X.509 数字证书

15331191 廖颖泓

一、实例

RSA Self-Signed Certificate

```
  0  574: SEQUENCE {
  4  423:   SEQUENCE {
  8    3:     [0] {
 10    1:       INTEGER 2
       :       }
 13    1:     INTEGER 17
 16   13:     SEQUENCE {
 18    9:       OBJECT IDENTIFIER
       :         sha1withRSAEncryption (1 2 840 113549 1 1 5)
 29    0:       NULL
       :       }
 31   67:     SEQUENCE {
 33   19:       SET {
 35   17:         SEQUENCE {
 37   10:           OBJECT IDENTIFIER
       :             domainComponent (0 9 2342 19200300 100 1 25)
 49    3:           IA5String 'com'
       :           }
       :         }
 54   23:       SET {
 56   21:         SEQUENCE {
 58   10:           OBJECT IDENTIFIER
       :             domainComponent (0 9 2342 19200300 100 1 25)
 70    7:           IA5String 'example'
       :           }
       :         }
 79   19:       SET {
 81   17:         SEQUENCE {
 83    3:           OBJECT IDENTIFIER commonName (2 5 4 3)
 88   10:           PrintableString 'Example CA'
       :           }
       :         }
       :       }
100   30:     SEQUENCE {
102   13:       UTCTime 30/04/2004 14:25:34 GMT
117   13:       UTCTime 30/04/2005 14:25:34 GMT
       :       }
132   67:     SEQUENCE {
134   19:       SET {
```

```
136   17:           SEQUENCE {
138   10:             OBJECT IDENTIFIER
      :               domainComponent (0 9 2342 19200300 100 1 25)
150    3:             IA5String 'com'
      :             }
      :           }
155   23:         SET {
157   21:           SEQUENCE {
159   10:             OBJECT IDENTIFIER
      :               domainComponent (0 9 2342 19200300 100 1 25)
171    7:             IA5String 'example'
      :             }
      :           }
180   19:         SET {
182   17:           SEQUENCE {
184    3:             OBJECT IDENTIFIER commonName (2 5 4 3)
189   10:             PrintableString 'Example CA'
      :             }
      :           }
      :         }
201  159:       SEQUENCE {
204   13:         SEQUENCE {
206    9:           OBJECT IDENTIFIER
      :             rsaEncryption (1 2 840 113549 1 1 1)
217    0:           NULL
      :           }
219  141:         BIT STRING, encapsulates {
223  137:           SEQUENCE {
226  129:             INTEGER
      :               00 C2 D7 97 6D 28 70 AA 5B CF 23 2E 80 70 39 EE
      :               DB 6F D5 2D D5 6A 4F 7A 34 2D F9 22 72 47 70 1D
      :               EF 80 E9 CA 30 8C 00 C4 9A 6E 5B 45 B4 6E A5 E6
      :               6C 94 0D FA 91 E9 40 FC 25 9D C7 B7 68 19 56 8F
      :               11 70 6A D7 F1 C9 11 4F 3A 7E 3F 99 8D 6E 76 A5
      :               74 5F 5E A4 55 53 E5 C7 68 36 53 C7 1D 3B 12 A6
      :               85 FE BD 6E A1 CA DF 35 50 AC 08 D7 B9 B4 7E 5C
      :               FE E2 A3 2C D1 23 84 AA 98 C0 9B 66 18 9A 68 47
      :               E9
358    3:             INTEGER 65537
      :             }
      :           }
      :         }
363   66:       [3] {
365   64:         SEQUENCE {
367   29:           SEQUENCE {
369    3:             OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
```

```
374   22:              OCTET STRING, encapsulates {
376   20:                OCTET STRING
        :                  08 68 AF 85 33 C8 39 4A 7A F8 82 93 8E 70 6A 4A
        :                  20 84 2C 32
        :                }
        :              }
398   14:            SEQUENCE {
400    3:              OBJECT IDENTIFIER keyUsage (2 5 29 15)
405    1:              BOOLEAN TRUE
408    4:              OCTET STRING, encapsulates {
410    2:                BIT STRING 1 unused bits
        :                  '0000011'B
        :                }
        :              }
414   15:            SEQUENCE {
416    3:              OBJECT IDENTIFIER basicConstraints (2 5 29 19)
421    1:              BOOLEAN TRUE
424    5:              OCTET STRING, encapsulates {
426    3:                SEQUENCE {
428    1:                  BOOLEAN TRUE
        :                  }
        :                }
        :              }
        :            }
        :          }
431   13:      SEQUENCE {
433    9:        OBJECT IDENTIFIER
        :            sha1withRSAEncryption (1 2 840 113549 1 1 5)
444    0:        NULL
        :        }
446  129:      BIT STRING
        :        6C F8 02 74 A6 61 E2 64 04 A6 54 0C 6C 72 13 AD
        :        3C 47 FB F6 65 13 A9 85 90 33 EA 76 A3 26 D9 FC
        :        D1 0E 15 5F 28 B7 EF 93 BF 3C F3 E2 3E 7C B9 52
        :        FC 16 6E 29 AA E1 F4 7A 6F D5 7F EF B3 95 CA F3
        :        66 88 83 4E A1 35 45 84 CB BC 9B B8 C8 AD C5 5E
        :        46 D9 0B 0E 8D 80 E1 33 2B DC BE 2B 92 7E 4A 43
        :        A9 6A EF 8A 63 61 B3 6E 47 38 BE E8 0D A3 67 5D
        :        F3 FA 91 81 3C 92 BB C5 5F 25 25 EB 7C E7 D8 A1
        :      }
```

