRL. The message is ExContent, and is included in the eContent. There are no signed or unsigned attributes.

```

0 30 850: SEQUENCE {
4 06 9: OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
: (PKCS #7)
15 A0 835: [0] {
19 30 831: SEQUENCE {
23 02 1: INTEGER 1
26 31 11: SET {
28 30 9: SEQUENCE {
30 06 5: OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
: (OIW)
37 05 0: NULL
: }
: }

```

```

39 30 43: SEQUENCE {
41 06 9:   OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:       (PKCS #7)
52 A0 30:   [0] {
54 04 28:     OCTET STRING 'This is some sample content.'
:       }
:   }
84 A0 560: [0] {
88 30 556:   SEQUENCE {
92 30 405:     SEQUENCE {
96 A0 3:       [0] {
98 02 1:         INTEGER 2
:       }
101 02 16:     INTEGER
:         46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:         C4 10 B3 B0
119 30 13:     SEQUENCE {
121 06 9:       OBJECT IDENTIFIER
:         sha1withRSAEncryption
:         (1 2 840 113549 1 1 5)
:         (PKCS #1)
132 05 0:       NULL
:     }
134 30 18:     SEQUENCE {
136 31 16:       SET {
138 30 14:         SEQUENCE {
140 06 3:           OBJECT IDENTIFIER
:             commonName (2 5 4 3)
:             (X.520 id-at (2 5 4))
145 13 7:           PrintableString 'CarlRSA'
:         }
:       }
:     }
154 30 30:     SEQUENCE {
156 17 13:       UTCTime '990919010847Z'
171 17 13:       UTCTime '391231235959Z'
:     }
186 30 19:     SEQUENCE {
188 31 17:       SET {
190 30 15:         SEQUENCE {
192 06 3:           OBJECT IDENTIFIER
:             commonName (2 5 4 3)
:             (X.520 id-at (2 5 4))
197 13 8:           PrintableString 'AlicerSA'
:         }
:       }
:     }
207 30 159: SEQUENCE {

```

```

210 30    13:      SEQUENCE {
212 06     9:      OBJECT IDENTIFIER
                :      rsaEncryption (1 2 840 113549 1 1 1)
                :      (PKCS #1)
223 05     0:      NULL
                :      }
225 03   141:      BIT STRING 0 unused bits, encapsulates {
229 30   137:      SEQUENCE {
232 02   129:      INTEGER
                :      00 E0 89 73 39 8D D8 F5 F5 E8 87 76
                :      39 7F 4E B0 05 BB 53 83 DE 0F B7 AB
                :      DC 7D C7 75 29 0D 05 2E 6D 12 DF A6
                :      86 26 D4 D2 6F AA 58 29 FC 97 EC FA
                :      82 51 0F 30 80 BE B1 50 9E 46 44 F1
                :      2C BB D8 32 CF C6 68 6F 07 D9 B0 60
                :      AC BE EE 34 09 6A 13 F5 F7 05 05 93
                :      DF 5E BA 35 56 D9 61 FF 19 7F C9 81
                :      E6 F8 6C EA 87 40 70 EF AC 6D 2C 74
                :      9F 2D FA 55 3A B9 99 77 02 A6 48 52
                :      8C 4E F3 57 38 57 74 57 5F
364 02     3:      INTEGER 65537
                :      }
                :      }
                :      }
369 A3   129:      [3] {
372 30   127:      SEQUENCE {
374 30   12:      SEQUENCE {
376 06     3:      OBJECT IDENTIFIER
                :      basicConstraints (2 5 29 19)
                :      (X.509 id-ce (2 5 29))
381 01     1:      BOOLEAN TRUE
384 04     2:      OCTET STRING, encapsulates {
386 30     0:      SEQUENCE {}
                :      }
                :      }
388 30   14:      SEQUENCE {
390 06     3:      OBJECT IDENTIFIER
                :      keyUsage (2 5 29 15)
                :      (X.509 id-ce (2 5 29))
395 01     1:      BOOLEAN TRUE
398 04     4:      OCTET STRING, encapsulates {
400 03     2:      BIT STRING 6 unused bits
                :      '11'B
                :      }
                :      }
404 30   31:      SEQUENCE {
406 06     3:      OBJECT IDENTIFIER
                :      authorityKeyIdentifier (2 5 29 35)

```

```

:                (X.509 id-ce (2 5 29))
411 04 24:      OCTET STRING, encapsulates {
413 30 22:      SEQUENCE {
415 80 20:      [0]
:                E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
:                42 37 4E 22 AE 9E 38 BB
:                }
:            }
:        }
437 30 29:      SEQUENCE {
439 06 3:        OBJECT IDENTIFIER
:                subjectKeyIdentifier (2 5 29 14)
:                (X.509 id-ce (2 5 29))
444 04 22:      OCTET STRING, encapsulates {
446 04 20:      OCTET STRING
:                77 D2 B4 D1 B7 4C 8A 8A A3 CE 45 9D
:                CE EC 3C A0 3A E3 FF 50
:                }
:            }
468 30 31:      SEQUENCE {
470 06 3:        OBJECT IDENTIFIER
:                subjectAltName (2 5 29 17)
:                (X.509 id-ce (2 5 29))
475 04 24:      OCTET STRING, encapsulates {
477 30 22:      SEQUENCE {
479 81 20:      [1] 'AliceRSA@example.com'
:                }
:            }
:        }
:    }
:    }
501 30 13:      SEQUENCE {
503 06 9:        OBJECT IDENTIFIER
:                sha1withRSAEncryption
:                (1 2 840 113549 1 1 5)
:                (PKCS #1)
514 05 0:        NULL
:            }
516 03 129:      BIT STRING 0 unused bits
:                3E 70 47 A8 48 CC 13 58 8F CA 51 71
:                6B 4E 36 18 5D 04 7E 80 B1 8D 4D CC
:                CA A3 8F CC 7D 56 C8 BC CF 6E B3 1C
:                59 A9 20 AA 05 81 A8 4E 25 AD A7 70
:                14 75 2F F5 C7 9B D1 0E E9 63 D2 64
:                B7 C6 66 6E 73 21 54 DF F4 BA 25 5D
:                7D 49 D3 94 6B 22 36 74 73 B8 4A EC
:                2F 64 ED D3 3D D2 A7 42 C5 E8 37 8A

```

```

:          B4 DB 9F 67 E4 BD 9F F9 FE 74 EF EA
:          F9 EE 63 6A D8 3F 4B 25 09 B5 D8 1A
:          76 AE EB 9B DB 49 B0 22
:      }
:  }
648 31 203: SET {
651 30 200:   SEQUENCE {
654 02 1:     INTEGER 1
657 30 38:     SEQUENCE {
659 30 18:       SEQUENCE {
661 31 16:         SET {
663 30 14:           SEQUENCE {
665 06 3:             OBJECT IDENTIFIER
:               commonName (2 5 4 3)
:               (X.520 id-at (2 5 4))
670 13 7:             PrintableString 'CarlRSA'
:           }
:         }
:       }
679 02 16:     INTEGER
:       46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:       C4 10 B3 B0
:     }
697 30 9:   SEQUENCE {
699 06 5:     OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:       (OIW)
706 05 0:     NULL
:   }
708 30 13: SEQUENCE {
710 06 9:   OBJECT IDENTIFIER
:     rsaEncryption (1 2 840 113549 1 1 1)
:     (PKCS #1)
721 05 0:   NULL
: }
723 04 128: OCTET STRING
:   2F 23 82 D2 F3 09 5F B8 0C 58 EB 4E
:   9D BF 89 9A 81 E5 75 C4 91 3D D3 D0
:   D5 7B B6 D5 FE 94 A1 8A AC E3 C4 84
:   F5 CD 60 4E 27 95 F6 CF 00 86 76 75
:   3F 2B F0 E7 D4 02 67 A7 F5 C7 8D 16
:   04 A5 B3 B5 E7 D9 32 F0 24 EF E7 20
:   44 D5 9F 07 C5 53 24 FA CE 01 1D 0F
:   17 13 A7 2A 95 9D 2B E4 03 95 14 0B
:   E9 39 0D BA CE 6E 9C 9E 0C E8 98 E6
:   55 13 D4 68 6F D0 07 D7 A2 B1 62 4C
:   E3 8F AF FD E0 D5 5D C7
: }
: }

```



```

:      }
:    }
:  }

```

4.3. Basic Signed Content, Detached Content

Same as 4.1, except with no eContent. A SignedData with no attribute certificates, signed by Alice using DSS, just her certificate (not Carl's root cert), no CRL. The message is ExContent, but the eContent is not included. There are no signed or unsigned attributes.

```

0 30 887: SEQUENCE {
4 06 9:  OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
:      (PKCS #7)
15 A0 872:  [0] {
19 30 868:    SEQUENCE {
23 02 1:      INTEGER 1
26 31 9:      SET {
28 30 7:        SEQUENCE {
30 06 5:          OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:          (OIW)
:        }
:      }
:    }
37 30 11:    SEQUENCE {
39 06 9:      OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:      (PKCS #7)
:    }
50 A0 736:  [0] {
54 30 732:    SEQUENCE {
58 30 667:      SEQUENCE {
62 A0 3:        [0] {
64 02 1:          INTEGER 2
:        }
67 02 2:          INTEGER 200
71 30 9:          SEQUENCE {
73 06 7:            OBJECT IDENTIFIER
:            dsaWithShal (1 2 840 10040 4 3)
:            (ANSI X9.57 algorithm)
:          }
82 30 18:        SEQUENCE {
84 31 16:          SET {
86 30 14:            SEQUENCE {
88 06 3:              OBJECT IDENTIFIER
:              commonName (2 5 4 3)
:              (X.520 id-at (2 5 4))
93 13 7:              PrintableString 'CarlDSS'
:            }

```

```

:      }
:    }
102 30 30: SEQUENCE {
104 17 13:   UTCTime '990817011049Z'
119 17 13:   UTCTime '391231235959Z'
:     }
134 30 19: SEQUENCE {
136 31 17:   SET {
138 30 15:     SEQUENCE {
140 06 3:      OBJECT IDENTIFIER
:         commonName (2 5 4 3)
:         (X.520 id-at (2 5 4))
145 13 8:      PrintableString 'AliceDSS'
:     }
:   }
: }
155 30 438: SEQUENCE {
159 30 299:   SEQUENCE {
163 06 7:     OBJECT IDENTIFIER
:         dsa (1 2 840 10040 4 1)
:         (ANSI X9.57 algorithm)
172 30 286:     SEQUENCE {
176 02 129:       INTEGER
:         00 81 8D CD ED 83 EA 0A 9E 39 3E C2
:         48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
:         53 C5 AB 84 08 4F FF 94 E1 73 48 7E
:         0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
:         2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
:         DC 5F 69 8A E4 75 D0 37 0C 91 08 95
:         9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
:         8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
:         C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
:         78 BD FF 9D B0 84 97 37 F2 E4 51 1B
:         B5 E4 09 96 5C F3 7E 5B DB
308 02 21:       INTEGER
:         00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:         B8 37 21 2B 62 8B F7 93 CD
331 02 128:       INTEGER
:         26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:         4B 59 6A 4C 76 23 39 04 02 35 5C F2
:         CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:         AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:         7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:         3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:         E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:         01 7C 6D 49 89 11 89 36 44 BD F8 C8
:         95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:         1F 11 7F C2 BD ED D1 50 FF 98 74 C2

```

```

:           D1 81 4A 60 39 BA 36 39
:           }
:         }
462 03 132: BIT STRING 0 unused bits, encapsulates {
466 02 128: INTEGER
:         5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:         3F A9 EC AC 5E DC BD B7 13 11 34 A6
:         16 89 28 11 23 D9 34 86 67 75 75 13
:         12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:         1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:         A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:         7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:         08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:         F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:         32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:         F2 A5 E2 F4 F2 83 E5 B8
:         }
:       }
597 A3 129: [3] {
600 30 127: SEQUENCE {
602 30 12: SEQUENCE {
604 06 3: OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
:         (X.509 id-ce (2 5 29))
609 01 1: BOOLEAN TRUE
612 04 2: OCTET STRING, encapsulates {
614 30 0: SEQUENCE {}
:       }
:     }
616 30 14: SEQUENCE {
618 06 3: OBJECT IDENTIFIER
:         keyUsage (2 5 29 15)
:         (X.509 id-ce (2 5 29))
623 01 1: BOOLEAN TRUE
626 04 4: OCTET STRING, encapsulates {
628 03 2: BIT STRING 6 unused bits
:         '11'B
:       }
:     }
632 30 31: SEQUENCE {
634 06 3: OBJECT IDENTIFIER
:         authorityKeyIdentifier (2 5 29 35)
:         (X.509 id-ce (2 5 29))
639 04 24: OCTET STRING, encapsulates {
641 30 22: SEQUENCE {
643 80 20: [0]
:         70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:         3D 20 BC 43 2B 93 F1 1F

```

```

:           }
:         }
:       }
665 30 29: SEQUENCE {
667 06 3:   OBJECT IDENTIFIER
:         subjectKeyIdentifier (2 5 29 14)
:         (X.509 id-ce (2 5 29))
672 04 22:   OCTET STRING, encapsulates {
674 04 20:     OCTET STRING
:         BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
:         13 01 E2 FD E3 97 FE CD
:       }
:     }
696 30 31: SEQUENCE {
698 06 3:   OBJECT IDENTIFIER
:         subjectAltName (2 5 29 17)
:         (X.509 id-ce (2 5 29))
703 04 24:   OCTET STRING, encapsulates {
705 30 22:     SEQUENCE {
707 81 20:       [1] 'AliceDSS@example.com'
:     }
:   }
: }
: }
: }
729 30 9: SEQUENCE {
731 06 7:   OBJECT IDENTIFIER
:         dsaWithShal (1 2 840 10040 4 3)
:         (ANSI X9.57 algorithm)
:   }
740 03 48: BIT STRING 0 unused bits, encapsulates {
743 30 45:   SEQUENCE {
745 02 20:     INTEGER
:         55 0C A4 19 1F 42 2B 89 71 22 33 8D
:         83 6A B5 3D 67 6B BF 45
767 02 21:     INTEGER
:         00 9F 61 53 52 54 0B 5C B2 DD DA E7
:         76 1D E2 10 52 5B 43 5E BD
:   }
: }
: }
: }
790 31 99: SET {
792 30 97:   SEQUENCE {
794 02 1:     INTEGER 1
797 30 24:   SEQUENCE {
799 30 18:     SEQUENCE {

```

```

801 31    16:          SET {
803 30    14:              SEQUENCE {
805 06     3:                  OBJECT IDENTIFIER
                        :                      commonName (2 5 4 3)
                        :                      (X.520 id-at (2 5 4))
810 13     7:                  PrintableString 'CarlDSS'
                        :                      }
                        :              }
                        :          }
819 02     2:              INTEGER 200
                        :          }
823 30     7:              SEQUENCE {
825 06     5:                  OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
                        :                      (OIW)
                        :                  }
832 30     9:              SEQUENCE {
834 06     7:                  OBJECT IDENTIFIER
                        :                      dsaWithSha1 (1 2 840 10040 4 3)
                        :                      (ANSI X9.57 algorithm)
                        :                  }
843 04    46:              OCTET STRING, encapsulates {
845 30    44:                  SEQUENCE {
847 02    20:                      INTEGER
                        :                          06 FB C7 2A 24 D5 34 89 F7 8B B5 FD
                        :                          73 24 A5 86 C8 0F 5A 6C
869 02    20:                      INTEGER
                        :                          66 69 19 BC 68 58 D1 8D B1 9D 52 3F
                        :                          DA 14 88 0D FD C9 A1 B8
                        :                      }
                        :                  }
                        :              }
                        :          }
                        :      }
                        :  }
                        :  }

```

4.4. Fancier Signed Content

Same as 4.1, but includes Carl's root cert, Carl's CRL, some signed and unsigned attributes (Countersignature by Diane). A SignedData with no attribute certificates, signed by Alice using DSS, her certificate and Carl's root cert, Carl's DSS CRL. The message is ExContent, and is included in the eContent. The signed attributes are Content Type, Message Digest and Signing Time; the unsigned attributes are content hint and counter signature. The message includes also Alice's RSA certificate.

```

0 30 2829: SEQUENCE {
4 06 9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
      :   (PKCS #7)
15 A0 2814: [0] {
19 30 2810:   SEQUENCE {
23 02 1:     INTEGER 1
26 31 9:     SET {
28 30 7:       SEQUENCE {
30 06 5:         OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
      :         (OIW)
      :       }
      :     }
37 30 43:   SEQUENCE {
39 06 9:     OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
      :     (PKCS #7)
50 A0 30:   [0] {
52 04 28:     OCTET STRING 'This is some sample content.'
      :   }
      : }
82 A0 1967: [0] {
86 30 556:   SEQUENCE {
90 30 405:     SEQUENCE {
94 A0 3:       [0] {
96 02 1:         INTEGER 2
      :       }
99 02 16:     INTEGER
      :         46 34 6B C7 80 00 56 BC 11 D3 6E 2E
      :         C4 10 B3 B0
117 30 13:   SEQUENCE {
119 06 9:     OBJECT IDENTIFIER
      :       sha1withRSAEncryption
      :       (1 2 840 113549 1 1 5)
      :       (PKCS #1)
130 05 0:     NULL
      :   }
132 30 18:   SEQUENCE {
134 31 16:     SET {
136 30 14:       SEQUENCE {
138 06 3:         OBJECT IDENTIFIER
      :         commonName (2 5 4 3)
      :         (X.520 id-at (2 5 4))
143 13 7:         PrintableString 'CarlRSA'
      :       }
      :     }
      :   }
152 30 30:   SEQUENCE {
154 17 13:     UTCTime '990919010847Z'
169 17 13:     UTCTime '391231235959Z'

```

```

:
184 30 19: }
186 31 17: SEQUENCE {
188 30 15: SET {
190 06 3: SEQUENCE {
: OBJECT IDENTIFIER
: commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
195 13 8: PrintableString 'AlicerSA'
: }
: }
: }
205 30 159: SEQUENCE {
208 30 13: SEQUENCE {
210 06 9: OBJECT IDENTIFIER
: rsaEncryption (1 2 840 113549 1 1 1)
: (PKCS #1)
221 05 0: NULL
: }
223 03 141: BIT STRING 0 unused bits, encapsulates {
227 30 137: SEQUENCE {
230 02 129: INTEGER
: 00 E0 89 73 39 8D D8 F5 F5 E8 87 76
: 39 7F 4E B0 05 BB 53 83 DE 0F B7 AB
: DC 7D C7 75 29 0D 05 2E 6D 12 DF A6
: 86 26 D4 D2 6F AA 58 29 FC 97 EC FA
: 82 51 0F 30 80 BE B1 50 9E 46 44 F1
: 2C BB D8 32 CF C6 68 6F 07 D9 B0 60
: AC BE EE 34 09 6A 13 F5 F7 05 05 93
: DF 5E BA 35 56 D9 61 FF 19 7F C9 81
: E6 F8 6C EA 87 40 70 EF AC 6D 2C 74
: 9F 2D FA 55 3A B9 99 77 02 A6 48 52
: 8C 4E F3 57 38 57 74 57 5F
362 02 3: INTEGER 65537
: }
: }
: }
367 A3 129: [3] {
370 30 127: SEQUENCE {
372 30 12: SEQUENCE {
374 06 3: OBJECT IDENTIFIER
: basicConstraints (2 5 29 19)
: (X.509 id-ce (2 5 29))
379 01 1: BOOLEAN TRUE
382 04 2: OCTET STRING, encapsulates {
384 30 0: SEQUENCE {}
: }
: }
386 30 14: SEQUENCE {

```

```

388 06      3:      OBJECT IDENTIFIER
                :      keyUsage (2 5 29 15)
                :      (X.509 id-ce (2 5 29))
393 01      1:      BOOLEAN TRUE
396 04      4:      OCTET STRING, encapsulates {
398 03      2:      BIT STRING 6 unused bits
                :      '11'B
                :      }
                :      }
402 30      31:     SEQUENCE {
404 06      3:      OBJECT IDENTIFIER
                :      authorityKeyIdentifier (2 5 29 35)
                :      (X.509 id-ce (2 5 29))
409 04      24:     OCTET STRING, encapsulates {
411 30      22:     SEQUENCE {
413 80      20:     [0]
                :      E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
                :      42 37 4E 22 AE 9E 38 BB
                :      }
                :      }
                :      }
435 30      29:     SEQUENCE {
437 06      3:      OBJECT IDENTIFIER
                :      subjectKeyIdentifier (2 5 29 14)
                :      (X.509 id-ce (2 5 29))
442 04      22:     OCTET STRING, encapsulates {
444 04      20:     OCTET STRING
                :      77 D2 B4 D1 B7 4C 8A 8A A3 CE 45 9D
                :      CE EC 3C A0 3A E3 FF 50
                :      }
                :      }
466 30      31:     SEQUENCE {
468 06      3:      OBJECT IDENTIFIER
                :      subjectAltName (2 5 29 17)
                :      (X.509 id-ce (2 5 29))
473 04      24:     OCTET STRING, encapsulates {
475 30      22:     SEQUENCE {
477 81      20:     [1] 'AliceRSA@example.com'
                :      }
                :      }
                :      }
                :      }
                :      }
                :      }
499 30      13:     SEQUENCE {
501 06      9:      OBJECT IDENTIFIER
                :      sha1withRSAEncryption
                :      (1 2 840 113549 1 1 5)

```



```

:          (PKCS #1)
512 05    0:          NULL
:          }
514 03    129:        BIT STRING 0 unused bits
:          3E 70 47 A8 48 CC 13 58 8F CA 51 71
:          6B 4E 36 18 5D 04 7E 80 B1 8D 4D CC
:          CA A3 8F CC 7D 56 C8 BC CF 6E B3 1C
:          59 A9 20 AA 05 81 A8 4E 25 AD A7 70
:          14 75 2F F5 C7 9B D1 0E E9 63 D2 64
:          B7 C6 66 6E 73 21 54 DF F4 BA 25 5D
:          7D 49 D3 94 6B 22 36 74 73 B8 4A EC
:          2F 64 ED D3 3D D2 A7 42 C5 E8 37 8A
:          B4 DB 9F 67 E4 BD 9F F9 FE 74 EF EA
:          F9 EE 63 6A D8 3F 4B 25 09 B5 D8 1A
:          76 AE EB 9B DB 49 B0 22
:          }
646 30    667:        SEQUENCE {
650 30    602:          SEQUENCE {
654 A0     3:            [0] {
656 02     1:              INTEGER 2
:              }
659 02     1:            INTEGER 1
662 30     9:          SEQUENCE {
664 06     7:            OBJECT IDENTIFIER
:              dsaWithShal (1 2 840 10040 4 3)
:              (ANSI X9.57 algorithm)
:            }
673 30    18:          SEQUENCE {
675 31    16:            SET {
677 30    14:              SEQUENCE {
679 06     3:                OBJECT IDENTIFIER
:                commonName (2 5 4 3)
:                (X.520 id-at (2 5 4))
684 13     7:                PrintableString 'CarlDSS'
:              }
:            }
:          }
693 30    30:          SEQUENCE {
695 17    13:            UTCTime '990816225050Z'
710 17    13:            UTCTime '391231235959Z'
:          }
725 30    18:          SEQUENCE {
727 31    16:            SET {
729 30    14:              SEQUENCE {
731 06     3:                OBJECT IDENTIFIER
:                commonName (2 5 4 3)
:                (X.520 id-at (2 5 4))
736 13     7:                PrintableString 'CarlDSS'

```

```

:      }
:    }
:  }
745 30 439: SEQUENCE {
749 30 299: SEQUENCE {
753 06 7:   OBJECT IDENTIFIER
:         dsa (1 2 840 10040 4 1)
:         (ANSI X9.57 algorithm)
762 30 286: SEQUENCE {
766 02 129:   INTEGER
:         00 B6 49 18 3E 8A 44 C1 29 71 94 4C
:         01 C4 12 C1 7A 79 CB 54 4D AB 1E 81
:         FB C6 4C B3 0E 94 09 06 EB 01 D4 B1
:         C8 71 4B C7 45 C0 50 25 5D 9C FC DA
:         E4 6D D3 E2 86 48 84 82 7D BA 15 95
:         4A 16 F6 46 ED DD F6 98 D2 BB 7E 8A
:         0A 8A BA 16 7B B9 50 01 48 93 8B EB
:         25 15 51 97 55 DC 8F 53 0E 10 A9 50
:         FC 70 B7 CD 30 54 FD DA DE A8 AA 22
:         B5 A1 AF 8B CC 02 88 E7 8B 70 5F B9
:         AD E1 08 D4 6D 29 2D D6 E9
898 02 21:   INTEGER
:         00 DD C1 2F DF 53 CE 0B 34 60 77 3E
:         02 A4 BF 8A 5D 98 B9 10 D5
921 02 128:   INTEGER
:         0C EE 57 9B 4B BD DA B6 07 6A 74 37
:         4F 55 7F 9D ED BC 61 0D EB 46 59 3C
:         56 0B 2B 5B 0C 91 CE A5 62 52 69 CA
:         E1 6D 3E BD BF FE E1 B7 B9 2B 61 3C
:         AD CB AE 45 E3 06 AC 8C 22 9D 9C 44
:         87 0B C7 CD F0 1C D9 B5 4E 5D 73 DE
:         AF 0E C9 1D 5A 51 F5 4F 44 79 35 5A
:         73 AA 7F 46 51 1F A9 42 16 9C 48 EB
:         8A 79 61 B4 D5 2F 53 22 44 63 1F 86
:         B8 A3 58 06 25 F8 29 C0 EF BA E0 75
:         F0 42 C4 63 65 52 9B 0A
:       }
:     }
1052 03 133: BIT STRING 0 unused bits, encapsulates {
1056 02 129:   INTEGER
:         00 99 87 74 27 03 66 A0 B1 C0 AD DC
:         2C 75 BB E1 6C 44 9C DA 21 6D 4D 47
:         6D B1 62 09 E9 D8 AE 1E F2 3A B4 94
:         B1 A3 8E 7A 9B 71 4E 00 94 C9 B4 25
:         4E B9 60 96 19 24 01 F3 62 0C FE 75
:         C0 FB CE D8 68 00 E3 FD D5 70 4F DF
:         23 96 19 06 94 F4 B1 61 8F 3A 57 B1
:         08 11 A4 0B 26 25 F0 52 76 81 EA 0B

