软件安全—恶意代码机理与防护

3.9 PE文件的数字签名与验证机制

杨秀璋

武汉大学国家网络安全学院

1455136241@qq.com

3.9 PE文件的数字签名与验证机制

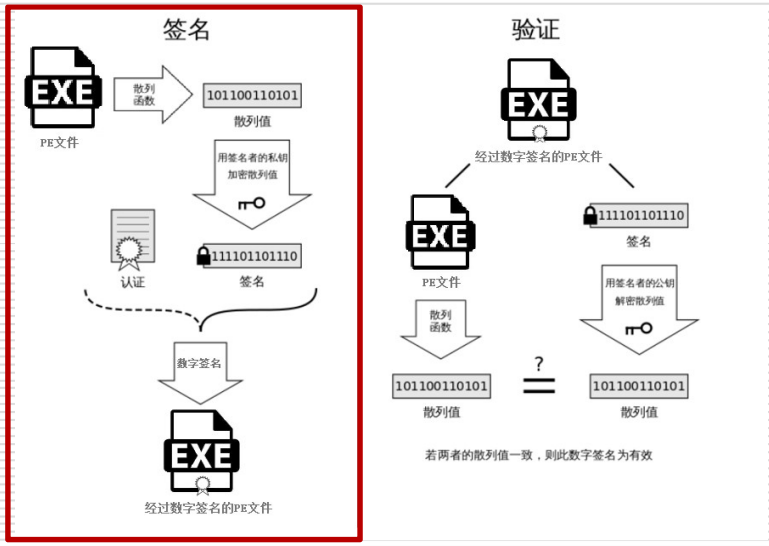
- PE文件的数字签名过程及验证机制
- 签名数据解析-以Zoomit为例

3.9.1 PE文件的数字签名过程及验证机制

- 对PE文件进行数字签名的作用
  - **防篡改：**通过对数字签名的验证，保证文件未被非法篡改。
  - **降低误报：**安全软件通过验证文件是否有正规厂商的数字签名来降低误报。

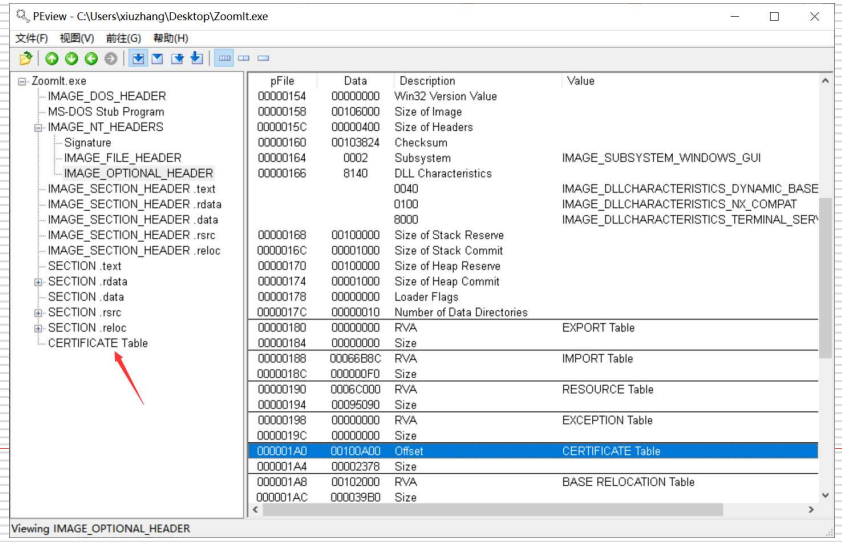
PE文件的数字签名过程

- **PE文件是怎样被签名的**
  - 软件发布者使用散列算法（如MD5或SHA）计算PE文件的散列值。
  - 软件发布者使用私钥对散列值进行签名得到签名数据。
  - 将签名私钥对应的公钥和签名数据等以证书的形式附加在PE文件之中，形成经过数字签名的PE文件。
  - 软件发布者将经过数字签名的PE文件进行发布。