二、实例工作原理分析

实例中的证书摘自 RFC 5280 Appendix C.1，属于 RSA 证书，包含 578 个字节，证书版本号为 3。该证书包含以下信息：

(a) 证书序列号是 17；
(b) 证书使用的 RSA 和 SHA-1 哈希算法签名；
(c) 证书发行者的名字是 cn=Example CA,dc=example,dc=com;
(d) 证书主体的名字是 cn=Example CA,dc=example,dc=com;
(e) 证书发行时间是 2004 年 4 月 30 日，过期时间是 2005 年 4 月 30 日；
(f) 证书包含一个 1024 位的 RSA 公钥；
(g) 证书包含一个使用了 160 位 SHA-1 哈希算法生成的使用者密钥标识符扩展；
(h) 证书是一个 CA 证书。

为了更好地理解证书的结构和工作原理，我从 RFC 5280 上摘录了一部分必要的结构知识以及在网络上搜索到相关的数据结构知识。

(1) X.509 数据证书的编码

X.509 证书的结构是用 ASN.1(Abstract Syntax Notation One)进行描述数据结构，并使用 ASN1 语法进行编码。

ASN.1 采用一个个的数据块来描述整个数据结构，每个数据块都有四个部分组成：

1、数据块数据类型标志(一个字节，即 8 位)

数据类型包括**简单类型**和**结构类型**。

简单类型是不能再分解类型，如整型(INTERGER)、比特串(BIT STRING)、字节串(OCTET STRING)、对象标示符(OBJECT IDENTIFIER)、日期型(UTCTime)等。

结构类型是由简单类型和结构类型组合而成的，如顺序类型(SEQUENCE, SEQUENCE OF)、选择类型（CHOICE）、集合类型（SET）等

- 顺序类型的数据块值由按给定顺序成员成员数据块值按照顺序组成；
- 选择类型的数据块值由多个成员数据数据块类型中选择一个的数据块值；
- 集合数据块类型由成员数据块类型的一个或多个值构成。

这个标识字节的结构如下
1.1 bit8-bit7

用来标示 TAG 类型，共有四种，分别是 universal(00)、application(01)、context-specific(10)和 private(11)。

这两位为 universal（00）时，bit5-bit1 的值表示不同的 universal 的值：

标记（TAG） 对应类型 （黑色加粗为证书中出现的类型）

[UNIVERSAL 1] BOOLEAN [有两个值:false 或 true]

[UNIVERSAL 2] INTEGER [整型值]

[UNIVERSAL 3] BIT STRING [0 位或多位]

[UNIVERSAL 4] OCTET STRING [0 字节或多字节]

[UNIVERSAL 5] NULL

[UNIVERSAL 6] OBJECT IDENTIFIER [相应于一个对象的独特标识数字]

[UNIVERSAL 7] OBJECT DESCRIPTOR [一个对象的简称]

[UNIVERSAL 8] EXTERNAL, INSTANCE OF [ASN.1 没有定义的数据类型]

[UNIVERSAL 9] REAL [实数值]