```

```

:          62 0D 95 2A E6 86 BA 72 B2 A7 50 83
:          0B AA 27 CD 1B A9 4D 89 9A D7 8D 18
:          39 84 3F 8B C5 56 4D 80 7A
:          }
:      }
1188 A3 66:      [3] {
1190 30 64:      SEQUENCE {
1192 30 15:      SEQUENCE {
1194 06 3:      OBJECT IDENTIFIER
:      basicConstraints (2 5 29 19)
:      (X.509 id-ce (2 5 29))
1199 01 1:      BOOLEAN TRUE
1202 04 5:      OCTET STRING, encapsulates {
1204 30 3:      SEQUENCE {
1206 01 1:      BOOLEAN TRUE
:      }
:      }
:      }
1209 30 14:      SEQUENCE {
1211 06 3:      OBJECT IDENTIFIER
:      keyUsage (2 5 29 15)
:      (X.509 id-ce (2 5 29))
1216 01 1:      BOOLEAN TRUE
1219 04 4:      OCTET STRING, encapsulates {
1221 03 2:      BIT STRING 1 unused bits
:      '1100001'B
:      }
:      }
1225 30 29:      SEQUENCE {
1227 06 3:      OBJECT IDENTIFIER
:      subjectKeyIdentifier (2 5 29 14)
:      (X.509 id-ce (2 5 29))
1232 04 22:      OCTET STRING, encapsulates {
1234 04 20:      OCTET STRING
:      70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:      3D 20 BC 43 2B 93 F1 1F
:      }
:      }
:      }
:      }
:      }
1256 30 9:      SEQUENCE {
1258 06 7:      OBJECT IDENTIFIER
:      dsaWithSha1 (1 2 840 10040 4 3)
:      (ANSI X9.57 algorithm)
:      }
1267 03 48:      BIT STRING 0 unused bits, encapsulates {
1270 30 45:      SEQUENCE {

```

```

1272 02 20:      INTEGER
      :          6B A9 F0 4E 7A 5A 79 E3 F9 BE 3D 2B
      :          C9 06 37 E9 11 17 A1 13
1294 02 21:      INTEGER
      :          00 8F 34 69 2A 8B B1 3C 03 79 94 32
      :          4D 12 1F CE 89 FB 46 B2 3B
      :      }
      :  }
      :  }
1317 30 732:      SEQUENCE {
1321 30 667:          SEQUENCE {
1325 A0 3:              [0] {
1327 02 1:                  INTEGER 2
      :              }
1330 02 2:          INTEGER 200
1334 30 9:          SEQUENCE {
1336 06 7:              OBJECT IDENTIFIER
      :                  dsaWithShal (1 2 840 10040 4 3)
      :                  (ANSI X9.57 algorithm)
      :              }
1345 30 18:          SEQUENCE {
1347 31 16:              SET {
1349 30 14:                  SEQUENCE {
1351 06 3:                      OBJECT IDENTIFIER
      :                          commonName (2 5 4 3)
      :                          (X.520 id-at (2 5 4))
1356 13 7:                      PrintableString 'CarlDSS'
      :                      }
      :              }
      :          }
1365 30 30:          SEQUENCE {
1367 17 13:              UTCTime '990817011049Z'
1382 17 13:              UTCTime '391231235959Z'
      :          }
1397 30 19:          SEQUENCE {
1399 31 17:              SET {
1401 30 15:                  SEQUENCE {
1403 06 3:                      OBJECT IDENTIFIER
      :                          commonName (2 5 4 3)
      :                          (X.520 id-at (2 5 4))
1408 13 8:                      PrintableString 'AliceDSS'
      :                      }
      :              }
      :          }
1418 30 438:          SEQUENCE {
1422 30 299:              SEQUENCE {
1426 06 7:                  OBJECT IDENTIFIER
      :                      dsa (1 2 840 10040 4 1)

```

```

:                               (ANSI X9.57 algorithm)
1435 30 286: SEQUENCE {
1439 02 129:   INTEGER
:           00 81 8D CD ED 83 EA 0A 9E 39 3E C2
:           48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
:           53 C5 AB 84 08 4F FF 94 E1 73 48 7E
:           0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
:           2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
:           DC 5F 69 8A E4 75 D0 37 0C 91 08 95
:           9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
:           8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
:           C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
:           78 BD FF 9D B0 84 97 37 F2 E4 51 1B
:           B5 E4 09 96 5C F3 7E 5B DB
1571 02 21:   INTEGER
:           00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:           B8 37 21 2B 62 8B F7 93 CD
1594 02 128:   INTEGER
:           26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:           4B 59 6A 4C 76 23 39 04 02 35 5C F2
:           CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:           AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:           7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:           3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:           E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:           01 7C 6D 49 89 11 89 36 44 BD F8 C8
:           95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:           1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:           D1 81 4A 60 39 BA 36 39
:           }
:   }
1725 03 132: BIT STRING 0 unused bits, encapsulates {
1729 02 128:   INTEGER
:           5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:           3F A9 EC AC 5E DC BD B7 13 11 34 A6
:           16 89 28 11 23 D9 34 86 67 75 75 13
:           12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:           1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:           A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:           7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:           08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:           F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:           32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:           F2 A5 E2 F4 F2 83 E5 B8
:           }
:   }
1860 A3 129: [3] {
1863 30 127:   SEQUENCE {

```

```

1865 30 12: SEQUENCE {
1867 06 3:   OBJECT IDENTIFIER
      :      basicConstraints (2 5 29 19)
      :      (X.509 id-ce (2 5 29))
1872 01 1:   BOOLEAN TRUE
1875 04 2:   OCTET STRING, encapsulates {
1877 30 0:     SEQUENCE {}
      :   }
      : }
1879 30 14: SEQUENCE {
1881 06 3:   OBJECT IDENTIFIER
      :      keyUsage (2 5 29 15)
      :      (X.509 id-ce (2 5 29))
1886 01 1:   BOOLEAN TRUE
1889 04 4:   OCTET STRING, encapsulates {
1891 03 2:     BIT STRING 6 unused bits
      :       '11'B
      :   }
      : }
1895 30 31: SEQUENCE {
1897 06 3:   OBJECT IDENTIFIER
      :      authorityKeyIdentifier (2 5 29 35)
      :      (X.509 id-ce (2 5 29))
1902 04 24:   OCTET STRING, encapsulates {
1904 30 22:     SEQUENCE {
1906 80 20:       [0]
      :       70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
      :       3D 20 BC 43 2B 93 F1 1F
      :     }
      : }
1928 30 29: SEQUENCE {
1930 06 3:   OBJECT IDENTIFIER
      :      subjectKeyIdentifier (2 5 29 14)
      :      (X.509 id-ce (2 5 29))
1935 04 22:   OCTET STRING, encapsulates {
1937 04 20:     OCTET STRING
      :       BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
      :       13 01 E2 FD E3 97 FE CD
      :     }
      : }
1959 30 31: SEQUENCE {
1961 06 3:   OBJECT IDENTIFIER
      :      subjectAltName (2 5 29 17)
      :      (X.509 id-ce (2 5 29))
1966 04 24:   OCTET STRING, encapsulates {
1968 30 22:     SEQUENCE {
1970 81 20:       [1] 'AliceDSS@example.com'

```

```

:
:
:
:
:
:
:
1992 30 9: SEQUENCE {
1994 06 7: OBJECT IDENTIFIER
: dsaWithShal (1 2 840 10040 4 3)
: (ANSI X9.57 algorithm)
:
:
2003 03 48: BIT STRING 0 unused bits, encapsulates {
2006 30 45: SEQUENCE {
2008 02 20: INTEGER
: 55 0C A4 19 1F 42 2B 89 71 22 33 8D
: 83 6A B5 3D 67 6B BF 45
2030 02 21: INTEGER
: 00 9F 61 53 52 54 0B 5C B2 DD DA E7
: 76 1D E2 10 52 5B 43 5E BD
:
:
:
:
:
:
:
:
:
:
2053 A1 219: [1] {
2056 30 216: SEQUENCE {
2059 30 153: SEQUENCE {
2062 30 9: SEQUENCE {
2064 06 7: OBJECT IDENTIFIER
: dsaWithShal (1 2 840 10040 4 3)
: (ANSI X9.57 algorithm)
:
:
:
2073 30 18: SEQUENCE {
2075 31 16: SET {
2077 30 14: SEQUENCE {
2079 06 3: OBJECT IDENTIFIER
: commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
2084 13 7: PrintableString 'CarlDSS'
:
:
:
:
:
2093 17 13: UTCTime '990827070000Z'
2108 30 105: SEQUENCE {
2110 30 19: SEQUENCE {
2112 02 2: INTEGER 200
2116 17 13: UTCTime '990822070000Z'
:
:
2131 30 19: SEQUENCE {

```

```

2133 02      2:      INTEGER 201
2137 17      13:     UTCTime '990822070000Z'
                :
2152 30      19:     SEQUENCE {
2154 02      2:      INTEGER 211
2158 17      13:     UTCTime '990822070000Z'
                :
2173 30      19:     SEQUENCE {
2175 02      2:      INTEGER 210
2179 17      13:     UTCTime '990822070000Z'
                :
2194 30      19:     SEQUENCE {
2196 02      2:      INTEGER 212
2200 17      13:     UTCTime '990824070000Z'
                :
                :
                :
                }
                }
2215 30      9:     SEQUENCE {
2217 06      7:     OBJECT IDENTIFIER
                :      dsaWithShal (1 2 840 10040 4 3)
                :      (ANSI X9.57 algorithm)
                :
                }
2226 03      47:     BIT STRING 0 unused bits, encapsulates {
2229 30      44:     SEQUENCE {
2231 02      20:     INTEGER
                :      7E 65 52 76 33 FE 34 73 17 D1 F7 96
                :      F9 A0 D4 D8 6D 5C 7D 3D
2253 02      20:     INTEGER
                :      02 7A 5B B7 D5 5B 18 C1 CF 87 EF 7E
                :      DA 24 F3 2A 83 9C 35 A1
                :
                :
                :
                }
                }
                :
                :
                }
                }
2275 31      554:    SET {
2279 30      550:    SEQUENCE {
2283 02      1:      INTEGER 1
2286 30      24:     SEQUENCE {
2288 30      18:     SEQUENCE {
2290 31      16:     SET {
2292 30      14:     SEQUENCE {
2294 06      3:      OBJECT IDENTIFIER
                :      commonName (2 5 4 3)
                :      (X.520 id-at (2 5 4))
2299 13      7:      PrintableString 'CarlDSS'
                :
                :
                :
                }
                }
                }

```



```

2308 02    2:    INTEGER 200
           :
           :    }
2312 30    7:    SEQUENCE {
2314 06    5:    OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
           :    (OIW)
           :    }
2321 A0    93:   [0] {
2323 30    24:   SEQUENCE {
2325 06    9:    OBJECT IDENTIFIER
           :    contentType (1 2 840 113549 1 9 3)
           :    (PKCS #9 (1 2 840 113549 1 9))
2336 31    11:   SET {
2338 06    9:    OBJECT IDENTIFIER
           :    data (1 2 840 113549 1 7 1)
           :    (PKCS #7)
           :    }
           :    }
           :    }
2349 30    28:   SEQUENCE {
2351 06    9:    OBJECT IDENTIFIER
           :    signingTime (1 2 840 113549 1 9 5)
           :    (PKCS #9 (1 2 840 113549 1 9))
2362 31    15:   SET {
2364 17    13:   UTCTime '030514153900Z'
           :    }
           :    }
2379 30    35:   SEQUENCE {
2381 06    9:    OBJECT IDENTIFIER
           :    messageDigest (1 2 840 113549 1 9 4)
           :    (PKCS #9 (1 2 840 113549 1 9))
2392 31    22:   SET {
2394 04    20:   OCTET STRING
           :    40 6A EC 08 52 79 BA 6E 16 02 2D 9E
           :    06 29 C0 22 96 87 DD 48
           :    }
           :    }
           :    }
2416 30    9:    SEQUENCE {
2418 06    7:    OBJECT IDENTIFIER
           :    dsaWithSha1 (1 2 840 10040 4 3)
           :    (ANSI X9.57 algorithm)
           :    }
2427 04    46:   OCTET STRING, encapsulates {
2429 30    44:   SEQUENCE {
2431 02    20:   INTEGER
           :    3B A5 E0 4A DB 6D 58 E0 19 D1 00 1C
           :    4F 44 9A 57 7A 71 66 68
2453 02    20:   INTEGER
           :    1A 11 98 D6 1F 1F AF 34 81 01 DE BE

```

```

      :      8B DC B6 A8 6A 91 69 13
      :      }
      :    }
2475 A1 354: [1] {
2479 30 62:   SEQUENCE {
2481 06 11:     OBJECT IDENTIFIER
      :     id-aa-contentHint
      :     (1 2 840 113549 1 9 16 2 4)
      :     (S/MIME Authenticated Attributes
      :     (1 2 840 113549 1 9 16 2))
2494 31 47:   SET {
2496 30 45:     SEQUENCE {
2498 0C 32:       UTF8String
      :       'Content Hints Description Buffer'
2532 06 9:     OBJECT IDENTIFIER
      :     data (1 2 840 113549 1 7 1)
      :     (PKCS #7)
      :   }
      : }
      : }
2543 30 286: SEQUENCE {
2547 06 9:   OBJECT IDENTIFIER
      :   countersignature (1 2 840 113549 1 9 6)
      :   (PKCS #9 (1 2 840 113549 1 9))
2558 31 271: SET {
2562 30 267:   SEQUENCE {
2566 02 1:     INTEGER 1
2569 30 38:     SEQUENCE {
2571 30 18:       SEQUENCE {
2573 31 16:         SET {
2575 30 14:           SEQUENCE {
2577 06 3:             OBJECT IDENTIFIER
      :             commonName (2 5 4 3)
      :             (X.520 id-at (2 5 4))
2582 13 7:             PrintableString 'CarlRSA'
      :           }
      :         }
      :       }
      :     }
2591 02 16:     INTEGER
      :     46 34 6B C7 80 00 56 BC 11 D3 6E 2E
      :     C4 10 B3 B0
      :   }
2609 30 7:   SEQUENCE {
2611 06 5:     OBJECT IDENTIFIER
      :     sha1 (1 3 14 3 2 26)
      :     (OIW)
      :   }
2618 A0 67: [0] {

```

```

2620 30      28:          SEQUENCE {
2622 06       9:              OBJECT IDENTIFIER
                :                  signingTime
                :                      (1 2 840 113549 1 9 5)
                :                      (PKCS #9 (1 2 840 113549 1 9))
2633 31     15:          SET {
2635 17     13:              UTCTime '030514153900Z'
                :                  }
                :            }
2650 30     35:          SEQUENCE {
2652 06       9:              OBJECT IDENTIFIER
                :                  messageDigest
                :                      (1 2 840 113549 1 9 4)
                :                      (PKCS #9 (1 2 840 113549 1 9))
2663 31     22:          SET {
2665 04     20:              OCTET STRING
                :                  02 5F 49 4E 39 98 50 85 B3 66 D3 8A
                :                  1F 7B 9E 69 AA FB D8 33
                :                  }
                :            }
2687 30     13:          SEQUENCE {
2689 06       9:              OBJECT IDENTIFIER
                :                  rsaEncryption
                :                      (1 2 840 113549 1 1 1)
                :                      (PKCS #1)
2700 05       0:              NULL
                :            }
2702 04    128:          OCTET STRING
                :                  6D AA 20 24 ED 7A EE A5 5E 87 DD 75
                :                  1F 2B 54 10 65 F4 CE 9B B1 2C 78 74
                :                  BC 8B 1C 60 B5 DB 8B 03 9E 49 F2 2B
                :                  7F 93 6E 3D 89 14 C9 E3 6B F4 F6 7D
                :                  76 AE 3E 58 1F 9B BB BC 7C 30 19 4E
                :                  10 F7 02 F1 8B 5B B4 DB 9A BB 93 B4
                :                  18 D0 CC 2B C9 91 A9 AD D9 46 F8 65
                :                  A9 E2 71 95 D0 D4 4E 1F CD 74 6F 82
                :                  E8 37 6F 5A 3D CB C7 D4 5F C2 80 1B
                :                  DA D3 84 40 68 5F 56 9A 62 F5 3B 0D
                :                  6C 33 C3 ED 67 3F 43 BF
                :                  }
                :            }
                :          }
                :        }
                :      }
                :    }
                :   }
                :  }

```

```
: }
```

4.5. All RSA Signed Message

Same as 4.2, but includes Carl's RSA root cert (but no CRL). A SignedData with no attribute certificates, signed by Alice using RSA, her certificate and Carl's root cert, no CRL. The message is ExContent, and is included in the eContent. There are no signed or unsigned attributes.

```

0 30 NDEF: SEQUENCE {
2 06 9: OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
: (PKCS #7)
13 A0 NDEF: [0] {
15 30 NDEF: SEQUENCE {
17 02 1: INTEGER 1
20 31 11: SET {
22 30 9: SEQUENCE {
24 06 5: OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
: (OIW)
31 05 0: NULL
: }
: }
33 30 NDEF: SEQUENCE {
35 06 9: OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
: (PKCS #7)
46 A0 NDEF: [0] {
48 24 NDEF: OCTET STRING {
50 04 4: OCTET STRING 'This'
56 04 24: OCTET STRING ' is some sample content.'
: }
: }
: }
88 A0 NDEF: [0] {
90 30 491: SEQUENCE {
94 30 340: SEQUENCE {
98 A0 3: [0] {
100 02 1: INTEGER 2
: }
103 02 16: INTEGER
: 46 34 6B C7 80 00 56 BC 11 D3 6E 2E
: 9F F2 50 20
121 30 13: SEQUENCE {
123 06 9: OBJECT IDENTIFIER
: sha1withRSAEncryption
: (1 2 840 113549 1 1 5)
: (PKCS #1)
134 05 0: NULL

```

```

:
136 30 18: }
138 31 16: SEQUENCE {
140 30 14: SET {
142 06 3: SEQUENCE {
: OBJECT IDENTIFIER
: commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
147 13 7: PrintableString 'CarlRSA'
: }
: }
: }
156 30 30: SEQUENCE {
158 17 13: UTCTime '990818070000Z'
173 17 13: UTCTime '391231235959Z'
: }
188 30 18: SEQUENCE {
190 31 16: SET {
192 30 14: SEQUENCE {
194 06 3: OBJECT IDENTIFIER
: commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
199 13 7: PrintableString 'CarlRSA'
: }
: }
: }
208 30 159: SEQUENCE {
211 30 13: SEQUENCE {
213 06 9: OBJECT IDENTIFIER
: rsaEncryption (1 2 840 113549 1 1 1)
: (PKCS #1)
224 05 0: NULL
: }
226 03 141: BIT STRING 0 unused bits, encapsulates {
230 30 137: SEQUENCE {
233 02 129: INTEGER
: 00 E4 4B FF 18 B8 24 57 F4 77 FF 6E
: 73 7B 93 71 5C BC 33 1A 92 92 72 23
: D8 41 46 D0 CD 11 3A 04 B3 8E AF 82
: 9D BD 51 1E 17 7A F2 76 2C 2B 86 39
: A7 BD D7 8D 1A 53 EC E4 00 D5 E8 EC
: A2 36 B1 ED E2 50 E2 32 09 8A 3F 9F
: 99 25 8F B8 4E AB B9 7D D5 96 65 DA
: 16 A0 C5 BE 0E AE 44 5B EF 5E F4 A7
: 29 CB 82 DD AC 44 E9 AA 93 94 29 0E
: F8 18 D6 C8 57 5E F2 76 C4 F2 11 60
: 38 B9 1B 3C 1D 97 C9 6A F1
365 02 3: INTEGER 65537
: }

```

```

:           }
:         }
370 A3 66: [3] {
372 30 64:   SEQUENCE {
374 30 15:     SEQUENCE {
376 06 3:       OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
:         (X.509 id-ce (2 5 29))
381 01 1:       BOOLEAN TRUE
384 04 5:       OCTET STRING, encapsulates {
386 30 3:         SEQUENCE {
388 01 1:           BOOLEAN TRUE
:         }
:       }
:     }
391 30 14:   SEQUENCE {
393 06 3:     OBJECT IDENTIFIER
:       keyUsage (2 5 29 15)
:       (X.509 id-ce (2 5 29))
398 01 1:     BOOLEAN TRUE
401 04 4:     OCTET STRING, encapsulates {
403 03 2:       BIT STRING 1 unused bits
:       '1100001'B
:     }
:   }
407 30 29: SEQUENCE {
409 06 3:   OBJECT IDENTIFIER
:     subjectKeyIdentifier (2 5 29 14)
:     (X.509 id-ce (2 5 29))
414 04 22:   OCTET STRING, encapsulates {
416 04 20:     OCTET STRING
:     E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
:     42 37 4E 22 AE 9E 38 BB
:   }
: }
: }
: }
438 30 13: SEQUENCE {
440 06 9:   OBJECT IDENTIFIER
:     sha1withRSAEncryption
:     (1 2 840 113549 1 1 5)
:     (PKCS #1)
451 05 0:   NULL
: }
453 03 129: BIT STRING 0 unused bits
:         B7 9E D4 04 D3 ED 29 E4 FF 89 89 15
:         2E 4C DB 0C F0 48 0F 32 61 EE C4 04

```

```

:          EC 12 5D 2D FF 0F 64 59 7E 0A C3 ED
:          18 FD E3 56 40 37 A7 07 B5 F0 38 12
:          61 50 ED EF DD 3F E3 0B B8 61 A5 A4
:          9B 3C E6 9E 9C 54 9A B6 95 D6 DA 6C
:          3B B5 2D 45 35 9D 49 01 76 FA B9 B9
:          31 F9 F9 6B 12 53 A0 F5 14 60 9B 7D
:          CA 3E F2 53 6B B0 37 6F AD E6 74 D7
:          DB FA 5A EA 14 41 63 5D CD BE C8 0E
:          C1 DA 6A 8D 53 34 18 02
:      }
585 30 556:      SEQUENCE {
589 30 405:      SEQUENCE {
593 A0 3:      [0] {
595 02 1:      INTEGER 2
:      }
598 02 16:      INTEGER
:      46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:      C4 10 B3 B0
616 30 13:      SEQUENCE {
618 06 9:      OBJECT IDENTIFIER
:      sha1withRSAEncryption
:      (1 2 840 113549 1 1 5)
:      (PKCS #1)
629 05 0:      NULL
:      }
631 30 18:      SEQUENCE {
633 31 16:      SET {
635 30 14:      SEQUENCE {
637 06 3:      OBJECT IDENTIFIER
:      commonName (2 5 4 3)
:      (X.520 id-at (2 5 4))
642 13 7:      PrintableString 'CarlRSA'
:      }
:      }
:      }
651 30 30:      SEQUENCE {
653 17 13:      UTCTime '990919010847Z'
668 17 13:      UTCTime '391231235959Z'
:      }
683 30 19:      SEQUENCE {
685 31 17:      SET {
687 30 15:      SEQUENCE {
689 06 3:      OBJECT IDENTIFIER
:      commonName (2 5 4 3)
:      (X.520 id-at (2 5 4))
694 13 8:      PrintableString 'AlicerSA'
:      }
:      }

```

```

:
704 30 159: SEQUENCE {
707 30 13: SEQUENCE {
709 06 9: OBJECT IDENTIFIER
: rsaEncryption (1 2 840 113549 1 1 1)
: (PKCS #1)
720 05 0: NULL
: }
722 03 141: BIT STRING 0 unused bits, encapsulates {
726 30 137: SEQUENCE {
729 02 129: INTEGER
: 00 E0 89 73 39 8D D8 F5 F5 E8 87 76
: 39 7F 4E B0 05 BB 53 83 DE 0F B7 AB
: DC 7D C7 75 29 0D 05 2E 6D 12 DF A6
: 86 26 D4 D2 6F AA 58 29 FC 97 EC FA
: 82 51 0F 30 80 BE B1 50 9E 46 44 F1
: 2C BB D8 32 CF C6 68 6F 07 D9 B0 60
: AC BE EE 34 09 6A 13 F5 F7 05 05 93
: DF 5E BA 35 56 D9 61 FF 19 7F C9 81
: E6 F8 6C EA 87 40 70 EF AC 6D 2C 74
: 9F 2D FA 55 3A B9 99 77 02 A6 48 52
: 8C 4E F3 57 38 57 74 57 5F
861 02 3: INTEGER 65537
: }
: }
: }
866 A3 129: [3] {
869 30 127: SEQUENCE {
871 30 12: SEQUENCE {
873 06 3: OBJECT IDENTIFIER
: basicConstraints (2 5 29 19)
: (X.509 id-ce (2 5 29))
878 01 1: BOOLEAN TRUE
881 04 2: OCTET STRING, encapsulates {
883 30 0: SEQUENCE {}
: }
: }
885 30 14: SEQUENCE {
887 06 3: OBJECT IDENTIFIER
: keyUsage (2 5 29 15)
: (X.509 id-ce (2 5 29))
892 01 1: BOOLEAN TRUE
895 04 4: OCTET STRING, encapsulates {
897 03 2: BIT STRING 6 unused bits
: '11'B
: }
: }
901 30 31: SEQUENCE {