PE文件签名位置

- 对哪些数据签名？
- 签名数据放在哪里？
  - Certificate Table：PE文件可选文件头DataDirecotry第5项（文件偏移及大小）



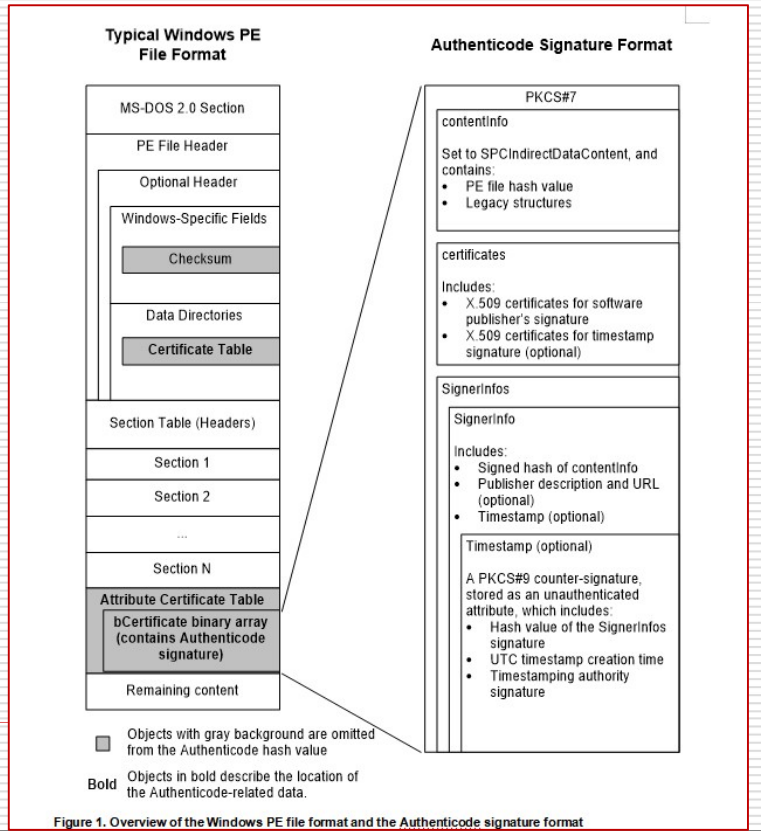
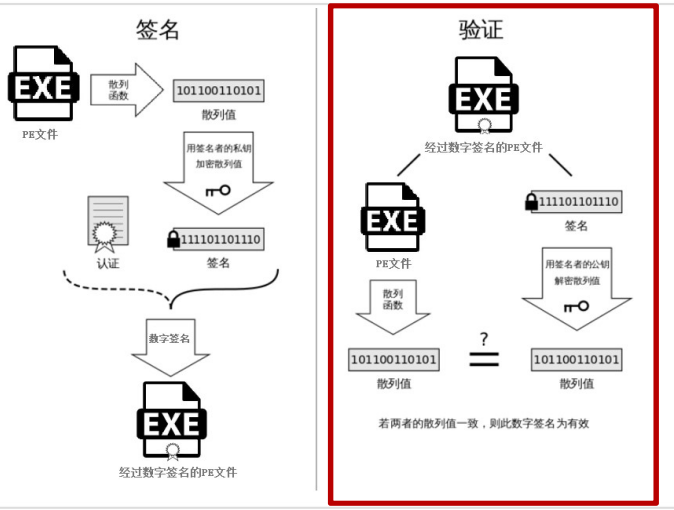


Figure 1. Overview of the Windows PE file format and the Authenticode signature format