[UNIVERSAL 10] ENUMERATED [数值列表，这些数据每个都有独特的标识符，作为 ASN.1 定义数据类型的一部分]

[UNIVERSAL 12] UTF8String

[UNIVERSAL 13] RELATIVE-OID

[UNIVERSAL 16] SEQUENCE, SEQUENCE OF [有序数列，SEQUENCE 里面的每个数值都可以是不同类型的，而 SEQUENCE OF 里是 0 个或多个类型相同的数据]

[UNIVERSAL 17] SET, SET OF [无序数列，SET 里面的每个数值都可以是不同类型的，而 SET OF 里是 0 个或多个类型相同的数据]

[UNIVERSAL 18] NumericString [0－9 以及空格]

[UNIVERSAL 19] PrintableString [A-Z、a-z、0-9、空格以及符号'()+,-./:=?]

[UNIVERSAL 20] TeletexString, T61String

[UNIVERSAL 21] VideotexString

[UNIVERSAL 22] IA5String

[UNIVERSAL 23] UTCTime [统一全球时间格式]

[UNIVERSAL 24] GeneralizedTime

[UNIVERSAL 25] GraphicString

[UNIVERSAL 26] VisibleString, ISO646String

[UNIVERSAL 27] GeneralString

[UNIVERSAL 28] UniversalString

[UNIVERSAL 29] CHARACTER STRING

[UNIVERSAL 30] BMPString

[UNIVERSAL 31]... reserved for future use

这两位为 context-specific（10）时，bit5-bit1 的值表示特殊内容：

[0] –- 表示证书的版本

[1] –- issuerUniqueID,表示证书发行者的唯一 id

[2] –- subjectUniqueID,表示证书主体的唯一 id

[3] –- 表示证书的扩展字段

1.2 bit6

表示是否为结构类型(1 位结构类型)；0 则表明编码类型是简单类型。

1.3 bit5-bit1

是类型的 TAG 值。根据 bit8-bit7 的不同值有不同的含义，具体含义见上面的描述。

如 SEQUENCE 类型数据块，其 TAG 类型位 UNIVERSAL（00）,属于结构类型（1），TAG 值为 16（10000）所以其类型标示字段值为（00110000），即为 0x30。

再如，证书扩展字段类型的数据块，TAG 类型为（10），属结构类型（1），TAG 的值为 3（00011），所以其类型标示字段值为（10100011），即为 0xA3。

2、数据块长度

长度字段，有两种编码格式。

若长度值小于等于 127，则用一个字节表示，bit8=0, bit7-bit1 存放长度值；

若长度值大于 127，则用多个字节表示，可以有 2 到 127 个字节。第一个字节的第 8 位为 1，其它低 7 位给出后面该域使用的字节的数量，从该域第二个字节开始给出数据的长度，高位优先。

还有一种特殊情况，这个字节为 0x80，表示数据块长度不定，由数据块结束标识结束数据块。

3、数据块的值

存放数据块的值，具体编码随数据块类型不同而不同。

(2) X.509 数据证书的结构

1、X.509 数据证书基本结构

1.1. 版本号.

标识证书的版本（版本 1、版本 2 或是版本 3）。

1.2. 序列号

标识证书的唯一整数，由证书颁发者分配的本证书的唯一标识符。

1.3. 签名

用于签证书的算法标识，由对象标识符加上相关的参数组成，用于说明本证书所用的数字签名算法。例如，SHA-1 和 RSA 的对象标识符就用来说明该数字签名是利用 RSA 对 SHA-1 杂凑加密。

1.4. 颁发者

证书颁发者的可识别名（DN）。

1.5. 有效期

证书有效期的时间段。本字段由”Not Before”和”Not After”两项组成，它们分别由 UTC 时间或一般的时间表示（在 RFC2459 中有详细的时间表示规则）。