```



```

903 06      3:      OBJECT IDENTIFIER
                :      authorityKeyIdentifier (2 5 29 35)
                :      (X.509 id-ce (2 5 29))
908 04      24:      OCTET STRING, encapsulates {
910 30      22:      SEQUENCE {
912 80      20:      [0]
                :      E9 E0 90 27 AC 78 20 7A 9A D3 4C F2
                :      42 37 4E 22 AE 9E 38 BB
                :      }
                :      }
                :      }
934 30      29:      SEQUENCE {
936 06      3:      OBJECT IDENTIFIER
                :      subjectKeyIdentifier (2 5 29 14)
                :      (X.509 id-ce (2 5 29))
941 04      22:      OCTET STRING, encapsulates {
943 04      20:      OCTET STRING
                :      77 D2 B4 D1 B7 4C 8A 8A A3 CE 45 9D
                :      CE EC 3C A0 3A E3 FF 50
                :      }
                :      }
965 30      31:      SEQUENCE {
967 06      3:      OBJECT IDENTIFIER
                :      subjectAltName (2 5 29 17)
                :      (X.509 id-ce (2 5 29))
972 04      24:      OCTET STRING, encapsulates {
974 30      22:      SEQUENCE {
976 81      20:      [1] 'AliceRSA@example.com'
                :      }
                :      }
                :      }
                :      }
                :      }
                :      }
998 30      13:      SEQUENCE {
1000 06      9:      OBJECT IDENTIFIER
                :      sha1withRSAEncryption
                :      (1 2 840 113549 1 1 5)
                :      (PKCS #1)
1011 05      0:      NULL
                :      }
1013 03      129:     BIT STRING 0 unused bits
                :      3E 70 47 A8 48 CC 13 58 8F CA 51 71
                :      6B 4E 36 18 5D 04 7E 80 B1 8D 4D CC
                :      CA A3 8F CC 7D 56 C8 BC CF 6E B3 1C
                :      59 A9 20 AA 05 81 A8 4E 25 AD A7 70
                :      14 75 2F F5 C7 9B D1 0E E9 63 D2 64
                :      B7 C6 66 6E 73 21 54 DF F4 BA 25 5D

```

```

:          7D 49 D3 94 6B 22 36 74 73 B8 4A EC
:          2F 64 ED D3 3D D2 A7 42 C5 E8 37 8A
:          B4 DB 9F 67 E4 BD 9F F9 FE 74 EF EA
:          F9 EE 63 6A D8 3F 4B 25 09 B5 D8 1A
:          76 AE EB 9B DB 49 B0 22
:      }
:  }
1147 31 203: SET {
1150 30 200:     SEQUENCE {
1153 02 1:         INTEGER 1
1156 30 38:         SEQUENCE {
1158 30 18:             SEQUENCE {
1160 31 16:                 SET {
1162 30 14:                     SEQUENCE {
1164 06 3:                         OBJECT IDENTIFIER
:                             commonName (2 5 4 3)
:                             (X.520 id-at (2 5 4))
1169 13 7:                             PrintableString 'CarlRSA'
:                             }
:                         }
:                     }
1178 02 16:     INTEGER
:         46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:         C4 10 B3 B0
:     }
1196 30 9:     SEQUENCE {
1198 06 5:         OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:         (OIW)
1205 05 0:         NULL
:     }
1207 30 13:     SEQUENCE {
1209 06 9:         OBJECT IDENTIFIER
:         rsaEncryption (1 2 840 113549 1 1 1)
:         (PKCS #1)
1220 05 0:         NULL
:     }
1222 04 128:     OCTET STRING
:         2F 23 82 D2 F3 09 5F B8 0C 58 EB 4E
:         9D BF 89 9A 81 E5 75 C4 91 3D D3 D0
:         D5 7B B6 D5 FE 94 A1 8A AC E3 C4 84
:         F5 CD 60 4E 27 95 F6 CF 00 86 76 75
:         3F 2B F0 E7 D4 02 67 A7 F5 C7 8D 16
:         04 A5 B3 B5 E7 D9 32 F0 24 EF E7 20
:         44 D5 9F 07 C5 53 24 FA CE 01 1D 0F
:         17 13 A7 2A 95 9D 2B E4 03 95 14 0B
:         E9 39 0D BA CE 6E 9C 9E 0C E8 98 E6
:         55 13 D4 68 6F D0 07 D7 A2 B1 62 4C
:         E3 8F AF FD E0 D5 5D C7

```

```

:      }
:    }
:  }
: }
:

```

4.6. Multiple Signers

Similar to 4.1, but the message is also signed by Diane. Two signerInfos (one for Alice, one for Diane) with no attribute certificates, each signed using DSS, Alice's and Diane's certificate (not Carl's root cert), no CRL. The message is ExContent, and is included in the eContent. There are no signed or unsigned attributes.

```

0 30 1463: SEQUENCE {
4 06 9: OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
: (PKCS #7)
15 A0 1448: [0] {
19 30 1444: SEQUENCE {
23 02 1: INTEGER 1
26 31 9: SET {
28 30 7: SEQUENCE {
30 06 5: OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
: (OIW)
: }
: }
: SEQUENCE {
37 30 43: SEQUENCE {
39 06 9: OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
: (PKCS #7)
50 A0 30: [0] {
52 04 28: OCTET STRING 'This is some sample content.'
: }
: }
82 A0 1180: [0] {
86 30 440: SEQUENCE {
90 30 375: SEQUENCE {
94 A0 3: [0] {
96 02 1: INTEGER 2
: }
99 02 2: INTEGER 210
103 30 9: SEQUENCE {
105 06 7: OBJECT IDENTIFIER
: dsaWithShal (1 2 840 10040 4 3)
: (ANSI X9.57 algorithm)
: }
114 30 18: SEQUENCE {
116 31 16: SET {

```

```

118 30 14:      SEQUENCE {
120 06 3:      OBJECT IDENTIFIER
:              commonName (2 5 4 3)
:              (X.520 id-at (2 5 4))
125 13 7:      PrintableString 'CarlDSS'
:      }
:    }
:  }
134 30 30:     SEQUENCE {
136 17 13:     UTCTime '990817020810Z'
151 17 13:     UTCTime '391231235959Z'
:     }
166 30 19:     SEQUENCE {
168 31 17:     SET {
170 30 15:     SEQUENCE {
172 06 3:     OBJECT IDENTIFIER
:           commonName (2 5 4 3)
:           (X.520 id-at (2 5 4))
177 13 8:     PrintableString 'DianeDSS'
:     }
:   }
: }
187 30 147:    SEQUENCE {
190 30 9:      SEQUENCE {
192 06 7:      OBJECT IDENTIFIER
:          dsa (1 2 840 10040 4 1)
:          (ANSI X9.57 algorithm)
:      }
201 03 133:    BIT STRING 0 unused bits, encapsulates {
205 02 129:    INTEGER
:      00 A0 00 17 78 2C EE 7E 81 53 2E 2E
:      61 08 0F A1 9B 51 52 1A DA 59 A8 73
:      2F 12 25 B6 08 CB CA EF 2A 44 76 8A
:      52 09 EA BD 05 22 D5 0F F6 FD 46 D7
:      AF 99 38 09 0E 13 CB 4F 2C DD 1C 34
:      F7 1C BF 25 FF 23 D3 3B 59 E7 82 97
:      37 BE 31 24 D8 18 C8 F3 49 39 5B B7
:      E2 E5 27 7E FC 8C 45 72 5B 7E 3E 8F
:      68 4D DD 46 7A 22 BE 8E FF CC DA 39
:      29 A3 39 E5 9F 43 E9 55 C9 D7 5B A6
:      81 67 CC C0 AA CD 2E C5 23
:    }
:  }
337 A3 129:    [3] {
340 30 127:    SEQUENCE {
342 30 12:    SEQUENCE {
344 06 3:    OBJECT IDENTIFIER
:          basicConstraints (2 5 29 19)

```

```

:                (X.509 id-ce (2 5 29))
349 01 1:        BOOLEAN TRUE
352 04 2:        OCTET STRING, encapsulates {
354 30 0:        SEQUENCE {}
:                }
:                }
356 30 14:       SEQUENCE {
358 06 3:        OBJECT IDENTIFIER
:                keyUsage (2 5 29 15)
:                (X.509 id-ce (2 5 29))
363 01 1:        BOOLEAN TRUE
366 04 4:        OCTET STRING, encapsulates {
368 03 2:        BIT STRING 6 unused bits
:                '11'B
:                }
:                }
372 30 31:       SEQUENCE {
374 06 3:        OBJECT IDENTIFIER
:                authorityKeyIdentifier (2 5 29 35)
:                (X.509 id-ce (2 5 29))
379 04 24:       OCTET STRING, encapsulates {
381 30 22:       SEQUENCE {
383 80 20:       [0]
:                70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:                3D 20 BC 43 2B 93 F1 1F
:                }
:                }
:                }
405 30 29:       SEQUENCE {
407 06 3:        OBJECT IDENTIFIER
:                subjectKeyIdentifier (2 5 29 14)
:                (X.509 id-ce (2 5 29))
412 04 22:       OCTET STRING, encapsulates {
414 04 20:       OCTET STRING
:                64 30 99 7D 5C DC 45 0B 99 3A 52 2F
:                16 BF 58 50 DD CE 2B 18
:                }
:                }
436 30 31:       SEQUENCE {
438 06 3:        OBJECT IDENTIFIER
:                subjectAltName (2 5 29 17)
:                (X.509 id-ce (2 5 29))
443 04 24:       OCTET STRING, encapsulates {
445 30 22:       SEQUENCE {
447 81 20:       [1] 'DianeDSS@example.com'
:                }
:                }
:                }

```

```

:           }
:         }
:       }
469 30      9:      SEQUENCE {
471 06      7:      OBJECT IDENTIFIER
:                  dsaWithShal (1 2 840 10040 4 3)
:                  (ANSI X9.57 algorithm)
:                }
480 03      48:      BIT STRING 0 unused bits, encapsulates {
483 30      45:      SEQUENCE {
485 02      21:      INTEGER
:                  00 A1 1A F8 17 0E 3E 5D A8 8C F4 B6
:                  55 33 1E 4B E3 2C AC B9 5F
508 02      20:      INTEGER
:                  28 4B 10 45 58 D2 1C 9D 55 35 14 18
:                  91 B2 3F 39 DF B5 6E D3
:                }
:              }
:            }
530 30      732:     SEQUENCE {
534 30      667:     SEQUENCE {
538 A0      3:       [0] {
540 02      1:       INTEGER 2
:               }
543 02      2:       INTEGER 200
547 30      9:       SEQUENCE {
549 06      7:       OBJECT IDENTIFIER
:                   dsaWithShal (1 2 840 10040 4 3)
:                   (ANSI X9.57 algorithm)
:                 }
558 30      18:     SEQUENCE {
560 31      16:     SET {
562 30      14:     SEQUENCE {
564 06      3:       OBJECT IDENTIFIER
:                   commonName (2 5 4 3)
:                   (X.520 id-at (2 5 4))
569 13      7:       PrintableString 'CarlDSS'
:               }
:             }
:           }
578 30      30:     SEQUENCE {
580 17      13:     UTCTime '990817011049Z'
595 17      13:     UTCTime '391231235959Z'
:           }
610 30      19:     SEQUENCE {
612 31      17:     SET {
614 30      15:     SEQUENCE {
616 06      3:       OBJECT IDENTIFIER

```

```

:               commonName (2 5 4 3)
:               (X.520 id-at (2 5 4))
621 13      8:      PrintableString 'AliceDSS'
:               }
:               }
:               }
631 30      438:      SEQUENCE {
635 30      299:      SEQUENCE {
639 06      7:      OBJECT IDENTIFIER
:                  dsa (1 2 840 10040 4 1)
:                  (ANSI X9.57 algorithm)
648 30      286:      SEQUENCE {
652 02      129:      INTEGER
:                  00 81 8D CD ED 83 EA 0A 9E 39 3E C2
:                  48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
:                  53 C5 AB 84 08 4F FF 94 E1 73 48 7E
:                  0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
:                  2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
:                  DC 5F 69 8A E4 75 D0 37 0C 91 08 95
:                  9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
:                  8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
:                  C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
:                  78 BD FF 9D B0 84 97 37 F2 E4 51 1B
:                  B5 E4 09 96 5C F3 7E 5B DB
784 02      21:      INTEGER
:                  00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:                  B8 37 21 2B 62 8B F7 93 CD
807 02      128:      INTEGER
:                  26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:                  4B 59 6A 4C 76 23 39 04 02 35 5C F2
:                  CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:                  AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:                  7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:                  3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:                  E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:                  01 7C 6D 49 89 11 89 36 44 BD F8 C8
:                  95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:                  1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:                  D1 81 4A 60 39 BA 36 39
:                  }
:               }
938 03      132:      BIT STRING 0 unused bits, encapsulates {
942 02      128:      INTEGER
:                  5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:                  3F A9 EC AC 5E DC BD B7 13 11 34 A6
:                  16 89 28 11 23 D9 34 86 67 75 75 13
:                  12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:                  1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45

```

```

:           A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:           7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:           08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:           F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:           32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:           F2 A5 E2 F4 F2 83 E5 B8
:           }
:       }
1073 A3 129: [3] {
1076 30 127:     SEQUENCE {
1078 30 12:       SEQUENCE {
1080 06 3:         OBJECT IDENTIFIER
:           basicConstraints (2 5 29 19)
:           (X.509 id-ce (2 5 29))
1085 01 1:         BOOLEAN TRUE
1088 04 2:         OCTET STRING, encapsulates {
1090 30 0:           SEQUENCE {}
:         }
:       }
1092 30 14:     SEQUENCE {
1094 06 3:       OBJECT IDENTIFIER
:         keyUsage (2 5 29 15)
:         (X.509 id-ce (2 5 29))
1099 01 1:       BOOLEAN TRUE
1102 04 4:       OCTET STRING, encapsulates {
1104 03 2:         BIT STRING 6 unused bits
:         '11'B
:       }
:     }
1108 30 31:     SEQUENCE {
1110 06 3:       OBJECT IDENTIFIER
:         authorityKeyIdentifier (2 5 29 35)
:         (X.509 id-ce (2 5 29))
1115 04 24:       OCTET STRING, encapsulates {
1117 30 22:         SEQUENCE {
1119 80 20:           [0]
:             70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:             3D 20 BC 43 2B 93 F1 1F
:           }
:         }
:       }
1141 30 29:     SEQUENCE {
1143 06 3:       OBJECT IDENTIFIER
:         subjectKeyIdentifier (2 5 29 14)
:         (X.509 id-ce (2 5 29))
1148 04 22:       OCTET STRING, encapsulates {
1150 04 20:         OCTET STRING
:         BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE

```



```

:           13 01 E2 FD E3 97 FE CD
:           }
:         }
1172 30    31: SEQUENCE {
1174 06    3:   OBJECT IDENTIFIER
:             subjectAltName (2 5 29 17)
:             (X.509 id-ce (2 5 29))
1179 04    24:   OCTET STRING, encapsulates {
1181 30    22:     SEQUENCE {
1183 81    20:       [1] 'AliceDSS@example.com'
:           }
:         }
:       }
:     }
:   }
: }
1205 30    9: SEQUENCE {
1207 06    7:   OBJECT IDENTIFIER
:             dsaWithShal (1 2 840 10040 4 3)
:             (ANSI X9.57 algorithm)
:           }
1216 03    48:   BIT STRING 0 unused bits, encapsulates {
1219 30    45:     SEQUENCE {
1221 02    20:       INTEGER
:             55 0C A4 19 1F 42 2B 89 71 22 33 8D
:             83 6A B5 3D 67 6B BF 45
1243 02    21:       INTEGER
:             00 9F 61 53 52 54 0B 5C B2 DD DA E7
:             76 1D E2 10 52 5B 43 5E BD
:           }
:         }
:       }
:     }
:   }
1266 31    198: SET {
1269 30    97:   SEQUENCE {
1271 02     1:     INTEGER 1
1274 30    24:     SEQUENCE {
1276 30    18:       SEQUENCE {
1278 31    16:         SET {
1280 30    14:           SEQUENCE {
1282 06     3:             OBJECT IDENTIFIER
:                 commonName (2 5 4 3)
:                 (X.520 id-at (2 5 4))
1287 13     7:             PrintableString 'CarlDSS'
:           }
:         }
:       }
:     }
1296 02     2:   INTEGER 200

```

```

:      }
1300 30 7:      SEQUENCE {
1302 06 5:      OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:      (OIW)
:      }
1309 30 9:      SEQUENCE {
1311 06 7:      OBJECT IDENTIFIER
:      dsaWithSha1 (1 2 840 10040 4 3)
:      (ANSI X9.57 algorithm)
:      }
1320 04 46:      OCTET STRING, encapsulates {
1322 30 44:      SEQUENCE {
1324 02 20:      INTEGER
:      48 24 DE 8B 85 F2 16 AF EC 82 61 A9
:      54 D0 2D 04 A1 CC 5A 4F
1346 02 20:      INTEGER
:      17 ED D5 77 02 EE 75 13 D8 10 BD 3D
:      97 17 20 88 BB FD 7B 81
:      }
:      }
:      }
1368 30 97:      SEQUENCE {
1370 02 1:      INTEGER 1
1373 30 24:      SEQUENCE {
1375 30 18:      SEQUENCE {
1377 31 16:      SET {
1379 30 14:      SEQUENCE {
1381 06 3:      OBJECT IDENTIFIER
:      commonName (2 5 4 3)
:      (X.520 id-at (2 5 4))
1386 13 7:      PrintableString 'CarlDSS'
:      }
:      }
:      }
1395 02 2:      INTEGER 210
:      }
1399 30 7:      SEQUENCE {
1401 06 5:      OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:      (OIW)
:      }
1408 30 9:      SEQUENCE {
1410 06 7:      OBJECT IDENTIFIER
:      dsaWithSha1 (1 2 840 10040 4 3)
:      (ANSI X9.57 algorithm)
:      }
1419 04 46:      OCTET STRING, encapsulates {
1421 30 44:      SEQUENCE {
1423 02 20:      INTEGER

```

```

:          15 FF 81 4D 8C AD 80 4E 9B 35 58 04
:          37 6E 63 6E E9 5B 83 FA
1445 02    20:      INTEGER
:          06 7E 58 4E 2B 31 84 41 ED 49 79 38
:          3E 77 D2 A6 8C 75 08 21
:          }
:        }
:      }
:    }
:  }
: }

```

4.7. Signing Using SKI

Same as 4.1, but the signature uses the SKI instead of the issuer/serial number in the cert. A SignedData with no attribute certificates, signed by Alice using DSS, just her certificate (not Carl's root cert), identified by the SKI, no CRL. The message is ExContent, and is included in the eContent. There are no signed or unsigned attributes.

```

0 30 915: SEQUENCE {
4 06 9:  OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
:      (PKCS #7)
15 A0 900:  [0] {
19 30 896:    SEQUENCE {
23 02 1:      INTEGER 3
26 31 9:      SET {
28 30 7:        SEQUENCE {
30 06 5:          OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:          (OIW)
:        }
:      }
:    }
37 30 43:    SEQUENCE {
39 06 9:      OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:      (PKCS #7)
50 A0 30:      [0] {
52 04 28:        OCTET STRING 'This is some sample content.'
:        }
:      }
82 A0 736:  [0] {
86 30 732:    SEQUENCE {
90 30 667:      SEQUENCE {
94 A0 3:        [0] {
96 02 1:          INTEGER 2
:          }
99 02 2:        INTEGER 200

```

```

103 30      9:      SEQUENCE {
105 06      7:      OBJECT IDENTIFIER
      :      dsaWithShal (1 2 840 10040 4 3)
      :      (ANSI X9.57 algorithm)
      :      }
114 30     18:      SEQUENCE {
116 31     16:      SET {
118 30     14:      SEQUENCE {
120 06      3:      OBJECT IDENTIFIER
      :      commonName (2 5 4 3)
      :      (X.520 id-at (2 5 4))
125 13      7:      PrintableString 'CarlDSS'
      :      }
      :      }
      :      }
134 30     30:      SEQUENCE {
136 17     13:      UTCTime '990817011049Z'
151 17     13:      UTCTime '391231235959Z'
      :      }
166 30     19:      SEQUENCE {
168 31     17:      SET {
170 30     15:      SEQUENCE {
172 06      3:      OBJECT IDENTIFIER
      :      commonName (2 5 4 3)
      :      (X.520 id-at (2 5 4))
177 13      8:      PrintableString 'AliceDSS'
      :      }
      :      }
      :      }
187 30    438:      SEQUENCE {
191 30    299:      SEQUENCE {
195 06      7:      OBJECT IDENTIFIER
      :      dsa (1 2 840 10040 4 1)
      :      (ANSI X9.57 algorithm)
204 30    286:      SEQUENCE {
208 02    129:      INTEGER
      :      00 81 8D CD ED 83 EA 0A 9E 39 3E C2
      :      48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
      :      53 C5 AB 84 08 4F FF 94 E1 73 48 7E
      :      0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
      :      2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
      :      DC 5F 69 8A E4 75 D0 37 0C 91 08 95
      :      9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
      :      8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
      :      C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
      :      78 BD FF 9D B0 84 97 37 F2 E4 51 1B
      :      B5 E4 09 96 5C F3 7E 5B DB
340 02     21:      INTEGER

```

```

:      00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:      B8 37 21 2B 62 8B F7 93 CD
363 02 128: INTEGER
:      26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:      4B 59 6A 4C 76 23 39 04 02 35 5C F2
:      CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:      AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:      7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:      3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:      E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:      01 7C 6D 49 89 11 89 36 44 BD F8 C8
:      95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:      1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:      D1 81 4A 60 39 BA 36 39
:      }
:      }
494 03 132: BIT STRING 0 unused bits, encapsulates {
498 02 128: INTEGER
:      5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:      3F A9 EC AC 5E DC BD B7 13 11 34 A6
:      16 89 28 11 23 D9 34 86 67 75 75 13
:      12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:      1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:      A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:      7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:      08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:      F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:      32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:      F2 A5 E2 F4 F2 83 E5 B8
:      }
:      }
629 A3 129: [3] {
632 30 127: SEQUENCE {
634 30 12: SEQUENCE {
636 06 3: OBJECT IDENTIFIER
:      basicConstraints (2 5 29 19)
:      (X.509 id-ce (2 5 29))
641 01 1: BOOLEAN TRUE
644 04 2: OCTET STRING, encapsulates {
646 30 0: SEQUENCE {}
:      }
:      }
648 30 14: SEQUENCE {
650 06 3: OBJECT IDENTIFIER
:      keyUsage (2 5 29 15)
:      (X.509 id-ce (2 5 29))
655 01 1: BOOLEAN TRUE
658 04 4: OCTET STRING, encapsulates {

```

```

660 03    2:          BIT STRING 6 unused bits
           :          '11'B
           :          }
           :        }
664 30    31:    SEQUENCE {
666 06    3:      OBJECT IDENTIFIER
           :      authorityKeyIdentifier (2 5 29 35)
           :      (X.509 id-ce (2 5 29))
671 04    24:      OCTET STRING, encapsulates {
673 30    22:        SEQUENCE {
675 80    20:          [0]
           :          70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
           :          3D 20 BC 43 2B 93 F1 1F
           :          }
           :        }
           :      }
697 30    29:    SEQUENCE {
699 06    3:      OBJECT IDENTIFIER
           :      subjectKeyIdentifier (2 5 29 14)
           :      (X.509 id-ce (2 5 29))
704 04    22:      OCTET STRING, encapsulates {
706 04    20:        OCTET STRING
           :        BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
           :        13 01 E2 FD E3 97 FE CD
           :        }
           :      }
728 30    31:    SEQUENCE {
730 06    3:      OBJECT IDENTIFIER
           :      subjectAltName (2 5 29 17)
           :      (X.509 id-ce (2 5 29))
735 04    24:      OCTET STRING, encapsulates {
737 30    22:        SEQUENCE {
739 81    20:          [1] 'AliceDSS@example.com'
           :          }
           :        }
           :      }
           :    }
           :  }
           :
761 30    9:    SEQUENCE {
763 06    7:      OBJECT IDENTIFIER
           :      dsaWithShal (1 2 840 10040 4 3)
           :      (ANSI X9.57 algorithm)
           :    }
772 03    48:    BIT STRING 0 unused bits, encapsulates {
775 30    45:      SEQUENCE {
777 02    20:        INTEGER
           :        55 0C A4 19 1F 42 2B 89 71 22 33 8D

```

```

      :                               83 6A B5 3D 67 6B BF 45
799 02   21:                           INTEGER
      :                               00 9F 61 53 52 54 0B 5C B2 DD DA E7
      :                               76 1D E2 10 52 5B 43 5E BD
      :                               }
      :                             }
      :                           }
      :                         }
      :                       SET {
822 31   95:                     SEQUENCE {
824 30   93:                       INTEGER 3
826 02    1:                      [0]
829 80   20:                        BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
      :                        13 01 E2 FD E3 97 FE CD
851 30    7:                     SEQUENCE {
853 06    5:                       OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
      :                       (OIW)
      :                     }
860 30    9:                     SEQUENCE {
862 06    7:                       OBJECT IDENTIFIER dsa (1 2 840 10040 4 1)
      :                       (ANSI X9.57 algorithm)
      :                     }
871 04   46:                     OCTET STRING, encapsulates {
873 30   44:                       SEQUENCE {
875 02   20:                         INTEGER
      :                         6D 8E 5A CD 28 A0 1F D9 86 AD 7A E9
      :                         DF AC D7 BE EC BE 3F F8
897 02   20:                         INTEGER
      :                         7C 8A 06 1E FC A4 41 35 7E F7 24 14
      :                         FD 3D C0 56 B7 05 27 D5
      :                         }
      :                       }
      :                     }
      :                   }
      :                 }
      :               }
      :             }
      :           }
      :         }
      :       }
      :     }
      :   }
      : }

```

4.8. S/MIME multipart/signed Message

A full S/MIME message, including MIME, that includes the body part from 4.3 and the body containing the content of the message.

MIME-Version: 1.0
To: User2@examples.com
From: aliceDss@examples.com
Subject: Example 4.8
Message-Id: <020906002550300.249@examples.com>

```
Date: Fri, 06 Sep 2002 00:25:21 -0300
Content-Type: multipart/signed;
    micalg=SHA1;
    boundary="-----_NextBoundary____Fri,_06_Sep_2002_00:25:21";
    protocol="application/pkcs7-signature"
```

This is a multi-part message in MIME format.

```
-----=_NextBoundary____Fri,_06_Sep_2002_00:25:21
```

This is some sample content.

```
-----=_NextBoundary____Fri,_06_Sep_2002_00:25:21
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
```

```
MIIDdwYJKoZIhvcNAQcCoIIDA2CCA2QCAQExCTAHBgUrDgMCGjALBgqhkiG9w0BBwGgggLG
gMIIC3DCCApugAwIBAgICAMgwCQYHKoZiZjgEAzASMRAwDgYDVQQDEwdDYXJsRFNTMB4XDT
k5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1MwggG2M
IIBKwYHKoZiZjgEATCCAR4CgYEAgY3N7YPqCp45PsJIKKPkR5PdDteoDuxTxauECE//lOFz
SH4M1vNESNH+n6+koYkv4dkwyDbep5u/t0zcX2mK5HXQNwYRCJWb3qde+fz0ny/dQ6iLVPE
/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDiR6YaRwa4E8
baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjVc8ssaMMMeUF3dmlnizaoFP
VjAe6I2uG4Hr32KQiWn9HXPSgheSz6Q+G3qnMkhiJt2FOnOLl2jB80jhbgbvMAF8bUmJEYk2
RL34yJVKU1a14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgFzjuVp1FJYLqXr
d4z+p7Kxe3L23ExE0phaJKBEj2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09B41bY8i7RaWgSu
OF1s4Ggd/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKsZfoVi/G7NNTSljf2YUeyxDKE8H5BQP1Gp
2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCBsAwHwYDVR0j
BBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVR0OBBYEFL5sobPjwfftQ3CkzhMB4v3
jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGv4YW1wbGUuY29tMAKGBYqGSM44BAMDMAAwLQ
IUVQykGR9CK4lxIjONG2q1PWdrv0UCFQCfYVNSVATcst3a53Yd4hBSW0NevTFjMGECAQEwG
DASMRawDgYDVQQDEwdDYXJsRFNTAgIAyDAHBgUrDgMCGjAJBgcqhkiG9w0BAQDBD4wLAIUM/mG
f6gk9p9Z0XtRdGimJeB/BxUCFGFFJqwyRtlWYcIOQoGiaowqGzVI
```

```
-----=_NextBoundary____Fri,_06_Sep_2002_00:25:21--
```

4.9. S/MIME application/pkcs7-mime Signed Message

A full S/MIME message, including the MIME parts.

```
MIME-Version: 1.0
To: User2@examples.com
From: alicedss@examples.com
Subject: Example 4.9
Message-Id: <021031164540300.304@examples.com>
Date: Thu, 31 Oct 2002 16:45:14 -0300
Content-Type: application/pkcs7-mime; smime-type=signed-data;
    name=smime.p7m
```


Content-Transfer-Encoding: base64

Content-Disposition: attachment; filename=smime.p7m

```
MIIDmQYJKoZIhvcNAQcCoIIDi jCCA4YCAQExCTAHBgUrDgMCGjAtBgkqhkiG9w0BBWgGIAQ
eDQpUaGlzIGlzlIHNvbWUgc2FtcGxlIGNvbnRlbnQuoIIC4DCCAtwwggKboAMCAQICAgDIMA
kGBYqGSM44BAMwEjEQMA4GA1UEAxMHQ2FybERTUzAeFw05OTA4MTcwMTEwNDlaFw0zOTEyM
zEyMzU5NTltaMBMxETAPBgNVBAMTCEFSaWNlRfNTMIIBt jCCASsGBYqGSM44BAEwgGEEAoGB
AIGNze2D6gqeOT7CSCij5EeT3Q7XqA7sU8WrhAhP/5Thc0h+DNbzREjR/p+vpKGJL+HZMMg
23j+bv7dM3F9piuR10DcMkQivm96nXvn89J8v3UOo1lTxP7AHCEdNXYjDw7Wz4lUIddU5dh
DEeL3/nbCElzf5FEbteQJllzzflvbAhUA4kemGkVmuBPG2o+4NyErYov3k80CgYAmONAUi
TKqOfs+bdllWWpMdiM5BA1lXPLLGjDDHlBd3ZtZ4s2qBTlYwHuiNrhuB699ikIlp/R1z0oI
Xks+kPht6pzJIYo7dhTpzi5dowfNI4W4LzABfG1JiRGJNks9+MiVSlNWteL5c+waYTYfEX/
Cve3RUP+YdMLRgUpGObO20QOBhAACgYBc47ladRSWC6l63eM/qeysXty9txMRNKYWiSgRI9
k0hmdldRMSpUNbb+VRv/qJ8qIbPiR9PQeNW2PIu0WloErjhdbOBoA/6CN+GvIkqlMauCcNH
u8Iv2YUGfxirGX6FYvxuzTU0pY39mFHssQyhPB+QUD9RqdjTjPypeL08oPluKOBgTB/MAwG
AlUdEwEB/wQCMAAwDgYDVROPAQH/BAQDAgBAMB8GA1UdIwQYMBaAFHBEPoIub4feStN14z0
gvEMrk/EfMB0GA1UdDgQWBBS+bKGz48H37UNwpM4TAeL945f+zTafBgNVHREEGDAWgRRBbG
ljZURTU0BleGftcGxlLmNvbTAJBgcqhkiOAAQDAZAAAMC0CFFUMpBkfQiuJcSIzjYNqtTlna
79FAhUAN2FTU1QLXLLd2ud2HeIQultDXr0xYzBhAgEBMBGwEjEQMA4GA1UEAxMHQ2FybERT
UwICAMgwBwYFKw4DAhowsCQYHkoZiZjgEAWQuMCwCFDlCSW6LIUFzeXle3YI5SKSBer/sAhQ
mCq7s/CTFHOEjgASeUjbMpx5g6A==
```

4.10. SignedData with Attributes

A SignedData message with the following list of signedAttributes:

```
-unknown OID
-contentHints
-smimeCapabilities
-securityLabel
-ContentReference
-smimeEncryptKeyPreference
-mLExpansionHistory
-EquivalentLabel
```

```
0 30 2047: SEQUENCE {
4 06      9:  OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
          :  (PKCS #7)
15 A0 2032:  [0] {
19 30 2028:  SEQUENCE {
23 02      1:  INTEGER 1
26 31      9:  SET {
28 30      7:  SEQUENCE {
30 06      5:  OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
          :  (OIW)
          :  }
          :  }
          :  }
37 30     43:  SEQUENCE {
```

```

39 06    9:      OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
           :      (PKCS #7)
50 A0    30:      [0] {
52 04    28:      OCTET STRING 'This is some sample content.'
           :      }
           :      }
82 A0    736:     [0] {
86 30    732:     SEQUENCE {
90 30    667:     SEQUENCE {
94 A0     3:      [0] {
96 02     1:      INTEGER 2
           :      }
99 02     2:      INTEGER 200
103 30     9:      SEQUENCE {
105 06     7:      OBJECT IDENTIFIER
           :      dsaWithShal (1 2 840 10040 4 3)
           :      (ANSI X9.57 algorithm)
           :      }
114 30    18:      SEQUENCE {
116 31    16:      SET {
118 30    14:      SEQUENCE {
120 06     3:      OBJECT IDENTIFIER
           :      commonName (2 5 4 3)
           :      (X.520 id-at (2 5 4))
125 13     7:      PrintableString 'CarlDSS'
           :      }
           :      }
           :      }
134 30    30:      SEQUENCE {
136 17    13:      UTCTime '990817011049Z'
151 17    13:      UTCTime '391231235959Z'
           :      }
166 30    19:      SEQUENCE {
168 31    17:      SET {
170 30    15:      SEQUENCE {
172 06     3:      OBJECT IDENTIFIER
           :      commonName (2 5 4 3)
           :      (X.520 id-at (2 5 4))
177 13     8:      PrintableString 'AliceDSS'
           :      }
           :      }
           :      }
187 30    438:     SEQUENCE {
191 30    299:     SEQUENCE {
195 06     7:      OBJECT IDENTIFIER
           :      dsa (1 2 840 10040 4 1)
           :      (ANSI X9.57 algorithm)
204 30    286:     SEQUENCE {

```

```

208 02 129:      INTEGER
:               00 81 8D CD ED 83 EA 0A 9E 39 3E C2
:               48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
:               53 C5 AB 84 08 4F FF 94 E1 73 48 7E
:               0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
:               2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
:               DC 5F 69 8A E4 75 D0 37 0C 91 08 95
:               9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
:               8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
:               C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
:               78 BD FF 9D B0 84 97 37 F2 E4 51 1B
:               B5 E4 09 96 5C F3 7E 5B DB
340 02 21:      INTEGER
:               00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:               B8 37 21 2B 62 8B F7 93 CD
363 02 128:      INTEGER
:               26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:               4B 59 6A 4C 76 23 39 04 02 35 5C F2
:               CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:               AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:               7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:               3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:               E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:               01 7C 6D 49 89 11 89 36 44 BD F8 C8
:               95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:               1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:               D1 81 4A 60 39 BA 36 39
:               }
:           }
494 03 132:      BIT STRING 0 unused bits, encapsulates {
498 02 128:      INTEGER
:               5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:               3F A9 EC AC 5E DC BD B7 13 11 34 A6
:               16 89 28 11 23 D9 34 86 67 75 75 13
:               12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:               1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:               A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:               7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:               08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:               F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:               32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:               F2 A5 E2 F4 F2 83 E5 B8
:               }
:           }
629 A3 129:      [3] {
632 30 127:      SEQUENCE {
634 30 12:      SEQUENCE {
636 06 3:      OBJECT IDENTIFIER

```

```

:                basicConstraints (2 5 29 19)
:                (X.509 id-ce (2 5 29))
641 01    1:    BOOLEAN TRUE
644 04    2:    OCTET STRING, encapsulates {
646 30    0:        SEQUENCE {}
:            }
:        }
648 30    14:    SEQUENCE {
650 06    3:        OBJECT IDENTIFIER
:            keyUsage (2 5 29 15)
:            (X.509 id-ce (2 5 29))
655 01    1:    BOOLEAN TRUE
658 04    4:    OCTET STRING, encapsulates {
660 03    2:        BIT STRING 6 unused bits
:            '11'B
:        }
:    }
664 30    31:    SEQUENCE {
666 06    3:        OBJECT IDENTIFIER
:            authorityKeyIdentifier (2 5 29 35)
:            (X.509 id-ce (2 5 29))
671 04    24:    OCTET STRING, encapsulates {
673 30    22:        SEQUENCE {
675 80    20:            [0]
:                70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:                3D 20 BC 43 2B 93 F1 1F
:            }
:        }
:    }
697 30    29:    SEQUENCE {
699 06    3:        OBJECT IDENTIFIER
:            subjectKeyIdentifier (2 5 29 14)
:            (X.509 id-ce (2 5 29))
704 04    22:    OCTET STRING, encapsulates {
706 04    20:        OCTET STRING
:            BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
:            13 01 E2 FD E3 97 FE CD
:        }
:    }
728 30    31:    SEQUENCE {
730 06    3:        OBJECT IDENTIFIER
:            subjectAltName (2 5 29 17)
:            (X.509 id-ce (2 5 29))
735 04    24:    OCTET STRING, encapsulates {
737 30    22:        SEQUENCE {
739 81    20:            [1] 'AliceDSS@example.com'
:        }
:    }

```

```

:
:
:
:
761 30 9: SEQUENCE {
763 06 7:   OBJECT IDENTIFIER
:         dsaWithSha1 (1 2 840 10040 4 3)
:         (ANSI X9.57 algorithm)
:       }
772 03 48: BIT STRING 0 unused bits, encapsulates {
775 30 45:   SEQUENCE {
777 02 20:     INTEGER
:         55 0C A4 19 1F 42 2B 89 71 22 33 8D
:         83 6A B5 3D 67 6B BF 45
799 02 21:     INTEGER
:         00 9F 61 53 52 54 0B 5C B2 DD DA E7
:         76 1D E2 10 52 5B 43 5E BD
:       }
:     }
:   }
: }
822 31 1225: SET {
826 30 1221:   SEQUENCE {
830 02 1:     INTEGER 1
833 30 24:     SEQUENCE {
835 30 18:       SEQUENCE {
837 31 16:         SET {
839 30 14:           SEQUENCE {
841 06 3:             OBJECT IDENTIFIER
:             commonName (2 5 4 3)
:             (X.520 id-at (2 5 4))
846 13 7:             PrintableString 'CarlDSS'
:           }
:         }
:       }
855 02 2:     INTEGER 200
:   }
859 30 7: SEQUENCE {
861 06 5:   OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
:   (OIW)
: }
868 A0 1119: [0] {
872 30 24:   SEQUENCE {
874 06 9:     OBJECT IDENTIFIER
:     contentType (1 2 840 113549 1 9 3)
:     (PKCS #9 (1 2 840 113549 1 9))
885 31 11:   SET {
887 06 9:     OBJECT IDENTIFIER

```

```

:           data (1 2 840 113549 1 7 1)
:           (PKCS #7)
:       }
:   }
898 30 35: SEQUENCE {
900 06 9:   OBJECT IDENTIFIER
:       messageDigest (1 2 840 113549 1 9 4)
:       (PKCS #9 (1 2 840 113549 1 9))
911 31 22: SET {
913 04 20:   OCTET STRING
:       40 6A EC 08 52 79 BA 6E 16 02 2D 9E
:       06 29 C0 22 96 87 DD 48
:   }
: }
935 30 56: SEQUENCE {
937 06 3:   OBJECT IDENTIFIER '1 2 5555'
942 31 49: SET {
944 04 47:   OCTET STRING
:       'This is a test General ASN Attribut'
:       'e, number 1.'
:   }
: }
993 30 62: SEQUENCE {
995 06 11:   OBJECT IDENTIFIER
:       id-aa-contentHint
:       (1 2 840 113549 1 9 16 2 4)
:       (S/MIME Authenticated Attributes
:       (1 2 840 113549 1 9 16 2))
1008 31 47: SET {
1010 30 45:   SEQUENCE {
1012 0C 32:     UTF8String
:       'Content Hints Description Buffer'
1046 06 9:   OBJECT IDENTIFIER
:       data (1 2 840 113549 1 7 1)
:       (PKCS #7)
:   }
: }
: }
1057 30 74: SEQUENCE {
1059 06 9:   OBJECT IDENTIFIER
:       sMIMECapabilities
:       (1 2 840 113549 1 9 15)
:       (PKCS #9
:       (1 2 840 113549 1 9))
1070 31 61: SET {
1072 30 59:   SEQUENCE {
1074 30 7:     SEQUENCE {
1076 06 5:       OBJECT IDENTIFIER '1 2 3 4 5 6'

```

```

:
1083 30 48:      }
1085 06 6:      SEQUENCE {
1093 04 38:      OBJECT IDENTIFIER '1 2 3 4 5 6 77'
:      OCTET STRING
:      'Smime Capabilities parameters buffe'
:      'r 2'
:      }
:      }
:      }
:      }
:      }
1133 30 109:    SEQUENCE {
1135 06 11:      OBJECT IDENTIFIER
:      id-aa-securityLabel
:      (1 2 840 113549 1 9 16 2 2)
:      (S/MIME Authenticated Attributes
:      (1 2 840 113549 1 9 16 2))
1148 31 94:    SET {
1150 31 92:      SET {
1152 02 1:        INTEGER 1
1155 06 7:        OBJECT IDENTIFIER '1 2 3 4 5 6 7 8'
1164 13 27:        PrintableString
:        'THIS IS A PRIVACY MARK TEST'
1193 31 49:        SET {
1195 30 47:          SEQUENCE {
1197 80 8:            [0]
:            2A 03 04 05 06 07 86 78
1207 A1 35:            [1] {
1209 13 33:              PrintableString
:              'THIS IS A TEST SECURITY-'
:              'CATEGORY.'
:              }
:            }
:          }
:        }
:      }
:    }
:  }
1244 30 111:  SEQUENCE {
1246 06 11:    OBJECT IDENTIFIER
:    id-aa-contentReference
:    (1 2 840 113549 1 9 16 2 10)
:    (S/MIME Authenticated Attributes
:    (1 2 840 113549 1 9 16 2))
1259 31 96:  SET {
1261 30 94:    SEQUENCE {
1263 06 5:      OBJECT IDENTIFIER '1 2 3 4 5 6'
1270 04 43:      OCTET STRING
:      'Content Reference Content Identifie'
:      'r Buffer'

```

```

1315 04 40:      OCTET STRING
                  :      'Content Reference Signature Value B'
                  :      'uffer'
                  :      }
                  :      }
                  :      }
1357 30 115:    SEQUENCE {
1359 06 11:      OBJECT IDENTIFIER
                  :      id-aa-encrypKeyPref
                  :      (1 2 840 113549 1 9 16 2 11)
                  :      (S/MIME Authenticated Attributes
                  :      (1 2 840 113549 1 9 16 2))
1372 31 100:    SET {
1374 A0 98:      [0] {
1376 30 90:        SEQUENCE {
1378 31 11:          SET {
1380 30 9:            SEQUENCE {
1382 06 3:              OBJECT IDENTIFIER
                  :              countryName (2 5 4 6)
                  :              (X.520 id-at (2 5 4))
1387 13 2:              PrintableString 'US'
                  :              }
                  :            }
1391 31 22:          SET {
1393 30 20:            SEQUENCE {
1395 06 3:              OBJECT IDENTIFIER
                  :              organizationName (2 5 4 10)
                  :              (X.520 id-at (2 5 4))
1400 13 13:              PrintableString 'US Government'
                  :              }
                  :            }
1415 31 17:          SET {
1417 30 15:            SEQUENCE {
1419 06 3:              OBJECT IDENTIFIER
                  :              organizationalUnitName
                  :              (2 5 4 11)
                  :              (X.520 id-at (2 5 4))
1424 13 8:              PrintableString 'VDA Site'
                  :              }
                  :            }
1434 31 12:          SET {
1436 30 10:            SEQUENCE {
1438 06 3:              OBJECT IDENTIFIER
                  :              organizationalUnitName
                  :              (2 5 4 11)
                  :              (X.520 id-at (2 5 4))
1443 13 3:              PrintableString 'VDA'
                  :              }

```



```

:
1448 31 18:      }
1450 30 16:      SET {
1452 06 3:        SEQUENCE {
:                  OBJECT IDENTIFIER
:                  commonName (2 5 4 3)
:                  (X.520 id-at (2 5 4))
1457 13 9:        PrintableString 'Daisy RSA'
:                  }
:                  }
:                  }
1468 02 4:      INTEGER 173360179
:      }
:      }
:      }
1474 30 252:     SEQUENCE {
1477 06 11:       OBJECT IDENTIFIER
:       id-aa-mExpandHistory
:       (1 2 840 113549 1 9 16 2 3)
:       (S/MIME Authenticated Attributes
:       (1 2 840 113549 1 9 16 2))
1490 31 236:     SET {
1493 30 233:       SEQUENCE {
1496 30 230:         SEQUENCE {
1499 04 7:         OCTET STRING '5738299'
1508 18 15:         GeneralizedTime '19990311104433Z'
1525 A1 201:         [1] {
1528 30 198:           SEQUENCE {
1531 A4 97:             [4] {
1533 30 95:               SEQUENCE {
1535 31 11:                 SET {
1537 30 9:                 SEQUENCE {
1539 06 3:                   OBJECT IDENTIFIER
:                   countryName (2 5 4 6)
:                   (X.520 id-at (2 5 4))
1544 13 2:                   PrintableString 'US'
:                   }
:                   }
1548 31 22:                 SET {
1550 30 20:                   SEQUENCE {
1552 06 3:                     OBJECT IDENTIFIER
:                     organizationName
:                     (2 5 4 10)
:                     (X.520 id-at (2 5 4))
1557 13 13:                     PrintableString
:                     'US Government'
:                     }
:                     }
1572 31 17:                   SET {

```

```

1574 30 15:      SEQUENCE {
1576 06 3:      OBJECT IDENTIFIER
                  organizationalUnitName
                  (2 5 4 11)
                  (X.520 id-at (2 5 4))
1581 13 8:      PrintableString
                  'VDA Site'
                  }
                  }
1591 31 12:      SET {
1593 30 10:      SEQUENCE {
1595 06 3:      OBJECT IDENTIFIER
                  organizationalUnitName
                  (2 5 4 11)
                  (X.520 id-at (2 5 4))
1600 13 3:      PrintableString 'VDA'
                  }
                  }
1605 31 23:      SET {
1607 30 21:      SEQUENCE {
1609 06 3:      OBJECT IDENTIFIER
                  commonName (2 5 4 3)
                  (X.520 id-at (2 5 4))
1614 13 14:      PrintableString
                  'Bugs Bunny DSA'
                  }
                  }
                  }
                  }
1630 A4 97:      [4] {
1632 30 95:      SEQUENCE {
1634 31 11:      SET {
1636 30 9:      SEQUENCE {
1638 06 3:      OBJECT IDENTIFIER
                  countryName (2 5 4 6)
                  (X.520 id-at (2 5 4))
1643 13 2:      PrintableString 'US'
                  }
                  }
1647 31 22:      SET {
1649 30 20:      SEQUENCE {
1651 06 3:      OBJECT IDENTIFIER
                  organizationName
                  (2 5 4 10)
                  (X.520 id-at (2 5 4))
1656 13 13:      PrintableString
                  'US Government'
                  }

```

```

:
1671 31 17:      }
1673 30 15:      SET {
1675 06 3:        SEQUENCE {
:                  OBJECT IDENTIFIER
:                  organizationalUnitName
:                  (2 5 4 11)
:                  (X.520 id-at (2 5 4))
1680 13 8:        PrintableString
:                  'VDA Site'
:                  }
:        }
1690 31 12:      SET {
1692 30 10:        SEQUENCE {
1694 06 3:          OBJECT IDENTIFIER
:          organizationalUnitName
:          (2 5 4 11)
:          (X.520 id-at (2 5 4))
1699 13 3:        PrintableString 'VDA'
:        }
:      }
1704 31 23:      SET {
1706 30 21:        SEQUENCE {
1708 06 3:          OBJECT IDENTIFIER
:          commonName (2 5 4 3)
:          (X.520 id-at (2 5 4))
1713 13 14:        PrintableString
:        'Elmer Fudd DSA'
:        }
:      }
:    }
:  }
: }
1729 30 258: SEQUENCE {
1733 06 11:   OBJECT IDENTIFIER
:   id-aa-equivalentLabels
:   (1 2 840 113549 1 9 16 2 9)
:   (S/MIME Authenticated Attributes
:   (1 2 840 113549 1 9 16 2))
1746 31 242: SET {
1749 30 239:   SEQUENCE {
1752 31 114:     SET {
1754 02 1:       INTEGER 1
1757 06 7:       OBJECT IDENTIFIER '1 2 3 4 5 6 7 9'

```

```

1766 13 38:      PrintableString
                  :      'EQUIVALENT THIS IS A PRIVACY MARK T'
                  :      'EST'
1806 31 60:      SET {
1808 30 58:          SEQUENCE {
1810 80 8:          [0]
                  :      2A 03 04 05 06 07 86 78
1820 A1 46:          [1] {
1822 13 44:              PrintableString
                  :      'EQUIVALENT THIS IS A TEST SECURITY-'
                  :      'CATEGORY.'
                  :      }
                  :      }
                  :      }
1868 31 121:     SET {
1870 02 1:         INTEGER 1
1873 06 7:         OBJECT IDENTIFIER
                  :      '1 2 3 4 5 6 7 10'
1882 13 45:         PrintableString
                  :      'EQUIVALENT THIS IS A SECOND PRIVACY'
                  :      ' MARK TEST'
1929 31 60:         SET {
1931 30 58:             SEQUENCE {
1933 80 8:             [0]
                  :      2A 03 04 05 06 07 86 78
1943 A1 46:             [1] {
1945 13 44:                 PrintableString
                  :      'EQUIVALENT THIS IS A TEST SECURITY-'
                  :      'CATEGORY.'
                  :      }
                  :      }
                  :      }
                  :      }
                  :      }
1991 30 9:         SEQUENCE {
1993 06 7:             OBJECT IDENTIFIER
                  :      dsaWithShal (1 2 840 10040 4 3)
                  :      (ANSI X9.57 algorithm)
                  :      }
2002 04 47:         OCTET STRING, encapsulates {
2004 30 45:             SEQUENCE {
2006 02 21:                 INTEGER
                  :      00 BC 33 37 65 C4 F7 70 5C 17 49 13
                  :      AA 4C 85 CA BB 52 91 48 59