1.6. 主体

证书拥有者的可识别名，这个字段必须是非空的，除非你在证书扩展中有别名。

1.7. 主体公钥信息

主体的公钥（以及算法标识符）。

1.8. 颁发者唯一标识符

标识符—证书颁发者的唯一标识符，仅在版本 2 和版本 3 中有要求，属于可选项。

1.9. 主体唯一标识符

证书拥有者的唯一标识符，仅在版本 2 和版本 3 中有要求，属于可选项。

2、X.509 数据证书扩展部分

可选的标准和专用的扩展（仅在版本 2 和版本 3 中使用），扩展部分的元素都有这样的结构：

```
Extension ::= SEQUENCE {
 extnID      OBJECT IDENTIFIER,
 critical    BOOLEAN DEFAULT FALSE,
 extnValue   OCTET STRING }
```

extnID：表示一个扩展元素的 OID

critical：表示这个扩展元素是否极重要

extnValue：表示这个扩展元素的值，字符串类型。

扩展部分包括：

2.1. 发行者密钥标识符

证书所含密钥的唯一标识符，用来区分同一证书拥有者的多对密钥。

2.2. 密钥使用

一个比特串，指明（限定）证书的公钥可以完成的功能或服务，如：证书签名、数据加密等。

如果某一证书将 KeyUsage 扩展标记为“极重要”，而且设置为“keyCertSign”，则在 SSL 通信期间该证书出现时将被拒绝，因为该证书扩展表示相关私钥应只用于签写证书，而不应该用于 SSL。

2.3. CRL 分布点

指明 CRL 的分布地点。

2.4. 私钥的使用期

指明证书中与公钥相联系的私钥的使用期限，它也有 Not Before 和 Not After 组成。若此项不存在时，公私钥的使用期是一样的。

2.5. 证书策略

由对象标识符和限定符组成，这些对象标识符说明证书的颁发和使用策略有关。

2.6. 策略映射

表明两个 CA 域之间的一个或多个策略对象标识符的等价关系，仅在 CA 证书里存在。

2.7. 主体别名

指出证书拥有者的别名，如电子邮件地址、IP 地址等，别名是和 DN 绑定在一起的。

2.8. 颁发者别名

指出证书颁发者的别名，如电子邮件地址、IP 地址等，但颁发者的 DN 必须出现在证书的颁发者字段。

2.9. 主体目录属性

指出证书拥有者的一系列属性。可以使用这一项来传递访问控制信息。

(3) X.509 数据证书详细描述

```
Certificate ::= SEQUENCE {
    tbsCertificate       TBSCertificate, --  证书主体
    signatureAlgorithm    AlgorithmIdentifier, --  证书签名算法标识
    signatureValue       BIT STRING --证书签名值,是使用 signatureAlgorithm 部分指定的签名算法对
                                          tbsCertificate 证书主题部分签名后的值.
    }
  TBSCertificate ::= SEQUENCE {
    version         [0] EXPLICIT Version DEFAULT v1, --  证书版本号
    serialNumber         CertificateSerialNumber, --  证书序列号，对同一 CA 所颁发的证书，序列号唯一标
                                                      识证书
    signature            AlgorithmIdentifier, --证书签名算法标识
    issuer               Name,              --证书发行者名称
    validity             Validity,          --证书有效期
    subject              Name,              --证书主体名称
    subjectPublicKeyInfo SubjectPublicKeyInfo,--证书公钥
    issuerUniqueID [1] IMPLICIT UniqueIdentifier OPTIONAL,
                  --  证书发行者 ID(可选)，只在证书版本 2、3 中才有
    subjectUniqueID [2] IMPLICIT UniqueIdentifier OPTIONAL,
                  --  证书主体 ID(可选)，只在证书版本 2、3 中才有
    extensions      [3] EXPLICIT Extensions OPTIONAL
                  --  证书扩展段（可选），只在证书版本 3 中才有
    }
  Version ::= INTEGER { v1(0), v2(1), v3(2) }
  CertificateSerialNumber ::= INTEGER

  AlgorithmIdentifier ::= SEQUENCE {
    algorithm               OBJECT IDENTIFIER,
    parameters               ANY DEFINED BY algorithm OPTIONAL }
```

```
parameters:
Dss-Parms ::= SEQUENCE { -- parameters ， DSA(DSS)算法时的 parameters,RSA 算法没有此参数
    p             INTEGER,
    q             INTEGER,
    g             INTEGER }

 signatureValue：
 Dss-Sig-Value ::= SEQUENCE { -- sha1DSA 签名算法时,签名值
                 r       INTEGER,
               s       INTEGER }

Name ::= CHOICE {
 RDNSequence }
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::=
 SET OF AttributeTypeAndValue
AttributeTypeAndValue ::= SEQUENCE {
 type      AttributeType,
 value     AttributeValue }
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY DEFINED BY AttributeType

Validity ::= SEQUENCE {
    notBefore      Time,   --  证书有效期起始时间
    notAfter       Time  --  证书有效期终止时间
    }
Time ::= CHOICE {
    utcTime        UTCTime,
    generalTime    GeneralizedTime }
UniqueIdentifier ::= BIT STRING
SubjectPublicKeyInfo ::= SEQUENCE {
    algorithm           AlgorithmIdentifier, --  公钥算法
    subjectPublicKey    BIT STRING          --  公钥值
    }
 subjectPublicKey:
 RSAPublicKey ::= SEQUENCE { -- RSA 算法时的公钥值
     modulus           INTEGER, -- n
     publicExponent    INTEGER -- e -- }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
Extension ::= SEQUENCE {
    extnID      OBJECT IDENTIFIER,
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING }
```