```

```

2029 02    20:                INTEGER
          :                63 96 A2 14 8B CF 57 DE B0 48 5F 6C
          :                64 DD 84 04 49 5F 1C CA
          :                }
          :            }
          :        }
          :    }
          : }
          :

```

4.11. SignedData with Certificates Only

CA SignedData message with no content or signature, containing only Alices's and Carl's certificates.

```

0 30 1672: SEQUENCE {
4 06      9:  OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
          :      (PKCS #7)
15 A0 1657:  [0] {
19 30 1653:      SEQUENCE {
23 02      1:          INTEGER 1
26 31      0:          SET {}
28 30      11:         SEQUENCE {
30 06      9:          OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
          :          (PKCS #7)
          :          }
41 A0 1407:  [0] {
45 30      667:      SEQUENCE {
49 30      602:          SEQUENCE {
53 A0      3:              [0] {
55 02      1:                  INTEGER 2
          :                  }
58 02      1:                  INTEGER 1
61 30      9:                  SEQUENCE {
63 06      7:                      OBJECT IDENTIFIER
          :                      dsaWithShal (1 2 840 10040 4 3)
          :                      (ANSI X9.57 algorithm)
          :                      }
72 30      18:          SEQUENCE {
74 31      16:              SET {
76 30      14:                  SEQUENCE {
78 06      3:                      OBJECT IDENTIFIER
          :                      commonName (2 5 4 3)
          :                      (X.520 id-at (2 5 4))
83 13      7:                      PrintableString 'CarlDSS'
          :                      }
          :          }

```

```

:
92 30 30:      }
94 17 13:      SEQUENCE {
109 17 13:      UTCTime '990816225050Z'
:              UTCTime '391231235959Z'
:              }
124 30 18:      SEQUENCE {
126 31 16:      SET {
128 30 14:      SEQUENCE {
130 06 3:      OBJECT IDENTIFIER
:              commonName (2 5 4 3)
:              (X.520 id-at (2 5 4))
135 13 7:      PrintableString 'CarlDSS'
:      }
:    }
:  }
144 30 439:    SEQUENCE {
148 30 299:    SEQUENCE {
152 06 7:    OBJECT IDENTIFIER
:            dsa (1 2 840 10040 4 1)
:            (ANSI X9.57 algorithm)
161 30 286:    SEQUENCE {
165 02 129:    INTEGER
:            00 B6 49 18 3E 8A 44 C1 29 71 94 4C
:            01 C4 12 C1 7A 79 CB 54 4D AB 1E 81
:            FB C6 4C B3 0E 94 09 06 EB 01 D4 B1
:            C8 71 4B C7 45 C0 50 25 5D 9C FC DA
:            E4 6D D3 E2 86 48 84 82 7D BA 15 95
:            4A 16 F6 46 ED DD F6 98 D2 BB 7E 8A
:            0A 8A BA 16 7B B9 50 01 48 93 8B EB
:            25 15 51 97 55 DC 8F 53 0E 10 A9 50
:            FC 70 B7 CD 30 54 FD DA DE A8 AA 22
:            B5 A1 AF 8B CC 02 88 E7 8B 70 5F B9
:            AD E1 08 D4 6D 29 2D D6 E9
297 02 21:    INTEGER
:            00 DD C1 2F DF 53 CE 0B 34 60 77 3E
:            02 A4 BF 8A 5D 98 B9 10 D5
320 02 128:    INTEGER
:            0C EE 57 9B 4B BD DA B6 07 6A 74 37
:            4F 55 7F 9D ED BC 61 0D EB 46 59 3C
:            56 0B 2B 5B 0C 91 CE A5 62 52 69 CA
:            E1 6D 3E BD BF FE E1 B7 B9 2B 61 3C
:            AD CB AE 45 E3 06 AC 8C 22 9D 9C 44
:            87 0B C7 CD F0 1C D9 B5 4E 5D 73 DE
:            AF 0E C9 1D 5A 51 F5 4F 44 79 35 5A
:            73 AA 7F 46 51 1F A9 42 16 9C 48 EB
:            8A 79 61 B4 D5 2F 53 22 44 63 1F 86
:            B8 A3 58 06 25 F8 29 C0 EF BA E0 75
:            F0 42 C4 63 65 52 9B 0A

```

```

:      }
:    }
451 03 133:    BIT STRING 0 unused bits, encapsulates {
455 02 129:    INTEGER
:      00 99 87 74 27 03 66 A0 B1 C0 AD DC
:      2C 75 BB E1 6C 44 9C DA 21 6D 4D 47
:      6D B1 62 09 E9 D8 AE 1E F2 3A B4 94
:      B1 A3 8E 7A 9B 71 4E 00 94 C9 B4 25
:      4E B9 60 96 19 24 01 F3 62 0C FE 75
:      C0 FB CE D8 68 00 E3 FD D5 70 4F DF
:      23 96 19 06 94 F4 B1 61 8F 3A 57 B1
:      08 11 A4 0B 26 25 F0 52 76 81 EA 0B
:      62 0D 95 2A E6 86 BA 72 B2 A7 50 83
:      0B AA 27 CD 1B A9 4D 89 9A D7 8D 18
:      39 84 3F 8B C5 56 4D 80 7A
:    }
:  }
587 A3 66:    [3] {
589 30 64:      SEQUENCE {
591 30 15:        SEQUENCE {
593 06 3:          OBJECT IDENTIFIER
:            basicConstraints (2 5 29 19)
:            (X.509 id-ce (2 5 29))
598 01 1:          BOOLEAN TRUE
601 04 5:          OCTET STRING, encapsulates {
603 30 3:            SEQUENCE {
605 01 1:              BOOLEAN TRUE
:            }
:          }
:        }
608 30 14:      SEQUENCE {
610 06 3:        OBJECT IDENTIFIER
:          keyUsage (2 5 29 15)
:          (X.509 id-ce (2 5 29))
615 01 1:        BOOLEAN TRUE
618 04 4:        OCTET STRING, encapsulates {
620 03 2:          BIT STRING 1 unused bits
:            '1100001'B
:          }
:        }
624 30 29:      SEQUENCE {
626 06 3:        OBJECT IDENTIFIER
:          subjectKeyIdentifier (2 5 29 14)
:          (X.509 id-ce (2 5 29))
631 04 22:        OCTET STRING, encapsulates {
633 04 20:          OCTET STRING
:            70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:            3D 20 BC 43 2B 93 F1 1F

```

```

:
:
:
:
:
:
655 30 9: SEQUENCE {
657 06 7: OBJECT IDENTIFIER
: dsaWithShal (1 2 840 10040 4 3)
: (ANSI X9.57 algorithm)
:
:
666 03 48: BIT STRING 0 unused bits, encapsulates {
669 30 45: SEQUENCE {
671 02 20: INTEGER
: 6B A9 F0 4E 7A 5A 79 E3 F9 BE 3D 2B
: C9 06 37 E9 11 17 A1 13
693 02 21: INTEGER
: 00 8F 34 69 2A 8B B1 3C 03 79 94 32
: 4D 12 1F CE 89 FB 46 B2 3B
:
:
:
:
:
716 30 732: SEQUENCE {
720 30 667: SEQUENCE {
724 A0 3: [0] {
726 02 1: INTEGER 2
:
:
729 02 2: INTEGER 200
733 30 9: SEQUENCE {
735 06 7: OBJECT IDENTIFIER
: dsaWithShal (1 2 840 10040 4 3)
: (ANSI X9.57 algorithm)
:
:
744 30 18: SEQUENCE {
746 31 16: SET {
748 30 14: SEQUENCE {
750 06 3: OBJECT IDENTIFIER
: commonName (2 5 4 3)
: (X.520 id-at (2 5 4))
755 13 7: PrintableString 'CarlDSS'
:
:
:
:
764 30 30: SEQUENCE {
766 17 13: UTCTime '990817011049Z'
781 17 13: UTCTime '391231235959Z'
:
:
796 30 19: SEQUENCE {
798 31 17: SET {

```



```

800 30 15:      SEQUENCE {
802 06 3:      OBJECT IDENTIFIER
:              commonName (2 5 4 3)
:              (X.520 id-at (2 5 4))
807 13 8:      PrintableString 'AliceDSS'
:      }
:    }
:  }
817 30 438:    SEQUENCE {
821 30 299:    SEQUENCE {
825 06 7:      OBJECT IDENTIFIER
:      dsa (1 2 840 10040 4 1)
:      (ANSI X9.57 algorithm)
834 30 286:    SEQUENCE {
838 02 129:    INTEGER
:      00 81 8D CD ED 83 EA 0A 9E 39 3E C2
:      48 28 A3 E4 47 93 DD 0E D7 A8 0E EC
:      53 C5 AB 84 08 4F FF 94 E1 73 48 7E
:      0C D6 F3 44 48 D1 FE 9F AF A4 A1 89
:      2F E1 D9 30 C8 36 DE 3F 9B BF B7 4C
:      DC 5F 69 8A E4 75 D0 37 0C 91 08 95
:      9B DE A7 5E F9 FC F4 9F 2F DD 43 A8
:      8B 54 F1 3F B0 07 08 47 4D 5D 88 C3
:      C3 B5 B3 E3 55 08 75 D5 39 76 10 C4
:      78 BD FF 9D B0 84 97 37 F2 E4 51 1B
:      B5 E4 09 96 5C F3 7E 5B DB
970 02 21:    INTEGER
:      00 E2 47 A6 1A 45 66 B8 13 C6 DA 8F
:      B8 37 21 2B 62 8B F7 93 CD
993 02 128:    INTEGER
:      26 38 D0 14 89 32 AA 39 FB 3E 6D D9
:      4B 59 6A 4C 76 23 39 04 02 35 5C F2
:      CB 1A 30 C3 1E 50 5D DD 9B 59 E2 CD
:      AA 05 3D 58 C0 7B A2 36 B8 6E 07 AF
:      7D 8A 42 25 A7 F4 75 CF 4A 08 5E 4B
:      3E 90 F8 6D EA 9C C9 21 8A 3B 76 14
:      E9 CE 2E 5D A3 07 CD 23 85 B8 2F 30
:      01 7C 6D 49 89 11 89 36 44 BD F8 C8
:      95 4A 53 56 B5 E2 F9 73 EC 1A 61 36
:      1F 11 7F C2 BD ED D1 50 FF 98 74 C2
:      D1 81 4A 60 39 BA 36 39
:    }
:  }
1124 03 132:  BIT STRING 0 unused bits, encapsulates {
1128 02 128:  INTEGER
:      5C E3 B9 5A 75 14 96 0B A9 7A DD E3
:      3F A9 EC AC 5E DC BD B7 13 11 34 A6
:      16 89 28 11 23 D9 34 86 67 75 75 13

```

```

:      12 3D 43 5B 6F E5 51 BF FA 89 F2 A2
:      1B 3E 24 7D 3D 07 8D 5B 63 C8 BB 45
:      A5 A0 4A E3 85 D6 CE 06 80 3F E8 23
:      7E 1A F2 24 AB 53 1A B8 27 0D 1E EF
:      08 BF 66 14 80 5C 62 AC 65 FA 15 8B
:      F1 BB 34 D4 D2 96 37 F6 61 47 B2 C4
:      32 84 F0 7E 41 40 FD 46 A7 63 4E 33
:      F2 A5 E2 F4 F2 83 E5 B8
:      }
:    }
1259 A3 129: [3] {
1262 30 127:   SEQUENCE {
1264 30 12:     SEQUENCE {
1266 06 3:       OBJECT IDENTIFIER
:         basicConstraints (2 5 29 19)
:         (X.509 id-ce (2 5 29))
1271 01 1:       BOOLEAN TRUE
1274 04 2:       OCTET STRING, encapsulates {
1276 30 0:         SEQUENCE {}
:       }
:     }
1278 30 14:   SEQUENCE {
1280 06 3:     OBJECT IDENTIFIER
:       keyUsage (2 5 29 15)
:       (X.509 id-ce (2 5 29))
1285 01 1:     BOOLEAN TRUE
1288 04 4:     OCTET STRING, encapsulates {
1290 03 2:       BIT STRING 6 unused bits
:       '11'B
:     }
:   }
1294 30 31: SEQUENCE {
1296 06 3:   OBJECT IDENTIFIER
:     authorityKeyIdentifier (2 5 29 35)
:     (X.509 id-ce (2 5 29))
1301 04 24:   OCTET STRING, encapsulates {
1303 30 22:     SEQUENCE {
1305 80 20:       [0]
:       70 44 3E 82 2E 6F 87 DE 4A D3 75 E3
:       3D 20 BC 43 2B 93 F1 1F
:     }
:   }
1327 30 29: SEQUENCE {
1329 06 3:   OBJECT IDENTIFIER
:     subjectKeyIdentifier (2 5 29 14)
:     (X.509 id-ce (2 5 29))
1334 04 22:   OCTET STRING, encapsulates {

```

```

1336 04 20:          OCTET STRING
      :          BE 6C A1 B3 E3 C1 F7 ED 43 70 A4 CE
      :          13 01 E2 FD E3 97 FE CD
      :          }
      :          }
1358 30 31:          SEQUENCE {
1360 06 3:            OBJECT IDENTIFIER
      :            subjectAltName (2 5 29 17)
      :            (X.509 id-ce (2 5 29))
1365 04 24:          OCTET STRING, encapsulates {
1367 30 22:            SEQUENCE {
1369 81 20:              [1] 'AliceDSS@example.com'
      :              }
      :            }
      :          }
      :        }
      :      }
      :    }
1391 30 9:          SEQUENCE {
1393 06 7:            OBJECT IDENTIFIER
      :            dsaWithShal (1 2 840 10040 4 3)
      :            (ANSI X9.57 algorithm)
      :          }
1402 03 48:          BIT STRING 0 unused bits, encapsulates {
1405 30 45:            SEQUENCE {
1407 02 20:              INTEGER
      :              55 0C A4 19 1F 42 2B 89 71 22 33 8D
      :              83 6A B5 3D 67 6B BF 45
1429 02 21:            INTEGER
      :            00 9F 61 53 52 54 0B 5C B2 DD DA E7
      :            76 1D E2 10 52 5B 43 5E BD
      :          }
      :        }
      :      }
      :    }
1452 A1 219:        [1] {
1455 30 216:          SEQUENCE {
1458 30 153:            SEQUENCE {
1461 30 9:              SEQUENCE {
1463 06 7:                OBJECT IDENTIFIER
      :                dsaWithShal (1 2 840 10040 4 3)
      :                (ANSI X9.57 algorithm)
      :              }
1472 30 18:            SEQUENCE {
1474 31 16:              SET {
1476 30 14:                SEQUENCE {
1478 06 3:                  OBJECT IDENTIFIER
      :                  commonName (2 5 4 3)

```

```

:                               (X.520 id-at (2 5 4))
1483 13   7:   PrintableString 'CarlDSS'
:           }
:         }
:       }
1492 17   13:   UTCTime '990827070000Z'
1507 30   105:  SEQUENCE {
1509 30   19:    SEQUENCE {
1511 02   2:      INTEGER 200
1515 17   13:      UTCTime '990822070000Z'
:            }
1530 30   19:    SEQUENCE {
1532 02   2:      INTEGER 201
1536 17   13:      UTCTime '990822070000Z'
:            }
1551 30   19:    SEQUENCE {
1553 02   2:      INTEGER 211
1557 17   13:      UTCTime '990822070000Z'
:            }
1572 30   19:    SEQUENCE {
1574 02   2:      INTEGER 210
1578 17   13:      UTCTime '990822070000Z'
:            }
1593 30   19:    SEQUENCE {
1595 02   2:      INTEGER 212
1599 17   13:      UTCTime '990824070000Z'
:            }
:          }
:        }
1614 30   9:    SEQUENCE {
1616 06   7:      OBJECT IDENTIFIER
:                dsaWithShal (1 2 840 10040 4 3)
:                (ANSI X9.57 algorithm)
:            }
1625 03   47:  BIT STRING 0 unused bits, encapsulates {
1628 30   44:    SEQUENCE {
1630 02   20:      INTEGER
:                7E 65 52 76 33 FE 34 73 17 D1 F7 96
:                F9 A0 D4 D8 6D 5C 7D 3D
1652 02   20:      INTEGER
:                02 7A 5B B7 D5 5B 18 C1 CF 87 EF 7E
:                DA 24 F3 2A 83 9C 35 A1
:            }
:          }
:        }
:      }
1674 31   0:    SET {}
:  }

```

```

:   }
: }

```

5. Enveloped-data

5.1. Basic Encrypted Content, TripleDES and RSA

An EnvelopedData from Alice to Bob of ExContent using TripleDES for encrypting and RSA for key management. Does not have an OriginatorInfo.

```

0 30 286: SEQUENCE {
4 06 9:  OBJECT IDENTIFIER
:      envelopedData (1 2 840 113549 1 7 3)
:      (PKCS #7)
15 A0 271:  [0] {
19 30 267:    SEQUENCE {
23 02 1:      INTEGER 0
26 31 192:    SET {
29 30 189:      SEQUENCE {
32 02 1:        INTEGER 0
35 30 38:        SEQUENCE {
37 30 18:          SEQUENCE {
39 31 16:            SET {
41 30 14:              SEQUENCE {
43 06 3:                OBJECT IDENTIFIER
:                commonName (2 5 4 3)
:                (X.520 id-at (2 5 4))
48 13 7:                PrintableString 'CarlRSA'
:              }
:            }
:          }
57 02 16:        INTEGER
:        46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:        CD 5D 71 D0
:      }
75 30 13:    SEQUENCE {
77 06 9:      OBJECT IDENTIFIER
:      rsaEncryption (1 2 840 113549 1 1 1)
:      (PKCS #1)
88 05 0:    NULL
:    }
90 04 128:  OCTET STRING
:      0B 71 0D E6 71 88 88 98 B6 96 C1 8F
:      70 FD A2 27 DE DA E1 EF 24 6C A4 33
:      DF AC E0 E9 9D A2 D3 2C 7A CD 80 B8
:      99 9E E6 5F B1 41 B3 72 16 83 E7 FA
:      2A 00 8B C7 73 35 78 26 D6 C7 CF 8C

```

```

:          0C 56 DB A5 76 9D 08 38 0E F3 F9 D4
:          91 43 58 78 DC 49 B6 EC EE 6C 68 33
:          A3 21 1D F0 28 78 1F F7 5D F6 07 73
:          4D DF AD 69 31 20 4B 48 A9 75 22 6E
:          36 79 15 63 8F CC EB 9D A3 28 A1 D1
:          2C 57 F4 DA 1A 2C 75 1F
:      }
:  }
221 30 67: SEQUENCE {
223 06 9:   OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:       (PKCS #7)
234 30 20: SEQUENCE {
236 06 8:   OBJECT IDENTIFIER
:       des-EDE3-CBC (1 2 840 113549 3 7)
:       (RSADSI encryptionAlgorithm
:       (1 2 840 113549 3))
246 04 8:   OCTET STRING
:       2D 68 C5 E9 47 06 51 35
:       }
256 80 32: [0]
:       0E C8 92 7F C6 7D 3F 8D CB AD 8E 0E
:       C5 49 3A EB 47 2E D6 55 DE 09 21 4E
:       48 EA 4E 27 B1 6E 57 25
:       }
:   }
: }
: }

```

5.2. Basic Encrypted Content, RC2/128 and RSA

Same as 5.1, except using RC2/128 for encryption and RSA for key management. An EnvelopedData from Alice to Bob of ExContent using RC2/40 for encrypting and RSA for key management. Does not have an OriginatorInfo or any attributes.

```

0 30 291: SEQUENCE {
4 06 9:   OBJECT IDENTIFIER
:       envelopedData (1 2 840 113549 1 7 3)
:       (PKCS #7)
15 A0 276: [0] {
19 30 272: SEQUENCE {
23 02 1:   INTEGER 0
26 31 192: SET {
29 30 189: SEQUENCE {
32 02 1:   INTEGER 0
35 30 38: SEQUENCE {
37 30 18: SEQUENCE {
39 31 16: SET {

```

```

41 30 14:      SEQUENCE {
43 06 3:      OBJECT IDENTIFIER
:              commonName (2 5 4 3)
:              (X.520 id-at (2 5 4))
48 13 7:      PrintableString 'CarlRSA'
:              }
:      }
:      }
57 02 16:     INTEGER
:              46 34 6B C7 80 00 56 BC 11 D3 6E 2E
:              CD 5D 71 D0
:      }
75 30 13:     SEQUENCE {
77 06 9:     OBJECT IDENTIFIER
:             rsaEncryption (1 2 840 113549 1 1 1)
:             (PKCS #1)
88 05 0:     NULL
:     }
90 04 128:    OCTET STRING
:             85 42 BE E3 0B 2E E5 0F 09 AA 24 CA
:             DE DA C1 D3 09 B8 27 2B 25 CB D5 71
:             FB C9 9C DB F0 B2 6E A0 8A 5F 1C 9D
:             4A ED 98 9D 15 39 26 01 1A 2E 6B F0
:             44 39 89 37 3C 6F C7 4A 61 0B 0B 27
:             77 AA F9 D4 97 A4 D2 21 3F C2 3F 20
:             D4 DC 10 E9 D6 3F 00 DB 9C 82 47 D6
:             7E 96 FF 12 6E 87 84 A0 BA ED 81 0F
:             56 6D A6 1D EB AB C3 B7 A1 B9 F8 5F
:             8B CC 1B 4A E5 14 36 06 61 D0 C7 64
:             5F 69 67 91 A9 50 EE D8
:     }
:     }
221 30 72:    SEQUENCE {
223 06 9:    OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
:            (PKCS #7)
234 30 25:    SEQUENCE {
236 06 8:    OBJECT IDENTIFIER rc2CBC (1 2 840 113549 3 2)
:            (RSADSI encryptionAlgorithm
:            (1 2 840 113549 3))
246 30 13:    SEQUENCE {
248 02 1:    INTEGER 58
251 04 8:    OCTET STRING
:            E8 70 81 E2 EF C5 15 57
:    }
:    }
261 80 32:    [0]
:            06 53 0A 7B 8D 5C 16 0D CC D5 76 D6
:            8B 59 D6 45 8C 1A 1A 0C E6 1E F3 DE

```

```

:           43 56 00 9B 40 8C 38 5D
:           }
:         }
:       }
:     }

```

5.3. S/MIME application/pkcs7-mime Encrypted Message

A full S/MIME message, including MIME, that includes the body part from 5.1.

```

MIME-Version: 1.0
Message-Id: <00103112005203.00349@amyemily.ig.com>
Date: Tue, 31 Oct 2000 12:00:52 -0600 (Central Standard Time)
From: User1
To: User2
Subject: Example 5.3
Content-Type: application/pkcs7-mime;
              name=smime.p7m;
              smime-type=enveloped-data
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m

```

```

MIIBHgYJKoZIhvcNAQcDoIIBDzCCAQsCAQAxcAwgb0CAQAwJjASMRAwDgYDVQQDEwdDYXJ
sUlnBAhBGNGvHgABWvBHTbi7NXXHQMA0GCSqGSIb3DQEBAQUABIGAC3EN5nGIiJi2lsGPcP
2iJ97a4e8kbKQz36zg6Z2i0yx6zYC4mZ7mX7FBs3IWg+f6KgCLx3MleCbWx8+MDFbbpXadC
DgO8/nUkUNYeNxJtuzubGgzoyEd8Ch4H/dd9gdzTd+taTEgS0ipdSJUNnkVY4/M652jKKHR
LFf02hosdR8wQwYJKoZIhvcNAQcBMBQGCCqGSIb3DQMHBAGtaMXpRwZRNyAgDsiSf8Z9P43
LrY40xUk660cullXeCSFOSOpOJ7FuVyU=

```

6. Digested-data

A DigestedData from Alice to Bob of ExContent using SHA-1.

```

0 30    94: SEQUENCE {
2 06    9:   OBJECT IDENTIFIER digestedData (1 2 840 113549 1 7 5)
      :   (PKCS #7)
13 A0   81:   [0] {
15 30   79:     SEQUENCE {
17 02    1:       INTEGER 0
20 30    7:       SEQUENCE {
22 06    5:         OBJECT IDENTIFIER sha1 (1 3 14 3 2 26)
      :         (OIW)
      :       }
29 30   43:     SEQUENCE {
31 06    9:       OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
      :       (PKCS #7)
42 A0   30:     [0] {

```



```

44 04 28:      OCTET STRING 'This is some sample content.'
      :      }
      :      }
74 04 20:      OCTET STRING
      :      40 6A EC 08 52 79 BA 6E 16 02 2D 9E
      :      06 29 C0 22 96 87 DD 48
      :      }
      :      }
      :      }

```

7. Encrypted-data

7.1. Simple EncryptedData

An EncryptedData from Alice to Bob of ExContent with no attributes.

```

0 30 87: SEQUENCE {
2 06 9:  OBJECT IDENTIFIER
      :  encryptedData (1 2 840 113549 1 7 6)
      :  (PKCS #7)
13 A0 74:  [0] {
15 30 72:  SEQUENCE {
17 02 1:  INTEGER 0
20 30 67:  SEQUENCE {
22 06 9:  OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
      :  (PKCS #7)
33 30 20:  SEQUENCE {
35 06 8:  OBJECT IDENTIFIER
      :  des-EDE3-CBC (1 2 840 113549 3 7)
      :  (RSADSI encryptionAlgorithm
      :  (1 2 840 113549 3))
45 04 8:  OCTET STRING
      :  B3 6B 6B FB 62 31 08 4E
      :  }
55 80 32:  [0]
      :  FA FC ED DB 3F 18 17 1D 38 89 11 EA
      :  34 D6 20 DB F4 C3 D9 58 15 EF 93 3B
      :  9A F5 D7 04 F6 B5 70 E2
      :  }
      :  }
      :  }
      :  }

```

The TripleDES key is:

```

73 7c 79 1f 25 ea d0 e0 46 29 25 43 52 f7 dc 62
91 e5 cb 26 91 7a da 32

```

7.2. EncryptedData with Unprotected Attributes

An EncryptedData from Alice to Bob of ExContent with unprotected attributes.

```

0 30 149: SEQUENCE {
3 06 9:  OBJECT IDENTIFIER
      :  encryptedData (1 2 840 113549 1 7 6)
      :  (PKCS #7)
14 A0 135:  [0] {
17 30 132:    SEQUENCE {
20 02 1:      INTEGER 2
23 30 67:      SEQUENCE {
25 06 9:        OBJECT IDENTIFIER data (1 2 840 113549 1 7 1)
      :        (PKCS #7)
36 30 20:        SEQUENCE {
38 06 8:          OBJECT IDENTIFIER
      :          des-EDE3-CBC (1 2 840 113549 3 7)
      :          (RSADSI encryptionAlgorithm
      :            (1 2 840 113549 3))
48 04 8:          OCTET STRING
      :          07 27 20 85 90 9E B0 7E
      :        }
58 80 32:      [0]
      :      D2 20 8F 67 48 8A CB 41 E4 22 68 5D
      :      BE 77 05 52 26 ED E3 01 BD 00 91 58
      :      A7 35 6E BC 4B A2 07 33
      :    }
92 A1 58:      [1] {
94 30 56:        SEQUENCE {
96 06 3:          OBJECT IDENTIFIER '1 2 5555'
101 31 49:          SET {
103 04 47:            OCTET STRING
      :            'This is a test General ASN Attribut'
      :            'e, number 1.'
      :          }
      :        }
      :      }
      :    }
      :  }
      : }

```

8. Security Considerations

Because this document shows examples of S/MIME and CMS messages, this document also inherits all of the security considerations from [SMIME-MSG] and [CMS].

The Perl script in Appendix A writes to the user's local hard drive. A malicious attacker could modify the Perl script in this document. Be sure to read the Perl code carefully before executing it.

9. References

9.1. Normative References

- [CMS] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004.
- [PKIX] Housley, R., Polk, W., Ford, W., and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002.
- [SMIME-MSG] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, July 2004.

9.2. Informative References

- [DVCS] Adams, C., Sylvester, P., Zolotarev, M., and R. Zuccherato, "Internet X.509 Public Key Infrastructure Data Validation and Certification Server Protocols", RFC 3029, February 2001.

A. Binaries of the Examples

This section contains the binaries of the examples shown in the rest of the document. The binaries are stored in a modified Base64 format. There is a Perl program that, when run over the contents of this document, will extract the following binaries and write them out to disk. The program requires Perl.

A.1. How the Binaries and Extractor Works

The program in the next section looks for lines that begin with a '|' character (or some whitespace followed by a '|'), ignoring all other lines. If the line begins with '|', the second character tells what kind of line it is:

```
A line that begins with |* is a comment
A line that begins with |> gives the name of a new file to start
A line that begins with |< tells to end the file (and checks the
    file name for sanity)
A line that begins with |anythingelse is a Base64 line
```

The program writes out a series of files, so you should run this in an empty directory. The program will overwrite files (if it can), but won't delete other files already in the directory.

Run this program with this document as the standard input, such as:

```
./extractsample.pl <draft-ietf-smime-examples
```

If you want to extract without the program, copy all the lines between the ">" and "<" markers, remove any page breaks, and remove the "|" in the first column of each line. The result is a valid Base64 blob that can be processed by any Base64 decoder.

A.2. Example Extraction Program

```
#!/usr/bin/perl

# CMS Samples extraction program. v 1.1

# Get all the input as an array of lines
@AllIn = (); while (<STDIN>) { push(@AllIn, $_) }

$Base64Chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz' .
    'stuvwxyz0123456789+/' ;
$LineCount = 0; $CurrFile = '';

foreach $Line (@AllIn) {
```

```

$LineCount++; # Keep the line counter for error messages
$Line =~ s/^\s*//; # Get rid of leading whitespace
chomp($Line); # Get rid of CR or CRLF at the end of the line
if(substr($Line, 0, 1) ne '|') { next } # Not a special line
elsif(substr($Line, 1, 1) eq '*') { next } # It is a comment
elsif(substr($Line, 1, 1) eq '>')
    { &StartNewFile(substr($Line, 2)) } # Start a new file
elsif(substr($Line, 1, 1) eq '<')
    { &EndCurrFile(substr($Line, 2)) } # End the current file
else { &DoBase64(substr($Line, 1)) } # It is a line of Base64
}

sub StartNewFile {
    $TheNewFile = shift(@_);
    if($CurrFile ne '') { die "Was about to start a new file at " .
        "line $LineCount, but the old file, $CurrFile, was open\n" }
    open(OUT, ">$TheNewFile") or
        die "Could not open $TheNewFile for writing: $!\n";
    binmode(OUT); # This is needed for Windows, is a noop on Unix
    $CurrFile = $TheNewFile;
    $LeftOver = 0; # Amount left from previous Base64 character
    $NextPos = 0; # Bit position to start the next Base64 character
                  # (bits are numbered 01234567)
    $OutString = ''; # Holds the text going out to the file
}

sub EndCurrFile {
    $FileToEnd = shift(@_);
    if($CurrFile ne $FileToEnd) { die "Was about to close " .
        "$FileToEnd at line $LineCount, but that name didn't match " .
        "the name of the currently open file, $CurrFile\n" }
    print OUT $OutString;
    close(OUT);
    $CurrFile = '';
}

sub DoBase64 {
    $TheIn = shift(@_);
    if($CurrFile eq '') { die "Got some Base64 at line $LineCount, " .
        "but appear to not be writing to any particular file.\n" }
    @Chars = split(//, $TheIn); # Make an array of the characters
    foreach $ThisChar (@Chars) {
        # $ThisVal is the position in the string and the Base64 value
        $ThisVal = index($Base64Chars, $ThisChar);
        if($ThisVal == -1) { die "At line $LineCount, found the " .
            "character $ThisChar, which is not a Base64 character\n" }
        if($ThisVal == 64) { last } # It is a "=", so we're done
        if ($NextPos == 0 ) {

```

```

    # Don't output anything, just fill the left of $LeftOver
    $LeftOver = $ThisVal * 4;
    $NextPos = 6;
} elsif ($NextPos == 2) {
    # Add $ThisVal to $LeftOver, output, and reset
    $OutString .= chr($LeftOver + $ThisVal);
    $LeftOver = 0;
    $NextPos = 0;
} elsif ($NextPos == 4) {
    # Add upper 4 bits of $ThisVal to $LeftOver and output
    $Upper4 = ($ThisVal & 60);
    $OutString .= chr($LeftOver + ($Upper4/4));
    $LeftOver = (($ThisVal - $Upper4) * 64);
    $NextPos = 2;
} elsif ($NextPos == 6) {
    # Add upper 2 bits of $ThisVal to $LeftOver and output
    $Upper2 = ($ThisVal & 48);
    $OutString .= chr($LeftOver + ($Upper2/16));
    $LeftOver = (($ThisVal - $Upper2) * 16);
    $NextPos = 4;
} else { die "\$NextPos has an illegal value: $NextPos." }
}
}

```

B. Examples in Order of Appearance

From Section 2.1

ExContent.bin

```

| * Section 2.1
| >ExContent.bin
| VghpcyBpcyBzb21lIHhnbXBsZSBjb250ZW50Lg==
| <ExContent.bin

```

From Section 2.2

AlicePrivDSSSign.pri

```

| * Example AlicePrivDSSSign.pri
| >AlicePrivDSSSign.pri
| MIIBSwIBADCCASsGByqGSM44BAEwggeEeAoGBAIGNze2D6gqeOT7CSCij5EeT3Q7XqA7sU8
| WrhAhP/5Thc0h+DNbzREjR/p+vpKGJL+HZMMg23j+bv7dM3F9piuR10DcMkQiVm96nXvn8
| 9J8v3UOoi1TxP7AHCEdNXYjDw7Wz41UIddU5dhDEeL3/nbCElzf5FEbteQJllzzflvbAh
| UA4kemGkVmuBPG2o+4NyErYov3k80CgYAmONAUiTqOfs+bdllWWpMdiM5BAI1XPLLgJDD
| HlBd3ZtZ4s2qBTlywHuiNrhuB699ikIlp/Rlz0oIXks+kPht6pzJIYo7dhTpzi5dowfNI4
| W4LzABfG1JiRGJNks9+MiVSlNWteL5c+waYTYfEX/Cve3RUP+YdMLRgUpgObo2OQQAUA
| u0RG0aXJRgcu0P561pIH8JqFiT8=

```

```

|<AlicePrivDSSSign.pri
***AlicePrivRSASign.pri***

| * Example AlicePrivRSASign.pri
|>AlicePrivRSASign.pri
|MIICdgIBADANBgkqhkiG9w0BAQEFAASCAMAwggJcAgEAAoGBAOCJczmN2PX16Id2OX9OsA
|W7U4PeD7er3H3HdSkNBS5tEt+mhibU0m+qWCn8l+z6glEPMIC+sVCeRkTxLLvYMs/GaG8H
|2bBgrL7uNAlqE/X3BQWT3166NVbZYf8Zf8mB5vhs6odAcO+sbSx0ny36VTq5mXcCpkhSjE
|7zVzhXdFdfAgMBAAECgYAApAPDJ0d2NDRspoaleUkBSy6K0shissfXSAlqi5H3NvJ1luJN
|FZBgJzFHNWRNlclnY860nlasLzduHO4Ovygt9DmQbzTYbghblWVq2EHZE9ctOV7+M8v/Ke
|QDCz0Foo+38Y6idjewevfTLyvehwYifQRmXskbr4saw+yRRkt/IQJBAPbW4CIhTF8KcP8n
|/OWzUGqd5Q+lhZbGQPqoCrSbmwxVwgEd+TeCihTI8pMOKs2lZiG5PNIGv7RVMcncrcqYLd
|ECQQDo3rARJQnSALeB3oromFDld3dhpEWTawhVlnNd9MhbEpMic4t/03B/9aSqu3T9PCJq
|2jiRkoZbbBTorkye+o4vAkEAL0zwh5sXf+4bgxsUtgtqkF+GJlHht6B/9eSI4lm5+R6b0y
|l3OCJiIyKxJZi6PvltTt/oeILLIURYjdZNR56vN8QJALPAkW/qgzYUi6tBuT/pszsHTyOTx
|hERIZHPXKY9+RozsFd7kUbuOU5yyZLVVleyTqo2IfPmxNZ0ERO+G+6YMCgwJAWIjZoVA4hG
|qrA7y730v0Ng+4tCol+/bkBS9u4oiJIW9LJZ7Qq1CTyr9AcewhJcV/+wLpIza4M83ixpXu
|b4lfKA==
|<AlicePrivRSASign.pri

***BobPrivRSAEncrypt.pri***

| * Example BobPrivRSAEncrypt.pri
|>BobPrivRSAEncrypt.pri
|MIICChQIBADANBgkqhkiG9w0BAQEFAASCAMAwggJcAgEAAoGBAKnhZ5g/OdVf8qCTQV6meY
|mFyDVdmpFb+x0B2hlwJhcPvaUi0DWFbXqYZhRBXM+3twg7CcmRuBlpN235ZR572akzJKN/
|O7uvRgGGNjQyywcDWVL8hYsxBLjMGAGUSOZPHPTdYMTgXB9T039T2Gk8BQX4enDRvoPGXz
|jPHCyqaqfrAgMBAAECgYBnzUhMmg2PmMIbZf8ig5xt8KYGHbztptwOI1PIcaw+LNd40gngw
|y+e6alatd8brUXlweQqg9P5F4Kmy9Bnah5jWMIR05PxZbMHGd9ypkdb8MKCixQheIXFD/A
|0HPfD6bRSeTmPwF1h5HEuYHD09sBvf+iU7o8AsmAX2EAnYh9sDGQJBANDDisbeopkydo+N
|vKZl1mY/1I1FUox29XLE6/BGmve+XKpVC5va3Wtt+Pw7PAhDk7Vb/s7q/WiEI2Kv8zHCue
|UCQQDQufweIrd7bWOAcjXq/JY1PeClPNTqBlFy2bKKB1f4hAr84/sajB0+E0R9KfEILVH
|IdxJAfkkICnwJAiEYH2PAkA0umTJSChXNdVUN5qSO8bKlocSHseIVnDYDubl6nA7xhmqU
|5iUjiEzuUJiEiUacUGfJlaV/4jbOSnI3vQgLeFAkEAni+zN5r7CwZdV+EJBqRd2ZCWBgVf
|JAZAcpw6iIWchw+dYhKIFmionRobQ+g4wJhprwMKSDIETukPj3d9NDAlBwJAVxhn1grSta
|vCunrnVNqcBU+B108BiR4yPWnLMcRSyFRVJQA7HCp8JlDV6abXd8vPFfXuC9WN7rOvTKF8
|Y0ZB9qANMASGa1UdDzEEAwIAEA==
|<BobPrivRSAEncrypt.pri

***CarlPrivDSSSign.pri***

| * Example CarlPrivDSSSign.pri
|>CarlPrivDSSSign.pri
|MIIBSgIBADCCASsGBYqGSM44BAEwggEeAoGBALZJGD6KRMEpcZRMACQSwXp5y1RNqx6B+8
|ZMsw6UCQbrAdSxyHFLx0XAUCVdnPza5G3T4oZiHij9uhWVShb2Ru3d9pjSu36KCoq6Fnu5
|UAFIk4vrJRVRl1Xcj1MOEKlQ/HC3zTBu/dreqKoitaGvi8wCiOeLcF+5reEi1G0pLdbpAh
|UA3cEv31POCzRgdz4CpL+KXZi5ENUCgYAM7lebS73atgdqdDdPVX+d7bxhDetGWTxWCytb

```

```
|DJHOpWJSacrhbT69v/7ht7krYTyty65F4wasjCKdnESHc8fN8BzZtU5dc96vDskdWlH1T0
|R5NVpzqn9GUR+pQhacSOuKeWG01S9TikRjH4a4o1gGJfgpwO+64HXwQsRjZVKbCgQWAhQZ
|szilIWixUOV/uT4IRnjRPrXlcg==
|<CarlPrivDSSSign.pri
```

CarlPrivRSASign.pri

```
| * Example CarlPrivRSASign.pri
|>CarlPrivRSASign.pri
|MIICdgIBADANBgkqhkiG9w0BAQEFAASCAMAwggJcAgEAAoGBAORL/xi4Jff0d/9uc3uTcV
|y8MxqSknIj2EFGOMORogSzjq+Cnb1RHhd68nYsK4Y5p73XjRpT7OQA1ejsojax7eJQ4jIJ
|ij+fmSWPuE6ruX3VlmXaFqDFvg6uRFvvXvSnKcuC3axe6aqTlCkO+BjWyFde8nbE8hFgOL
|kbPB2XyWrxAgMBAAECgYEArnPkW19bZlrJl8bvOF9TISovYv7eKZp6hmc253lieHU9c6C8
|KQ7zj73Dycm2+LrWE5vDl3rKavC4hWVOD72nqPdUBkG969wgd5DfYZuab3Te6jvUnIdg7X
|aE8WowN9XgkBb4gEfDGWvtdXe6Su05t10CRztfG8gcq8vo9SY/pIECQD/3wmgVgtCUp7E
|TZOzsEm73ueBfSiZ0LFiugs54Rx7IhgztKD2v9yuHdChrQRxWmEKbjvOMNo2n2UlKbunDn
|8LAkEA5GloGF/5V9B8ZokPumMdcssgpIF2ZInNfdHCJ6kurHpWmoUH2TADowOrf4iSUCQB
|qhsHHyBMT8l7Vve2wn6rcwJAVzZsj4wEdmy2104kRAD4gOKvQgGpDxSE+Oca4I+MJ6QtX6
|LlbbVjwK1E6XaRpxlJLkb4d4VLO4cE8K/S2FQmlQJAZKEPrFV0G70NYXsXA82w5qcZHYCv
|8UFI2Bq2iBSgLRfdtQPDh96KrJuNwSrOUVzukaOD42CXyIUBc+io/N8gwJAJh4dHKGyK+
|Tb00hXbmtzGYhhOvp0SjaLR2hdUOsm4+p9m05lqa97q0sud1E9qNARq6PWqMANh1UC6qn
|0W2N+g==
|<CarlPrivRSASign.pri
```

DianePrivDSSSign.pri

```
| * Example DianePrivDSSSign.pri
|>DianePrivDSSSign.pri
|MIIBSwIBADCCASsGBYqGSM44BAEwgGEEAoGBALZJGD6KRMEpcZRMACQSwXp5y1RNqx6B+8
|ZMsw6UCQbrAdSxyHFLx0XAUCVdnPza5G3T4oZiHlIJ9uhWVShb2Ru3d9pjSu36KCoq6Fnu5
|UAFIk4vrJRVRl1XcjlMOEKlQ/HC3zTBU/dreqKoitaGvi8wCiOeLcF+5reEilG0pLdbpAh
|UA3cEv31POCzRgdz4CpL+KXZi5ENUCgYAM7lebS73atgdqddDPVX+d7bxbDetGWTxWCytb
|DJHOpWJSacrhbT69v/7ht7krYTyty65F4wasjCKdnESHc8fN8BzZtU5dc96vDskdWlH1T0
|R5NVpzqn9GUR+pQhacSOuKeWG01S9TikRjH4a4o1gGJfgpwO+64HXwQsRjZVKbCgQXAhUA
|lpX54MHgQS0yD4tCUpMq5h4OISk=
|<DianePrivDSSSign.pri
```

DianePrivRSASignEncrypt.pri

```
| * Example DianePrivRSASignEncrypt.pri
|>DianePrivRSASignEncrypt.pri
|MIICdwIBADANBgkqhkiG9w0BAQEFAASCAMAwggJdAgEAAoGBANb9uMBwxkwl7OrP6ny7om
|L68OYyOlP/sZJaF/Qg4ZkkgrQ9nz7RMqLJwbxfiYDqXadz+ygLHCW8oNC9tS3KAq7+L9K
|TBk/B9ugwWAet35n996xw2BJrEXX+MbvCDchk0fu8Hm1crACxPMRw15H5Qq3g/HbdG1ki0
|QdlV3NKMCFagMBAAECgYA9vc3CDmEUW0vnn2AjBCvFazW1lkUj/Gl9kzwP0yWWumJSQuKW
|z/5YgI/rsYy91Al10Dp3RSSeDOuGgM0sIRfxROOyqKkurBfSo4QlY7W8Lx7d9iH/FSAkW/
|GAL9VBDjIk99RKmp65SdgZjj85jWK9gPwMJJKT5MPXBZFTu5a2QQJBAP04P0rRlLCRYBNB
|kg2NRD93Hf+WIOQI1AtwyRqv6ZCU8rDVX08ZhVChkJGuvQV2UrMi2Kh8j1R/AHJPNNvoc7
```



```
|UCQQDh0ucRVwauCpUiFqoCtFrTtp2CEU+WPIbJEI1WezFleWnndWg4AEsu0iYy3bHi4CxU
|gAplutFmlhuwDqB+0ruRAkEAR7a82yJzQ0HstLVnqaGZ/O/Sjv0d++Upi/4K39TIXlclCl
|0r1AmgVlvFsWL8IL4ILeMhtaHns//EwKVfrBJcqQJBALmYQfwIUB9zYIoBonxSiiBa6iyJ
|2aUZ3ZTGG8MlwIJR5O4rmhncc+3pHSfU+GwD3asdCHulrH/pgpvxiYpx22ECQAEHIzdfem
|Co/VpcB9+o3vfisTR9/OuRvbBzdMjEvj9YRTAGkLOsacyz9z98rMe4G2WhFjk5sON0fc/N
|xaxsv+U=
|<DianePrivRSASignEncrypt.pri
```

From Section 2.3

AlicedSSSSignByCarlNoInherit.cer

```
|* Example AlicedSSSSignByCarlNoInherit.cer
|>AlicedSSSSignByCarlNoInherit.cer
|MIIC3DCCApugAwIBAgICAMgwCQYHKoZiZjgEAzASMRAwDgYDVQQDEwdDYXJsRFNTMB4XDT
|k5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1MwggG2
|MIIBKwYHKoZiZjgEATCCAR4CgYEAgY3N7YPqCp45PsJIKKPkR5PdDteoDuxTxauECE//lO
|FzSH4MlvNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny/dQ6iL
|VPE/sACir01diMPDtbPjvQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDiR6YaRW
|a4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjVc8ssaMMMeUF3dmlni
|zaofPVjAe6I2uG4Hr32KQiWn9HXPSgheS6Q+G3qnMkhiJt2FOnOLl2jB80jhbqvMAF8bU
|mJEYk2RL34yJVKU1a14vlz7BphNh8Rf8K97dFQ/5h0wtGBSmA5ujY5A4GEAAKBGfZjuVp1
|FJYLqXrd4z+p7Kxe3L23ExE0phaJKBEj2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09B41bY8
|i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTSljF2YUeyxDKE
|8H5BQP1Gp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADA0BgNVHQ8BAf8EBAMCBs
|AwHwYDVR0jBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVR0OBBYEF5sobPjwfft
|Q3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAKGBYqGSM
|44BAMDMAAwLQIUUVQyKGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4hBSW0Ne
|vQ==
|<AlicedSSSSignByCarlNoInherit.cer
```

AlicerSASignByCarl.cer

```
|* Example AlicerSASignByCarl.cer
|>AlicerSASignByCarl.cer
|MIICLDCCAZWgAwIBAgIQRjRrx4AAVrwR024uxBCzsDANBgkqhkiG9w0BAQUFADASMRAwDg
|YDVQQDEwdDYXJsUlnBMB4XDTk5MDkxOTAxMDg0N1oXDTM5MTIzMTIzNTk1OVowEzERMA8G
|A1UEAxMIQWxpY2VSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOCJczmN2PX16I
|d2OX9OsAW7U4PeD7er3H3HdSkNBS5tEt+mhibU0m+qWCn8l+z6glePMIC+sVCeRkTxLLvY
|Ms/GaG8H2bBgrL7uNalqE/X3BQWT3166NVbZyF8Zf8mB5vhs6odAcO+sbSx0ny36VTq5mX
|cCpkhSjE7zVzhXdfdfAgMBAAGjgYEWfzAMBGNVHRMBaf8EAjAAMA4GA1UdDwEB/wQEAwIG
|wDAfBgNVHSMEGDAWgBTp4JAnrHggeprTTPJCN04irp44uzAdBgNVHQ4EFgQUd9K00bdMio
|qjzkWdzuw8oDrj/1AwHwYDVR0RBBgwFoEUQWxpY2VSU0FAZXhhbXBsZS5jb2wDQYJKoZI
|hvcNAQEFBQADgYEAPnBHqEjMEliPylFxa042GF0EfoCxjU3MyqOPzH1WyLzPbrMcWakgqg
|WBqE4lradwFHUv9ceb0Q7pY9Jkt8ZmbnMhVN/0uiVdfUntlGsiNnRzuErsL2Tt0z3Sp0LF
|6DeKtNufZ+S9n/n+dO/q+e5jatg/SyUJtdgadq7rm9tJsCI=
|<AlicerSASignByCarl.cer
```

BobRSASignByCarl.cer

```
* Example BobRSASignByCarl.cer
>BobRSASignByCarl.cer
MIICJzCCAZCgAwIBAgIQRjRrx4AAVrwR024uzV1x0DANBgkqhkiG9w0BAQUFADASMRAwDg
YDVQQDEwdDYXJsU1NBMB4XDtk5MDkxOTAxMDkwMloXDTM5MTIzMTIzNTk1OVowETEPMA0G
A1UEAxMGQm9iU1NBMIgfMA0GCSqGSIB3DQEBAQUAA4GNADCBiQKBgQCp4WeYPznVX/Kgk0
FepnmJhcg1XZqRW/sdAdoZcCYXD72lItAlhW16mGYUQVzPt7cIOwnJkbgZaTdt+WUee9mp
MySjFzu7r0YBhjY0MssHA1lS/IWLMQS4zBgIFEjmTxz7XWDE4FwfU9N/U9hpAfEF+Hpw0b
6Dxl84zxwsqmqn6wIDAQABo38wftAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAwIFIDAf
BgNVHSMEGDAWgBTp4JAnrHggeprTTPJCN04irp44uzAdBgNVHQ4EFgQU6PS4Z9izlqQq8x
GqKdOVWoYwTCQwHQYDVR0RBBywFIESQm9iU1NBQGV4YW1wbGUuY29tMA0GCSqGSIB3DQEB
BQUAA4GBAHu0ZsXxED8QIEyIcat7QGshM/pKld6dDltrlCEFWPLhfirNnJOIh/uLt359QW
Hh5NZt+eIEVWFFvGqNRmChvVl52R1kPCHWRbBdaDOS6qzxV+WBfZjmNZGjOd539OgcOync
f1EHl/M28FAK3Zvetl44ESv7V+qJba3JiNiPzyvT
<BobRSASignByCarl.cer
```