根据以上三部分知识我们对实例进行具体分析。

```
  0   574: SEQUENCE {          // Certificate:: SEQUENCE 类型，长度为 574
  4   423:   SEQUENCE {       // tbsCertificate:: SEQUENCE 类型，长度为 423
  8     3:     [0] {           // Version:: 证书版本，长度为 3
 10     1:       INTEGER 2 // 整形类型，长度为 1，证书版本为 3(2)
               :       }
 13     1:     INTEGER 17 // SerialNumber::整形类型，长度为 1，序列号为 17
 16    13:     SEQUENCE {// Signature:: EQUENCE 类型，签名，长度为 13
 18     9:       OBJECT IDENTIFIER// Signature:: OBJECT IDENTIFIER 类型，签名，长度为 9
               :         sha1withRSAEncryption (1 2 840 113549 1 1 5) // 表示 sha1 和 RSA 算法
 29     0:       NULL    // Signature:: NULL 类型，长度为 0
               :       }
 31    67:     SEQUENCE {                // 以下红色的数据块表示发行者 issuer 信息
 33    19:       SET {
 35    17:         SEQUENCE {            // 发行者 issuer 域名 domainComponent
 37    10:           OBJECT IDENTIFIER
               :             domainComponent (0 9 2342 19200300 100 1 25)
 49     3:           IA5String 'com'
               :           }
               :       }
 54    23:       SET {
 56    21:         SEQUENCE {            // 发行者 issuer 域名 domainComponent
 58    10:           OBJECT IDENTIFIER
               :             domainComponent (0 9 2342 19200300 100 1 25)
 70     7:           IA5String 'example'
               :           }
               :       }
 79    19:       SET {
 81    17:         SEQUENCE {            // 发行者 issuer 名称 common name
 83     3:           OBJECT IDENTIFIER commonName (2 5 4 3)
 88    10:           PrintableString 'Example CA'
               :           }
               :       }
               :     }
100    30:     SEQUENCE {          // validity:: SEQUENCE 类型,长度 30
102    13:       UTCTime 30/04/2004 14:25:34 GMT // notBefore:: UTCTime 类型，证书有效期起始时间
，长度 13
117    13:       UTCTime 30/04/2005 14:25:34 GMT // notBefore:: UTCTime 类型，证书有效期结束时间
，长度 13
               :       }
132    67:     SEQUENCE {                // 以下红色的数据块表示使用者 subject 信息
134    19:       SET {
136    17:         SEQUENCE {            // 使用者 subject 域名 domainComponent
138    10:           OBJECT IDENTIFIER
               :             domainComponent (0 9 2342 19200300 100 1 25)
150     3:           IA5String 'com'
```

```
      :             }
      :           }
  155  23:         SET {
  157  21:           SEQUENCE {          // 使用者 subject 域名 domainComponent
  159  10:             OBJECT IDENTIFIER
      :               domainComponent (0 9 2342 19200300 100 1 25)
  171   7:             IA5String 'example'
      :             }
      :           }
  180  19:         SET {
  182  17:           SEQUENCE {          // 使用者 subject 名称 commonName
  184   3:             OBJECT IDENTIFIER commonName (2 5 4 3)
  189  10:             PrintableString 'Example CA'
      :             }
      :           }
      :         }
  201 159:       SEQUENCE {     // subjectPublicKeyInfo:: SEQUENCE 类型,长度 159
  204  13:         SEQUENCE {
  206   9:           OBJECT IDENTIFIER
      :             rsaEncryption (1 2 840 113549 1 1 1) //  使用 RSA 加密算法
  217   0:           NULL
      :           }
  219 141:         BIT STRING, encapsulates {    // subjectPublicKey::公钥值，BIT STRING 类型
  223 137:           SEQUENCE {