CarlDSSSelf.cer

```
* Example CarlDSSSelf.cer
>CarlDSSSelf.cer
MIICmzCCAlqgAwIBAgIBATAJBgcqhkiG9w0BAQFADASMRAwDgYDVQQDEwdDYXJsU1NBMB4XDtk5MDkxOTAxMDkwMloXDTM5MTIzMTIzNTk1OVowETEPMA0G
A1UEAxMGQm9iU1NBMIgfMA0GCSqGSIB3DQEBAQUAA4GNADCBiQKBgQCp4WeYPznVX/Kgk0
FepnmJhcg1XZqRW/sdAdoZcCYXD72lItAlhW16mGYUQVzPt7cIOwnJkbgZaTdt+WUee9mp
MySjFzu7r0YBhjY0MssHA1lS/IWLMQS4zBgIFEjmTxz7XWDE4FwfU9N/U9hpAfEF+Hpw0b
6Dxl84zxwsqmqn6wIDAQABo38wftAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAwIFIDAf
BgNVHSMEGDAWgBTp4JAnrHggeprTTPJCN04irp44uzAdBgNVHQ4EFgQU6PS4Z9izlqQq8x
GqKdOVWoYwTCQwHQYDVR0RBBywFIESQm9iU1NBQGV4YW1wbGUuY29tMA0GCSqGSIB3DQEB
BQUAA4GBAHu0ZsXxED8QIEyIcat7QGshM/pKld6dDltrlCEFWPLhfirNnJOIh/uLt359QW
Hh5NZt+eIEVWFFvGqNRmChvVl52R1kPCHWRbBdaDOS6qzxV+WBfZjmNZGjOd539OgcOync
f1EHl/M28FAK3Zvetl44ESv7V+qJba3JiNiPzyvT
<CarlDSSSelf.cer
```

CarlRSASelf.cer

```
* Example CarlRSASelf.cer
>CarlRSASelf.cer
MIIB6zCCAVSgAwIBAgIQRjRrx4AAVrwR024un/JQIDANBgkqhkiG9w0BAQUFADASMRAwDg
YDVQQDEwdDYXJsU1NBMB4XDtk5MDgxODA3MDAwMFoXDTM5MTIzMTIzNTk1OVowEjEQMA4G
A1UEAxMHQ2FyYbFJTQTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwYkCgYEA5Ev/GLgkV/R3/2
5ze5NxXLwzGpKSciPYQUbQzRE6BL0Or4KdvVEeF3rydiwrhjmrvdeNGLps5ADV6OyiNrHt
4lDiMgmKP5+ZJY+4Tqu5fdWWZdoWoMW+Dq5EW+9e9Kcpy4LdrETpqpOUKQ74GNbIV17yds
TyEWA4uRs8HZfJavECAwEAAaNCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBByEFongkCeseCB6mtNM8kI3TiKunji7MA0GCSqGSIB3DQEBBQUAA4GBAL
eelATT7Snk/4mJFS5M2wzwsA8yYe7EBOWSXS3/D2RzfgrD7Rj941ZAN6cHtfA4EmFQ7e/d
```

```
|P+MLuGGlpJs85p6cVJq2ldbabDu1LUUlnUkBdvq5uTH5+WsSU6D1FGCbfco+8lNrsDdvre
|Z019v6WuoUQWNdz7IDSHaaolTNBgC
|<CarlRSASelf.cer
```

DianeDSSSignByCarlInherit.cer

```
|* Example DianeDSSSignByCarlInherit.cer
|>DianeDSSSignByCarlInherit.cer
|MIIBuDCCAXegAwIBAgICANIwCQYHKoZiZjgEAzASMRAwDgYDVQQDEwDYXJsRfNTMB4XDT
|k5MDgxNzAyMDgxMFOxDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIRGlhbmVEU1MwgZMw
|CQYHKoZiZjgEAQOBhQACgYEAoAAxeCzufoFTLi5hCA+hm1FSGtpZqHMvEiW2CMvK7ypEdo
|pSCeq9BSLVD/b9RtevmTgJDhPLTyzdHDT3HL81/yPT0lnngpc3vjEk2BjI80k5W7fi5Sd+
|/IxFclt+Po9oTdlGeik+jv/M2jKpoznlN0PpVcnXW6aBZ8zAqs0uxSOjgYEwfzAMBgNVHR
|MBAf8EAjAAMA4GA1UdDwEB/wQEAWIGwDafBgNVHSMEGDAWgBRwRD6CLm+H3krTdeM9ILxD
|K5PxHzAdBgNVHQ4EFgQUZDCZfVzcrQuZ01IvFr9YUN3OKxgWHwYDVR0RBbgwFoEURGlhbm
|VEU1NAZXhhbXBsZS5jb20wCQYHKoZiZjgEAwMwADAtAhUAOr4Fw4+XaiM9LZVMx5L4yys
|uV8CFChLEEVY0hydVTUUGJGyPznftW7T
|<DianeDSSSignByCarlInherit.cer
```

DianeRSASignByCarl.cer

```
|* Example DianeRSASignByCarl.cer
|>DianeRSASignByCarl.cer
|MIICLDCCAZWgAwIBAgIQRjRrx4AAVrwR024u1ZowkDANBgkqhkiG9w0BAQUFADASMRAwDg
|YDVQQDEwDYXJsUlnBMB4XDTk5MDgxOTA3MDAwMFOxDTM5MTIzMTIzNTk1OVowEzERMA8G
|A1UEAxMIRGlhbmVSU0EwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANb9uMBwxkw17O
|rP6ny7omL68OYyOlP/sZJaF/Qg4ZkkgrQ9nz7RMqLJwbxfiYDqXadz+ygLHCW8oNC9tS3
|KAq7+L9KTbk/B9ugwWAet35n996xw2BjREXX+MbvCDchk0fu8HMLcrACxPMRw15H5Qq3g/
|HbdG1ki0QdlV3NKMCFAGMBAAAGjgYEwfzAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAWIF
|4DAfBgNVHSMEGDAWgBTp4JAnrHggeprTTPJCN04irp44uzAdBgNVHQ4EFgQUjPPLdQ6NMf
|bUKdpEknW4/ulPOQwwHwYDVR0RBbgwFoEURGlhbmVSU0FAZXhhbXBsZS5jb20wDQYJKoZI
|hvcNAQEFBQADgYEAfaYstXhClnnzMf72QsoPEweSCRvgb7CRGPa/SvvMY3n7gb/dl8eQa8
|sKNytBagOYxRs+MshFK4YBnBziNu8WwRqSuL5i+1m+SUCxLnkKlimBoPwsqe7hX7VxtrO
|nHsxctei6kGrasDdh7kURBjPhFdm6MXmuNwtSx8bKEM2dXo=
|<DianeRSASignByCarl.cer
```

From Section 2.4

CarlDSSCRLForAll.crl

```
|* Example CarlDSSCRLForAll.crl
|>CarlDSSCRLForAll.crl
|MIHYMIGZMAkGBYqGSM44BAMwEjEQMA4GA1UEAxMHQ2FybERTUxcNOTkwODI3MDcwMDAwWj
|BpMBMCAGDIFw05OTA4MjIwNzAwMDBaMBMCAGDJFw05OTA4MjIwNzAwMDBaMBMCAGDTFw05
|OTA4MjIwNzAwMDBaMBMCAGDSFw05OTA4MjIwNzAwMDBaMBMCAGDUFw05OTA4MjQwNzAwMD
|BaMAkGBYqGSM44BAMDlwAwLAIUfmVSdjP+NHMX0feW+aDU2G1cft0CFAJ6W7fVWxjBz4fv
|ftok8yqDnDWh
|<CarlDSSCRLForAll.crl
```

CarlDSSCRLForCarl.crl

```
* Example CarlDSSCRLForCarl.crl
>CarlDSSCRLForCarl.crl
MIGDMEQwCQYHKOZIZjgEAzASMRAdgYDVQQDEwdDYXJsRFNFTFw05OTA4MjUwNzAwMDBaMB
QwEgIBARcNOTkwODIyMDcwMDAwWjAJBgcqhkJ00AQDAzAAMC0CFQCzH8VPej3sdtVg+d55
IuxPsJD+lwIUWovDhLxmhu/eYJbCl0H9rqpBSk=
<CarlDSSCRLForCarl.crl
```

CarlDSSCRLEmpty.crl

```
* Example CarlDSSCRLEmpty.crl
>CarlDSSCRLEmpty.crl
MG0wLjAJBgcqhkJ00AQDMBIxEDA0BgNVBAMTB0NhcmxEU1MXDTk5MDgyMDA3MDAwMFowCQ
YHKOZIZjgEAwMwADAtAhRiPzYXNVGuZ1B59Q1LjK3Ua/RknwIVALU7TqFMe/0Pw42btv7D
XW/eZSh9
<CarlDSSCRLEmpty.crl
```

CarlRSACRLForAll.crl

```
* Example CarlRSACRLForAll.crl
>CarlRSACRLForAll.crl
MIIBMzCBnTANBgkqhkiG9w0BAQQFADASMRAdgYDVQQDEwdDYXJsU1NBFW05OTA4MjcwNz
AwMDBaMGkwIQIQRjRrx4AAVrwR024uxBCzsBcNOTkwODIyMDcwMDAwWjAhAhBGNGvHgABW
vBHTbi7VmJCQFw05OTA4MjUwNzAwMDBaMCECEYY0a8eAAFa8EdNuLs1dcdAXDTk5MDgyND
A3MDAwMFowDQYJKoZIhvcNAQEEBQADgYEA7OXqlPwMiEWK3eSemu7l8jc6vH6ZhYwDrWe
XPCBlF6zbsG1a4zUXSVN+0deZvNdq+W0GDZgqE2cPInsbye/NVBxgcK5RftiiRkSMal7mt
PMZssR2QsQR3etTyLZ5X8w8lv8lFGlWHY7H6hGph/2od5Voe0xiGmXDwjTlAxgWx4=
<CarlRSACRLForAll.crl
```

CarlRSACRLForCarl.crl

```
* Example CarlRSACRLForCarl.crl
>CarlRSACRLForCarl.crl
MIHsMFcwDQYJKoZIhvcNAQEEBQAWEjEQMA4GA1UEAxMHQ2FybFJTQRcNOTkwODI1MDcwMD
AwWjAjbCECEYY0a8eAAFa8EdNuLp/yUCAXDTk5MDgyMjA3MDAwMFowDQYJKoZIhvcNAQEE
BQADgYEAIE8hlMEahZVJa8pFYtZXCf+pUS6O2UcY+vjlct1P7XR04/NlMmUoLJodV+XVJg
bqleYjlYSNDome7psML84H96PRa4VMD//m3fzczXMsHn3csHHFTPwBblJXaR45Y98SIjDH
ElWUBW4qAKlBxCpmlGLONjPCK2NHJZ3z3nDuAFY=
<CarlRSACRLForCarl.crl
```

CarlRSACRLEmpty.crl

```
* Example CarlRSACRLEmpty.crl
>CarlRSACRLEmpty.crl
MIHHMDIwDQYJKoZIhvcNAQEEBQAWEjEQMA4GA1UEAxMHQ2FybFJTQRcNOTkwODIwMDcwMD
AwWjANBgkqhkiG9w0BAQQFAAOBgQCpxSG4E3x087UR7ATzIEWGHgtuf4NtX/Q0dgZZJQ4E
PYgJiIE3xNwgmPoXgQs3lKy0j3tRiRSky3JzFAe8IpxAoQf8RHyFDwui0e7hDq/2FnStoa
```

```
| /BAHUAZOqlmvYLCKLb1Rlfpqe5OUU1Cg72XoTn+LlayRjCDriglr6B0oBtyQ==
| <CarlRSACRLEmpty.crl
```

Rest of the sections

3.1.bin

```
| * Example 3.1.bin
| >3.1.bin
| MIAGCSqGS1b3DQEHAaCAJIAEBFRoaXMEGCBpcyBzb21lIHNhbXBsZSBjb250ZW50LgAAAA
| AAAA==
| <3.1.bin
```

3.2.bin

```
| * Example 3.2.bin
| >3.2.bin
| MCsGCSqGS1b3DQEHAaAeBBxUaGlzIGlziHNvbWUgc2FtcGx1IGNvbnRlbnQu
| <3.2.bin
```

4.1.bin

```
| * Example 4.1.bin
| >4.1.bin
| MIIDlwYJKoZIhvcNAQcCoIIDiDCCA4QCAQExCTAHBgUrDgMCGjArBgkqhkiG9w0BBwGgHg
| QcVGhpcyBpcyBzb21lIHNhbXBsZSBjb250ZW50LQCCAUAwggLcMIIICm6ADAgECAgIAyDAJ
| BgcqhkiG9w0AQDMBIXEDAOBgNVBAMTB0NhcmxUeU1MwHhcNOTkwODE3MDExMDQ5WhcNMzkxMj
| MxMjM1OTU5WjATMREwDwYDVQQDEwhBbG1jZURTUzCCAbYwggErBgcqhkiG9w0AQBMIIIBHgKB
| gQCBjc3tg+oKnjk+wkgo+RHk90016g07FPFq4QIT/+U4XNIfgzW80RI0f6fr6ShiS/h2T
| DINT4/m7+3TNxfayrkddA3DJE1lZvep175/PSfL91DqItU8T+wBwhHTV2Iw801s+NVCHXV
| OXYQxHi9/52whJc38uRRG7XkCZzc835b2wIVA0JHphfZrgTtxtqPuDchK2KL95PNAoGAJj
| jQFIkyqjn7Pm3ZS1lqTHYjOQQCNVzyyxowwx5QXd2bWeLNqgU9WMB7oja4bgevfYpCJaf0
| dc9KCF5LPpD4beqcySGKO3YU6c4uXaMHZSOFuC8wAXxtSYkRiTZEvfjIlUpTVrXi+XPsgm
| E2HxF/wr3t0VD/mHTC0YFKYDm6NjkdGyQAAoGAXOO5WnUULgupet3jp6nsrF7cvbcTETSm
| FokoESPZNIzndXUTEj1DW2/1Ub/6ifKiGz4kft0HjVtjyLtFpaBK44XWzgaAP+gjfhryJK
| tTGrgnDR7vCL9mFIBcYqxl+hWL8bs01NKWN/Zhr7LEMoTwfkFA/UanY04z8qXi9PKD5biJ
| gYEWfzAMBgNVHRMBAf8EAjAAMA4GA1UdDwEB/wQEAwIGwDAfBgNVHSMEGDAWgBRwRD6CLm
| +H3krTdeM9ILxdk5PxHzAdBgNVHQ4EFgQUvmyhs+PB9+1DcKTOEwHi/eOX/s0wHwYDVR0R
| BBgwFoEUQWxpY2VEU1NAZXhhbXBsZS5jb20wCQYHkoZiZjgEAWMwADAtAhRVDKQZH0IriX
| EiM42DarU9Z2u/RQIVAJ9hU1JUC1yy3drndh3iEFJbQ169MWMwYQIBATAYMBIXEDAOBgNV
| BAMTB0NhcmxUeU1MCAgDIMAcGBSsOAwIaMAkGBYqGSM44BAMELjAsAhQJkf7r0mn1GLfXzV
| X0geoqQmqtAwIUOgfMwyG+4RpLfz61Ddu6HOq8zyk=
| <4.1.bin
```

4.2.bin

```
* Example 4.2.bin
>4.2.bin
MIIDUgYJKoZIhvcNAQcCoIIDQzCCAz8CAQEXCzAJBgUrDgMCGGUAMCScGCSqGSIb3DQEHAA
AeBBxUaGlzIGlziHNvbWUgc2FtcGxlIGNvbmlbnQuoIICMDCCAIwwggGVoAMCAQICEEY0
a8eAAFa8EdNuLsQQs7AwDQYJKoZIhvcNAQEFBQAWEjEQMA4GA1UEAxMHQ2FybFJTQTAEFw
05OTA5MTkwMTA4NDdaFw0zOTEyMzEyMzU5NTlaMBMxETAPBgNVBAMTCEFSaWNlUlNBMIgf
MA0GCSqGSIb3DQEBQUAA4GNADCBiQKBgQDgiXM5jdj19eiHdj1/TrAFulOD3g+3q9x9x3
UpDQUubRLfpoYm1NJvqlgp/Jfs+oJRDzCAvrFQnkZE8Sy72DLpxmhvB9mwYKy+7jQJahPl
9wUFk99eujVW2WH/GX/Jgeb4bOqHQHDvrG0sdJ8t+1U6uZ13AqZIUoxO81c4V3RXXwIDAQ
ABo4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8EBAMCBsAwHwYDVR0jBBgwFoAU6eCQ
J6x4IHqa00zyQjdOIq6eOLswHQYDVR0OBBYEFHfStNG3TIqKo85Fnc7sPKA64/9QMB8GA1
UdEQQYMBaBFEFsaWNlUlNBQGV4YW1wbGUuY29tMA0GCSqGSIb3DQEBBQUAA4GBAD5wr6hI
zBNYj8pRcWtONhhdBH6AsY1NzMqjj8x9Vsi8z26zHFmpIKoFgahOJa2ncBR1L/XHm9EO6W
PSZLfGZm5zIVTf9LolXX1J05RrIjZ0c7hK7C9k7dM90qdCzeg3irTbn2fkvZ/5/nTv6vnu
Y2rYP0slCbXYGnau65vbSbAiMYHLMiHIAgEBMCYwEjEQMA4GA1UEAxMHQ2FybFJTQQIQRj
Rrx4AAVrwr024uxBCzsDAJBgUrDgMCGGUAMA0GCSqGSIb3DQEBQUAAIGALyOC0vMJX7gM
W0tOnb+JmohldcSRPdPQ1Xu21f6UoYqs48SE9c1gTiev9s8AhnZ1Pyvw59QCZ6f1x40WBK
WztefZMvAk7+cgrNWfB8VTJPrOAR0PFxOnKpWdK+QDlRQL6TkNus5unJ4M6JjmVRPUaG/Q
B9eisWJM44+v/eDVXcc=
<4.2.bin
```

4.3.bin

```
* Example 4.3.bin
>4.3.bin
MIIDdwYJKoZIhvcNAQcCoIIDAaDCCA2QCAQEXCTAHBgUrDgMCGjALBqkqhkiG9w0BBwGggg
LgMIIC3DCCApugAwIBAgICAMgwCQYHKOZiZjgEAAZASMRAwDgYDVQQDEwdDYXJsRFNTMB4X
DTk5MDgxNzAxMTA0OV0XDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1Mwgg
G2MIIBKwYHKOZiZjgEATCCAR4CgYEAgY3N7YPqCp45PsJIKKPkR5PdDteoDuxTxauECE//
lOFzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny/dQ6
iLVPE/sAcIR0ldiMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDiR6Ya
RWa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjVc8ssaMMMeUF3dm1
nizaoFPVjAe6I2uG4Hr32KQiWn9HXPSgheS6Q+G3qnMkhijt2FOnOLl2jB80jhbgvMAF8
bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSMA5ujY5A4GEAAKBgfzjuV
p1fJYLqXrd4z+p7Kxe3L23ExE0phaJKBEj2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09B41b
Y8i7RaWgSuOF1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGksZfoVi/G7NNTSljf2YUeyxD
KE8H5BQP1Gp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8EBAMC
BsAwHwYDVR0jBBgwFoAUcEQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVR0OBBYEF5sobPjwf
ftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAKGBYqG
SM44BAMDMAAwLQIUUVQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVatcst3a53Yd4hBSW0
NevTFjMGECAQEwGDASMRawDgYDVQQDEwdDYXJsRFNTAgIAyDAHBgUrDgMCGjAJBgqhkiG9
OAQDBC4wLAIUBvvHKiTVNIn3i7X9cySlhsgPWmwCFGZpGbxoWNGNsZ1SP9oUiA39yaG4
<4.3.bin
```

4.4.bin

* Example 4.4.bin

>4.4.bin

```

MIILDQYJKoZIhvcNAQcCoIIK/ jCCCvoCAQEXCTAHBgUrDgMCGjArBgkqhkiG9w0BBwGgHg
QcVGhpcyBpcyBzb21lIHhnbXBsZSBjb250ZW50LQCCB68wggIsMIIBlaADAgECAhBGNGvH
gABWvBHTbi7EELowMA0GCSqGSIB3DQEBBQUAMBIxEDAObgNVBAMTB0NhcmxSU0EwHhcNOT
kwOTE5MDEwODQ3WhcNMzkxMjMxMjM1OTU5WjATMREwDwYDVQQDEwhBbGljZVJTQTCBnzAN
BgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEA4IlzOY3Y9fXoh3Y5f06wBbtTg94Pt6vcfcld1KQ
0FLm0S36aGJtTSb6pYKfyX7PqCUQ8wgL6xUJ5GRPEsu9gyz8ZobwfZsGCsvu40CWOT9fcF
BZPfXrolVtlh/xl/yYHm+Gzqh0Bw76xtLHSfLfpVOrmZdwKmSFKMTvNXOfd0V18CAwEAAa
OBgTB/MAWGA1UdEwEB/wQCMAAwDgYDVROPAQH/BAQDAgBAMB8GA1UdIwQYMBaAFongkCes
eCB6mtNM8ki3TiKunji7MB0GA1UdDgQWBRR30rTrt0yKiQPORZ307DygoUp/UDAFBgNVHR
EEGDAWgRRBbGljZVJTQUBleGftcGx1LmNvbTANBgkqhkiG9w0BAQUFAAOBgQA+cEeoSMwT
WI/KUXFrTjYYXQR+gLNTczKo4/MfVbIvM9usxxZqSCqBYGoTiWtp3AUDS/1x5vRDulj0m
S3xmZucyFU3/S6JV19SdOUayI2dH04SuwvZO3TPdKnQsXoN4q0259n5L2f+f507+r57mNq
2D9LJQm12Bp2ruub20mwIjCCApwggJaoAMCAQICAQEwCQYHkoZiZjgEAzASMRawDYDVQ
QDEwdDYXJSrFNTMB4XDTk5MDgxNjIyNTA1MFoXDTM5MTIzMTIzNTk1OVowEjEQMA4GA1UE
AxMHQ2FybERTUzCCAbcwggErBgCqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
PjwfftQ3CkzhMB4v3jl/7NMB8GA1UdEQQYMBaBFEFsaWNlRFNTQGV4YW1wbGUuY29tMAkG
ByqGSM44BAMDMAAwLQIUvQykGR9CK4lxIjONg2q1PWdrv0UCFQCfYVNSVAtcst3a53Yd4h
BSW0NevaGB2zCB2DCBmTAJBgcqhkiG9w0BAQgBMBIIBHGBgQC2SRg+ikTBKXGUTAHEESF6ec
tUTasegfvGTLMOlAkG6wHUSchxS8dFwFALXZz82uRt0+KGSISCfboVlUoW9kbt3faY0rt+
igqKuhZ7uVABSJOL6yUVUZdV3I9TDhCpUPxwt80wVP3a3qiqIrWhr4vMAojni3Bfua3hCN
RtKS3W6QIVAN3BL99Tzgs0YHc+AqS/il2YuRDVAoGADO5Xm0u92rYHanQ3T1V/ne28YQ3r
Rlk8VgsrWwyRzqvUmnK4W0+vb/+4be5K2E8rcuuReMGrIwinZxEhwvHzfAc2bVOXXPerw
7JHVP9R9UEeTVac6p/RlEfquIWN EjrinlhtNuvUyJYEx+GuKNYBiX4KcDvuuB18ELEY2VS
mwoDgYUAAoGBAJmHdCdZqCwK3cLHW74WxEnNohbU1HbbFiCenYrh7yOrSusaOOeptxTg
CUybQlTrlg1hkkAfnIDP51wPvO2GgA4/3Vce/fI5YZBpT0sWGPOlexCBGkCyYl8FJ2geoL
Yg2VKuaGanKyp1CDC6onzRupTYma140YOYQ/i8VWTYB6o0IwQDAPBgNVHRMBAf8EBTADAQ
H/MA4GA1UdEwEB/wQEAwIBhjAdBgNVHQ4EFgQUCeQ+gi5vh95K03XjPSC8QyuT8R8wCQYH
KoZiZjgEAwMwADAtAhRrqfBOelp54/m+PSvJBjfpERehEwIVA180aSqLSTwDeZQyTRIfzo
n7RrI7MIIC3DCCApugAWIBAgICAMgwCQYHkoZiZjgEAzASMRawDYDVQQDEwdDYXJSrFNT
MB4XDTk5MDgxNzAxMTA0OVoXDTM5MTIzMTIzNTk1OVowEzERMA8GA1UEAxMIQWxpY2VEU1
MwggG2MIIBKwYHkoZiZjgEATCCAR4CgYEAgy3N7YPqCp45PsJIKKPkR5PdDteoDuxTxaue
CE//lOfzSH4M1vNESNH+n6+koYkv4dkwyDbeP5u/t0zcX2mK5HXQNwyRCJWb3qde+fz0ny
/dQ6iLVPE/sAcIR01diMPDtbPjVQh11Tl2EMR4vf+dsISXN/LkURu15AmWXPn+W9sCFQDi
R6YaRwa4E8baj7g3Istii/eTzQKBgCY40BSJMqo5+z5t2UtZakx2IzKEAjvC8ssaMMMeUF
3dm1nizaoFPVjAe6I2uG4Hr32KQIwn9HXPSgheS6Q+G3qnMkhi jt2FOnOLl2jB80jhbgv
MAF8bUmJEYk2RL34yJVKUla14vlz7BphNh8Rf8K97dFQ/5h0wtGBSma5ujY5A4GEAAKBgF
zjuVplFJYLqXrd4z+p7Kxe3L23ExE0phaJKB Ej2TSGZ3V1ExI9Q1tv5VG/+onyohs+JH09
B41bY8i7RaWgSuOf1s4GgD/oI34a8iSrUxq4Jw0e7wi/ZhSAXGKSzfoVi/G7NNTS1jF2YU
eyxDKE8H5BQPlGp2NOM/Kl4vTyg+W4o4GBMH8wDAYDVR0TAQH/BAIwADAObgNVHQ8BAf8E
BAMCBsAwHwYDVROjBBgwFoAUceQ+gi5vh95K03XjPSC8QyuT8R8wHQYDVRO0BBYEFL5sob
Pjw
```

4.5.bin

[Page 128]

4.7.bin

[Page 129]

4.8.eml

[Page 130]

```
|T0FRREJDNDhMQULVTS9tR2Y2Z2sKZ3A5WjBYdFJkR2ltSmVCL0J4VUNGR0ZGSnF3WVJ0MV
|dZY0lPUW9HaWFvd3FHelZJCgotLS0tLS09X05leHRCb3VuZHZJ5Xl9fX0ZyaSxfMDZfU2Vw
|XzIwMDJfMDA6MjU6MjEtLQo=
|<4.8.eml
```

4.9.eml

```
|* Example 4.9.eml
|>4.9.eml
|TULNRS1WZXJzaW9uOiAxLjAKVG86IFVzZXIyQGv4YWlwbGVzLmNvbQpGcm9tOiBhbG1jZU
|Rzc0BleGFtcGxlcY5jb20KU3ViamVjdDogRXhhbXBsZSA0LjkKTWVzc2FnZS1JZDogPDAY
|MTAzMTE2NDU0MDMwMz4zMdRAZxhxbXBsZXMuY29tPgpeYXRlOiBUaHUsIDMxIE9jdCAyMD
|AyIDE2OjQ1OjE0IC0wMzAwIApDb250ZW50LVR5cGU6IGFwcGxpY2F0aW9uL3BrY3M3LW1p
|bWU7IHNtaW1lLXR5cGU9c2lnbmVklWRhdGE7CiAgICBuYW1lPWNtaW1lLnA3bQpDb250ZW
|50LVRyYW5zZmVyLUVuY29kaW5nOiBiYXNlNjQKQ29udGVudC1EaXNwb3NpdGlvbjogYXR0
|YWNobWVudDsgZmlsZW5hbWU9c2lpbWUucDdtCgpNSUleBVfZSktvWklodmNOQVFjQ29JSU
|RpakNDQTRZQ0FRRXhDVEFIQmdVckRnTUNHakF0QmdrcWhraUc5dzBCQndHZ0lBUWVEUXBV
|CmFhbkpJR2x6SUhOdmJXVWdjMkZ0Y0d4bElHTnZib1JsYm5RdW9JSUM0RENDQXR3d2dnS2
|JvQUlDQVFjQ0FnRElNQWtHQnlxR1NNNDQKQkFNd0VqRVFNQTRHQTFRVUF4TUhRMkZ5YkVS
|VFV6QWVGdzA1T1RBNE1UY3dnVEV3TkR5YUz3MHPVEV5TXpFeU16VTVOVGxhTUJNeApFVE
|FQQmdOVkZBTVRDRUZZYVd0bFJGTlRNSUlcGpDQ0FTc0dCeXFHU000NEJBRXdnZ0VlQW9H
|QkFJR056ZTJENmdxZU9UN0NTQ2lqcjVfZVZQZUTdYcUE3c1U4V3JoQWwQLzVUaGMwaCtETm
|J6UkVqUi9wK3ZwS0dKTctlWklNZZIzaitidjdKTTNGOXBpdVixMERjTWtRaVYKbTk2blh2
|bjg5Sjh2M1VPb2kxVHhQN0FIQ0VktlhzakR3Nld6NDVSWRkVTVkaERFZUwzL25iQ0Vsem
|Z5NUZFYnRlUUpSbHp6Zmx2YgpBaFVBNGtlbUdrVm1lQlBHMm8rNE55RXJZb3YzazgwQ2dZ
|QWlPTkFVaVRLcU9mcytiZGxMVldwTWRpTTVCQUkxWFBMTEDqRERlBEJkCjNadFo0czJxQl
|QxWXJdIdWlOcmhlQjY5OWlrSWxwL1IxejBvSVhrcytrUGh0NnB6Sk1ZbzdkaFRwemk1ZG93
|Zk5JNfc0THpBQmZHMUoKaVJHsk5rUzkrTWlWU2x0V3RlTDVjK3dhWVRZZkVYL0N2ZTNSVV
|ArWWRNTFJnVXBnt2JvMk9RT0JoQUFDZ1lCYzQ3bGFkUlNXQzZsNgozZU0vcWV5c1h0eTl0
|eElSTktzV2lTz1JJOwswaG1kMWRSTVNQVU5iYitWUnYvcUo4cUliUGlSOVBRZU5Xm1BJdT
|BXbGZFcmpoZGZlPCkVjvQ3ZJa3ExtWF1Q2NOSHU4SXyYwVWVnRnhpckdYNkZzdnh1
|elRVMHBZMzltRkhzc1F5aFBCK1FVRDlScWRqVGPQeXBlTDAKOG9QbHVLTP0JnVEIvTUF3R0
|ExVVRFD0VCL3dRQ01BQXdEZ1lEVlIiwUEFRSC9CQVFEQWdiQU1COEdBMVVkSXdrWU1CYUFG
|SEJFUG9JdQpInGZlU3ROMTR6MGd2RUlyay9FZk1CMEdBMVVkRGdRV0JCuytiS0d6NDhIMz
|dVTndwTTRUQUVMOTQ1Zit6VEFmQmdOVkhSRUVHREFXCmduSUKjiR2xqWlVSFVUwQmxlR0Z0
|Y0d4bExtTnZiVEFKQmdjcWhrak9PQVFEQXpBU1DMENGRlVNCeJrZlFpdUpjU0l6alloCX
|RUMW4KYtc5RkFoVUFuMkZUVWxRTFhMTGQydWQySGVJUUVsdERYcjb4WXpCaEFnRUJNQmd3
|RWpFUU1BNEdBMVVFQXhNSFEyRnliRVJUUVxdJQWpBTWd3QndZRkt3NERBaG93QlFZSEtVWk
|l6amdFQXdRdU1Dd0NGRDFjUlc2TElVRnplWGx1M1lJNVNLU0Jlci9zQWhRbUNxN3MvQ1RG
|CkhPRWpnQVnlVWp1TXB4NWc2QT09Cg==
|<4.9.eml
```

4.10.bin

```
|* Example 4.10.bin
|>4.10.bin
|MIIH/wYJKoZIhvcNAQcCoIIH8DCCB+wCAQEXCTAHBgUrDgMCGjArBgkqhkiG9w0BBwGgHg
|QcVGhpcyBpcyBzb21lIHNhbXBsZSBjb250ZW50LqCCAUAwggLcMIICm6ADAgECAgIAyDAJ
```

```

|Bgcqhkj0OAQDMBIXEDA0BgNVBAMTB0NhcmxU1MwHhcNOTkwODE3MDExMDQ5WhcNMzKxMj
|MxMjM1OTU5WjATMREwDwYDVQQDEwhBbGljZURTUzCCAbYwggErBgcqhkj0OAQBMIIbHgKB
|gQCBjc3tg+oKnjk+wkgo+RHk90016gO7FPFq4QIT/+U4XNIfgzW80RI0f6fr6ShiS/h2T
|DINT4/m7+3TNxfayrkddA3DJE1lZvep175/PSfL91DqItU8T+wBwhHTV2Iw80ls+NVCHXV
|OXYQxHi9/52whJc38uRRG7XkCZzc835b2wIVA0JHphpFZrgTtxtqPuDchK2KL95PNAoGAJj
|jQFIkyqjn7Pm3ZS1lqTHYjOQQCNVzyyxowwx5QXd2bWeLNqgU9WMB7oja4bgevfYpCJaf0
|dc9KCF5LPpD4beqcySGKO3YU6c4uXaMHZSOFuC8wAXxtSYkRiTZEvfjIlUpTVrXi+XPsGm
|E2HxF/wr3t0VD/mHTC0YFKYDm6NjkdGyQAAoGAXO05WnUULgupet3jP6nsrF7cvbcTETSm
|FokoESPZNIzndXUTEj1DW2/lUb/6ifKiGz4kft0HjVtjyLtFpaBK44XWzgaAP+gjfhryJK
|tTGrgnDR7vCL9mFIBcYqxl+hWL8bs0lNKWN/Zhr7LEMoTfwkFA/UanY04z8qXi9PKD5bi j
|gYEWfzAMBgNVHRMBAF8EAjAAMA4GA1UdDwEB/wQEAWIGwDAfBgNVHSMEGDAWgBRwRD6CLm
|+H3krTdeM9ILxDK5PxHzAdBgNVHQ4EFgQUvmyhs+PB9+1DcKTOEwhi/eOX/s0wHwYDVR0R
|BBgwFoEUQWxpY2VEU1NAZXhhbXBsZS5jb20wCQYHKoZiZjgEAWMwADAtAhRVDKQZH0IriX
|Eim42DarU9Z2u/RQIVAJ9hU1JUC1yy3drndh3iEFJbQ169MYIEyTCCBMUCAQEwGDASMRaw
|DgYDVQQDEwDYXJsRFNTAgIAyDAHBgUrDgMCGqCCBF8wGAYJKoZIhvcNAQkDMQsGCsGSI
|b3DQEHAATAjBgkqhkiG9w0BCQQxfgQUQGrscFJ5um4Wai2eBinAIpaH3UgWOAYDKqsZMTEE
|L1RoAXMgaXMGYSB0ZXN0IEdlbmVyYWwgQVNOIEF0dHJpYnV0ZSwgbnVtYmVyIDEuMDQ5
|qGSib3DQEJEAEIEMS8wLQwgQ29udGVudCBiaW50cyBEZXNjcmlwdGlvbiBCdWZmZXIGCSqG
|Sib3DQEHAATBkgkqhkiG9w0BCQ8xPTA7MACGBSoDBAUGMDAGBiODBAUGTQQmU21pbWUgQ2
|FwYUWJpbGl0aWVzIHhcmFtZXRLcnMgYnVmZmVYIDUwYU9wYU9wYU9wYU9wYU9wYU9wYU9w
|BgcqAwQFBgcIEExtUSElTIEElTIEEGUFJjVjVjVjVjVjVjVjVjVjVjVjVjVjVjVjVjVjVjVj
|EjEYFUSElTIEElTIEEGVEVTVCBTRUNVUklUWS1DQVRFR09SWS4wbwYLKoZIhvcNAQkQAgox
|YDBegUqAwQFBgQrQ29udGVudCBSZWZlcmVUy2UgQ29udGVudCBjZGVudG1maWVzIEJlZm
|ZlcgQoQ29udGVudCBSZWZlcmVUy2UgU2lnbmF0dXJlIFZhbHVlIEJlZmZlcjBzBgsqhkiG
|9w0BCRACCzFkoGIwWjELMAkGALUEBhMCMVVMxjAUBgNVBAoTDVVTIEEdvdmVybml1bnQxET
|APBgNVBAsTCFZEQSBTAxRlMQwwCgYDVQQLEWNNWREExEjAQBgNVBAMTCURhaXN5IFJTTQIE
|ClVEMzCB/AYLKOZIhvcNAQkQAgoMxgewwgekwgeYEBZU3MzgyOTkyDzE5OTkwMzExMTA0ND
|MzWqGBYTCBxqRhMF8xCzAJBgNVBAYTA1VTMRYwFAYDVQQKEw1VUyBHb3Zlcm5tZW50MREw
|DwYDVQQLEWhWREEGU210ZTEMMAAoGALUECXMdVkrBMRcwFQYDVQQDEw5CdWdzIEJlZm55IE
|RTQARhMF8xCzAJBgNVBAYTA1VTMRYwFAYDVQQKEw1VUyBHb3Zlcm5tZW50MREwDwYDVQQLE
|WhWREEGU210ZTEMMAAoGALUECXMdVkrBMRcwFQYDVQQDEw5FbGl1ciBGdWRKIERETQCCAQ
|IGCyqGSib3DQEJEAEIjMYHyMIHvMXICAQEGBYoDBAUGBwkTJkVRVU1WQUxFTlQgVEhJUyBJ
|UyBBIFBSSVZBQ1kgTUFSSyBURVNUMTwwOoAIKMEBQYHhnhLhMsRVFVSVZBTEVOVCBUSE
|lTIElTIEEGVEVTVCBTRUNVUklUWS1DQVRFR09SWS4xeQIBAQYHKgMEBQYHChMTRVFSVZB
|TEVOVCBUSElTIEElTIEEGU0VDT05EIFBSSVZBQ1kgTUFSSyBURVNUMTwwOoAIKMEBQYHhnh
|ihLhMsRVFVSVZBTEVOVCBUSElTIEElTIEEGVEVTVCBTRUNVUklUWS1DQVRFR09SWS4wCQYH
|KoZiZjgEAWQvMC0CFQC8MzdlxPdwxBDJE6pMhcq7UpFIWQIUy5aifIvPV96wSF9sZN2EBE
|lfHMo=
|<4.10.bin

```

4.11.bin

```

| * Example 4.11.bin
| >4.11.bin
|MIIGiAYJKoZIhvcNAQcCoIIIGeTCCBnUCAQEADALBgkqhkiG9w0BBWggggV/MIIcMzCCAl
|qgAwIBAgIBATAJBgcqhkj0OAQDMBIXEDA0BgNVBAMTB0NhcmxU1MwHhcNOTkwODE2MjI1
|MDUwWhcNMzKxMjMxMjM1OTU5WjASMRawDgYDVQQDEwDYXJsRFNTMIIBtzCCASsGBYqGSM
|44BAEWggEeAoGBALZJGD6KRMEpcZRMACQSwXp5y1RNqx6B+8ZMsw6UCQbrAdSxyHFLx0XA

```

```

|UCVdnPza5G3T4oZiHj9uhWVShb2Ru3d9pjSu36KCoq6Fnu5UAFIk4vrJVRl1Xcj1MOEK
|lQ/HC3zTBU/dreqKoitaGvi8wCiOeLcF+5reEI1G0pLdbpAhUA3cEv31POCzRgdz4CpL+K
|XZi5ENUCgYAM7lebS73atgdqddDpVX+d7bxhDetGWTxWCYtbDJHOpWJSacrhbT69v/7ht7
|krYTyty65F4wasjCKdnESHc8fN8BzZtU5dc96vDskdWlH1T0R5NVpzn9GUR+pQhacSOuK
|eWG01S9TIkrjH4a4o1gGJfgpwO+64HXwQsRjZVKbCgOBhQACgYEAmyd0JwNmOLHARDwsdb
|vhbESC2iFtTUdtsWIJ6diuHvI6tJSxo456m3FOAJTJtCVOUWCWGSQB82IM/nXA+87YaADj
|/dVwT98j1hkG1PSxYY86V7EIEaQLJiXwUnaB6gtiDZUq5oa6crKnUIMLqifNG6lNiZrXjR
|g5hD+LxVZNghqjQjBAMA8GA1UdEwEB/wQFMAMBAf8wDgYDVR0PAQH/BAQDAgGMB0GA1Ud
|DgQWBBRwRD6CLm+H3krTdem9ILxDK5PxHzAJBgcqhkJ00AQDazAAMC0CFGup8E56Wnnj+b
|49K8kGN+kRF6ETAhUAjzRpKouxPAN51DJNEH/OiftGsjswggLcMIICm6ADAgECAGIAyDAJ
|BgcqhkJ00AQDMBIxEDA0BgNVBAMTB0NhcmeU1MwHhcNOTkwODE3MDExMDQ5WhcNMzkxMj
|MxMjM1OTU5WjATMREwDwYDVQQDEwBbG1jZURTUzCCABYwggErBgcqhkJ00AQBMIIbHKB
|gQCBjc3tg+oKnjk+wkgo+RHk90016g07FPFq4QIT/+U4XNIfgzW80RI0f6fr6ShiS/h2T
|DINT4/m7+3TNxfayrkddA3DJEilZvepl75/PSfL91DqItU8T+wBwhHTV2Iw80Is+NVCHXV
|OXYQxHi9/52whJc38uRRG7XkCZzc835b2wIVA0JHphpfZrgTtxtqPuDchK2KL95PNAoGAJj
|jQFIkyqjn7Pm3ZS1lqTHYjOQQCNVzyyxowwx5QXd2bWeLNqgU9WMB7oja4bgevfYpCJaf0
|dc9KCF5LPpD4beqcySGK03YU6c4uXaMHZSOFuC8wAXxtSYkRiTZEvfjilUpTVrXi+XPsGm
|E2HxF/wr3t0VD/mHTC0YFKYDm6NjkdGyQAAoGAX005WnUULgupet3jp6nsrF7cvbcTETSm
|FokoESPZNIzndXUTEj1DW2/1Ub/6ifKiGz4kft0HjVtjyLtFpaBK44XWzgaAP+gjfhryJK
|tTGrgrnDR7vCL9mFIBcYqxl+hWL8bs01NKNW/Zhr7LEMoTfwkFA/Uany04z8qXi9PKD5biJ
|gYEWfzAMBgNVHRMBaf8EAjAAMA4GA1UdDwEB/wQEAwIGwDAfBgNVHSMEGDAWgBRwRD6CLm
|+H3krTdem9ILxDK5PxHzAdBgNVHQ4EFgQUvmyhs+PB9+1DcKTOEwHi/eOX/s0wHwYDVR0R
|BBgwFoEUQWxpY2VEU1NAZXhhbXBsZS5jb20wCQYHKoZIzjgEAwMwADAtAhRVDKQZH0IriX
|EiM42DarU9Z2u/RQIVAJ9hU1JUC1yy3drndh3iEFJbQ169oYHbMIHYMIGZMAkGBYqGSM44
|BAMWejEQMA4GA1UEAxMHQ2FybERTUxcNOTkwODI3MDcwMDAwWjBpMBMCAgDIFw05OTA4Mj
|IwNzAwMDBaMBMCAgDJFw05OTA4MjIwNzAwMDBaMBMCAgDTFw05OTA4MjIwNzAwMDBaMBMC
|AgDSFw05OTA4MjIwNzAwMDBaMBMCAgDUFw05OTA4MjQwNzAwMDBaMAkGBYqGSM44BAMDLw
|AwLAIUfmVSdjP+NHMX0feW+aDU2G1cfT0CFAJ6W7fVWxjBz4fvftok8yqDnDWhMQA=
|<4.11.bin

```

5.1.bin

```

| * Example 5.1.bin
|>5.1.bin
|MIIBHgyJKoZIhvcNAQcDoIIBDzCCAQsCAQAxcAwgb0CAQAwJjASMRAdGyYDVQQDEwdDYX
|JsUlnBAhBGNGvHgABWvBHTbi7NXXHQMA0GCSqGSIb3DQEBAQUABIGAC3EN5nGIiJi2lsGP
|cP2iJ97a4e8kbKQz36zg6Z2i0yx6zYC4mZ7mX7FBs3IWg+f6KgCLx3M1eCbWx8+MDFbbpX
|adCDgO8/nUkUNYeNxJtuzubGgzoyEd8Ch4H/dd9gdzTd+taTEgS0ipdSJUNnkVY4/M652j
|KKHRLff02hosdR8wQwYJKoZIhvcNAQcBMBQGCCqGSIb3DQMHBAGtaMXprWZRNYAgDsif8
|Z9P43LrY4OxUk660cu1lXecSFOSOpOJ7FuVyU=
|<5.1.bin

```

5.2.bin

```
* Example 5.2.bin
>5.2.bin
MIIBZQYJKoZIhvcNAQcDoIIBVjCCAVICAQIxggEAMIG9AgEAMCYweJEQMA4GA1UEAxMHQ2
FybFJTQQIQRjRrx4AAVrwR024uzV1x0DANBgkqhkiG9w0BAQEFAASBgJQmQo jGi7Z4IP+C
VypBmNFoCDoEp87khtgyff2N4SmqD3RXPx+8hbLQt9i3YcMwcap+aiOkyqjMalT03VUC0X
BOGv+HYI3HBZm/aFzxoq+YOWXAWs5xlGerZwTOc9j6AYlK4qXvnztR5SQ8TBjlzytm4V7zg
+TGrnGVNQBNw47Ewoj4CAQQwDQQLTWfPbExpc3RSQzIwEAYLkoZIhvcNAQkQAwwCAToEGH
cUr5MSJ/g9HnJVHsQ6X56VcwYb+OfOjTBJBgkqhkiG9w0BBwEwGgYIKoZIhvcNAwIwDgIC
AKAECJwE0hkuKlWhgCBekNXhojuej3org9Lt7n+wWxOhnky5V50vSpoYRfRRyw==
<5.2.bin
```

5.3.eml

```
* Example 5.3.eml
>5.3.eml
TULNRS1WZXJzaW9uOiAxLjAKTWVzc2FnZS1JZDogPDAwMTAzMTEyMDA1MjAzLjAwMzQ5QG
FteWVtaWx5LmlnLmNvbT4KRGF0ZTogVHVlLCAzMSPY3QgMjAwMCAxMjowMDolMiAtMDYw
MCAoQ2VudHJhbCBTdGFuZGFyZCBUaW1lKQpGcm9tOiBVc2VyMQpUbzogVXNlcjIKU3Viam
VjdDogRXhhbXBsZSA1LjMKQ29udGVudC1UeXB1OiBhcHBsaWNhdGlvbi9wa2NzNy1taW1l
OwoJbmFtZT1zbWltZS5wN207CglzbWltZS10eXB1PWVudmVsb3BlZC1kYXRhCkNvbnRlbn
QtVHJhbnNmZXItRW5jb2Rpbmc6IGJhc2U2NApDb250ZW50LURpc3Bvc2l0aW9uOiBhdHRh
Y2htZW50OyBmaWxlbmFtZT1zbWltZS5wN20KCK1JSUJIZ1lKS29aSWh2Y05BUWNEb01JQk
R6Q0NBUXNDQVFBGdJQXdnYjBDQVFBd0pqQVNNUkF3RGdZRFZRUURFd2REWVhKc1VsTkIK
QWhCR05HdkhnQUJXdkJIVGJpN05YWEhRTUEwR0NtcUdTSWIZRFFQkFRVUFCSudBQzNFTj
VuR0lpSmkybHhHUGNQMmlKOTdhNGU4awpiS1F6MzZ6ZzZaMmkweXg2ellDNGLaN21YN0ZC
czNJV2crZjZLZ0NMeDNNMWVDYld4OCTNREZiYnBYWWRDRGdPOC9uVWtVTl1lTnhKCnRlen
Vir2d6b3lFZDhDaDRIL2RkOWdke1RkK3RhVEVnUzBpcGRtSnVObmtWWTQvTTY1MmpLS0hS
TEZmMDJob3NkUjh3UXdzSkvtWkkKaHZjTkFRY0JNQ1FHQ0Nxr1NJYjNEUUIQkFndGFNWH
BSdlpST1lBZ0RzaVNmOFo5UDQzTHJZNE94VWs2NjBjdTFsWGVDU0ZPU09wTwPKN0Z1VnlV
PQoK
<5.3.eml
```

6.0.bin

```
* Example 6.0.bin
>6.0.bin
MF4GCSqSIB3DQEHBaBRME8CAQAwBwYFKw4DAhOWKwYJKoZIhvcNAQcBoB4EHFRoaXMgaX
Mgc29tZSBzYW1wbGUy29udGVudC4EFEBq7AhSebpuFgItngYpwCKWh91I
<6.0.bin
```

7.1.bin

```
| * Example 7.1.bin
| >7.1.bin
| MFcGCSqGSIB3DQEHBqBKMEgCAQAwQwYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAiza2v7Yj
| EIToAg+vzt2z8YFx04iRHqNNYg2/TD2VgV75M7mvXXBPalcOI=
| <7.1.bin
```

7.2.bin

```
| * Example 7.2.bin
| >7.2.bin
| MIGVBgkqhkiG9w0BBwaggYcwgYQCAQIwQwYJKoZIhvcNAQcBMBQGCCqGSIB3DQMHBAgHJy
| CFkJ6wfoAg0iCPZ0iKy0HkImhdvncFUibt4wG9AJFYpzVuvEuiBzOhOjA4BgMqqzMxMQQv
| VGhpcyBpcyBhIHRlc3QgR2VuZXJhbCBBU04gQXR0cmliZXRLLCBudWliZXIgcMS4=
| <7.2.bin
```

C. Acknowledgements

Blake Ramsdell, Jim Schaad, and John Pawling contributed the vast majority of the examples in this document, and/or correct examples during the early versions of this document. Additional examples came from many people, including Rob Colestock and Paul Hoffman. Additional testing came from Holger Ebel and Russ Housley.

The examples are displayed with a modified version of Peter Gutmann's "dumpasn1" program. Peter and Jim Schaad and Blake Ramsdell have been updating the program based on input from the process of writing this draft.

Editor's Address

Paul Hoffman
Internet Mail Consortium
127 Segre Place
Santa Cruz, CA 95060 USA

EMail: phoffman@imc.org

Full Copyright Statement

Copyright (C) The Internet Society (2005).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