  226 129:             INTEGER       //  subjectPublicKey::公钥值，integer 类型, 128 字节, 1024 位
      :               00 C2 D7 97 6D 28 70 AA 5B CF 23 2E 80 70 39 EE
      :               DB 6F D5 2D D5 6A 4F 7A 34 2D F9 22 72 47 70 1D
      :               EF 80 E9 CA 30 8C 00 C4 9A 6E 5B 45 B4 6E A5 E6
      :               6C 94 0D FA 91 E9 40 FC 25 9D C7 B7 68 19 56 8F
      :               11 70 6A D7 F1 C9 11 4F 3A 7E 3F 99 8D 6E 76 A5
      :               74 5F 5E A4 55 53 E5 C7 68 36 53 C7 1D 3B 12 A6
      :               85 FE BD 6E A1 CA DF 35 50 AC 08 D7 B9 B4 7E 5C
      :               FE E2 A3 2C D1 23 84 AA 98 C0 9B 66 18 9A 68 47
      :               E9
  358   3:             INTEGER 65537
      :             }
      :           }
      :         }
  363  66:       [3] {
  365  64:         SEQUENCE {                //扩展 subjectKeyIdentifier 的值，使用者密钥标识符
  367  29:           SEQUENCE {
  369   3:             OBJECT IDENTIFIER subjectKeyIdentifier (2 5 29 14)
  374  22:             OCTET STRING, encapsulates {//扩展 subjectKeyIdentifier 的值，使用者密钥标识
符
  376  20:               OCTET STRING
      :                 08 68 AF 85 33 C8 39 4A 7A F8 82 93 8E 70 6A 4A
```

```
          :                     20 84 2C 32
          :                   }
          :                 }
 398     14:               SEQUENCE {              // 扩展 keyUsage 的值，密钥使用，指明公钥可以完成的
用途和服务
 400      3:                 OBJECT IDENTIFIER keyUsage (2 5 29 15)
 405      1:                 BOOLEAN TRUE
 408      4:                 OCTET STRING, encapsulates {
 410      2:                   BIT STRING 1 unused bits
          :                     '0000011'B
          :                   }
          :                 }
 414     15:               SEQUENCE {            // 扩展 basicConstraints 的值，基本约束，确定该证书公
钥是否可以验证证书签名
 416      3:                 OBJECT IDENTIFIER basicConstraints (2 5 29 19)
 421      1:                 BOOLEAN TRUE
 424      5:                 OCTET STRING, encapsulates { // 扩展 basicConstraints 的值，基本约束，确定
该证书公钥是否可以验证证书签名
 426      3:                   SEQUENCE {
 428      1:                     BOOLEAN TRUE      // 该证书公钥可以验证证书签名
          :                     }
          :                   }
          :                 }
          :               }
          :             }
          :           }
 431     13:       SEQUENCE {            // 签名值
 433      9:         OBJECT IDENTIFIER
          :           sha1withRSAEncryption (1 2 840 113549 1 1 5)   // 使用 SHA1 和 RSA 加密算法的签名
 444      0:         NULL
          :         }
 446    129:       BIT STRING                  // 签名值,129 字节
          :         6C F8 02 74 A6 61 E2 64 04 A6 54 0C 6C 72 13 AD
          :         3C 47 FB F6 65 13 A9 85 90 33 EA 76 A3 26 D9 FC
          :         D1 0E 15 5F 28 B7 EF 93 BF 3C F3 E2 3E 7C B9 52
          :         FC 16 6E 29 AA E1 F4 7A 6F D5 7F EF B3 95 CA F3
          :         66 88 83 4E A1 35 45 84 CB BC 9B B8 C8 AD C5 5E
          :         46 D9 0B 0E 8D 80 E1 33 2B DC BE 2B 92 7E 4A 43
          :         A9 6A EF 8A 63 61 B3 6E 47 38 BE E8 0D A3 67 5D
          :         F3 FA 91 81 3C 92 BB C5 5F 25 25 EB 7C E7 D8 A1
          :       }
```