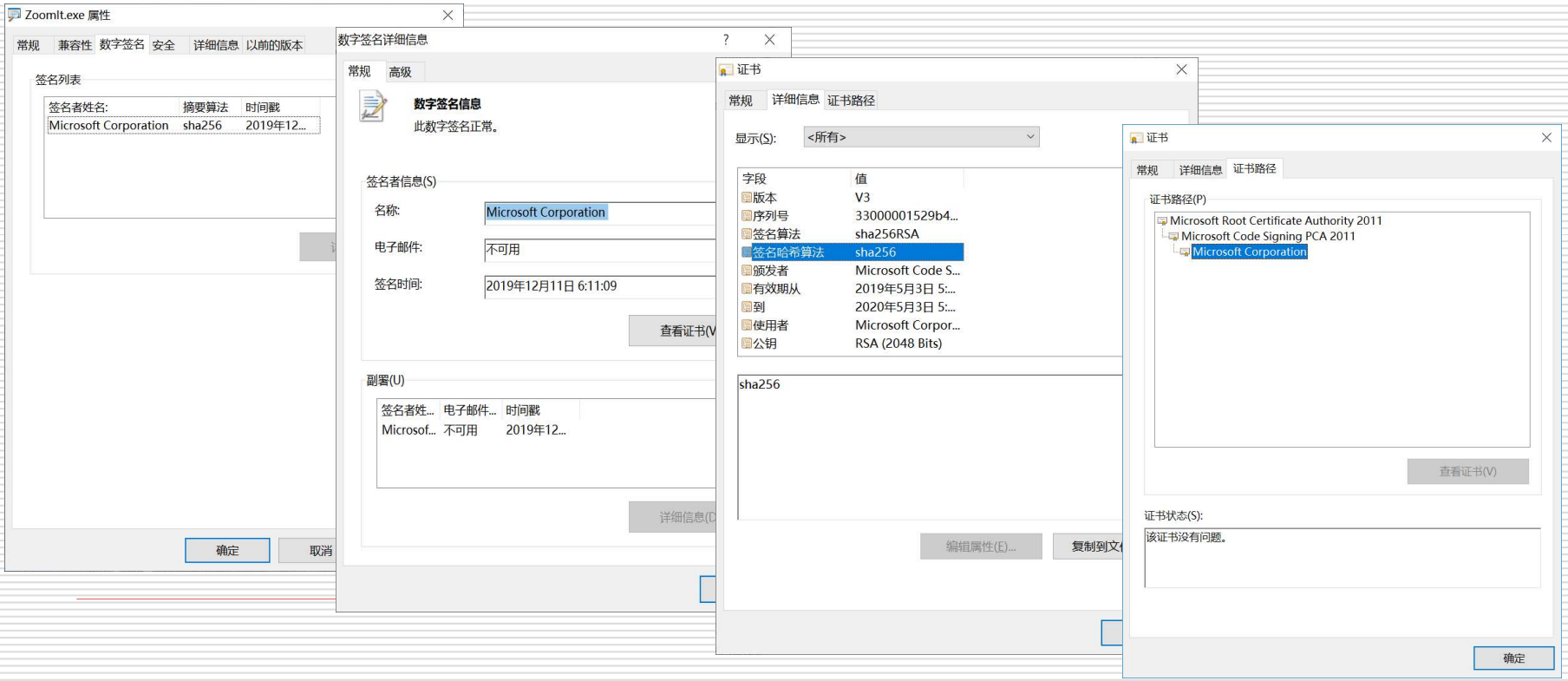
PE文件的签名验证机制

- **当软件使用者获得PE文件后，系统如何进行验证呢？**
  - 从PE文件证书中提取软件发布者的公钥、使用的散列算法、签名算法、原始散列值的签名数据。
  - 使用提取的公钥和对应签名验证算法将签名数据还原为原始PE文件的原始散列值。
  - 对现有PE文件使用同样的散列算法计算出对应的散列值。
  - 对比两个散列值是否一致，从而判断数据是否被破坏和篡改。



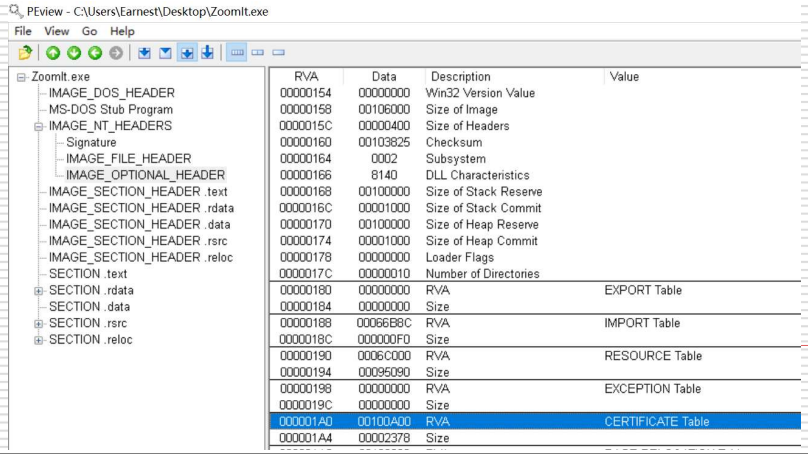
另外，系统还将验证证书路径和时间戳，以确保被验证PE文件的软件发布方身份和签名时间的可靠性。

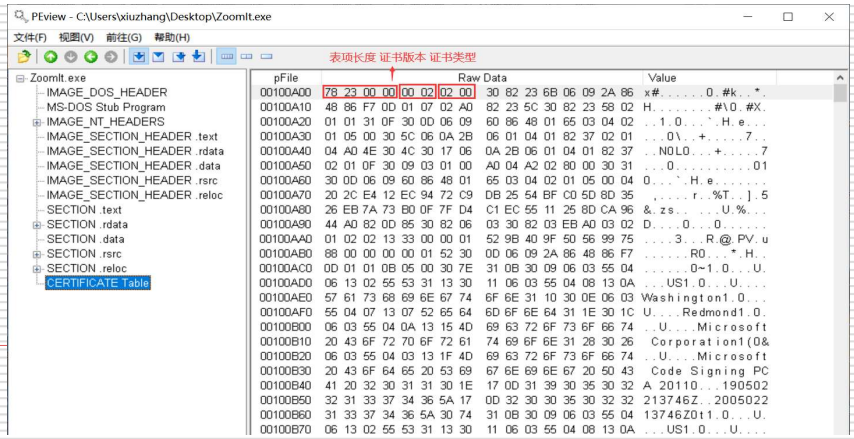
3.9.2 Zoomit签名数据解析



zoomit签名数据解析

- 文件开始位置：00100A00（长度：2378H）
  - 表项长度：4字节，头部和签名数据的总长度
  - 证书版本：2字节，常见0x0200表示WIN_CERT_REVISION_2
  - 证书类型：2字节，常见0x0002表示包含PKCS#7的SignData结构
  - **SignedData**：包含PE文件Hash值的签名数据、软件发布者公钥，选用的签名及散列算法等。（在文件中为ASN.1编码）





zoomit签名数据解析

- ①指定SignedData结构

  PKCS#7

  1.2.840.113549.1.7.2
- ②生成签名的哈希算法

  MD5 1.2.840.113549.2.5

  SHA1 1.3.14.3.2.26

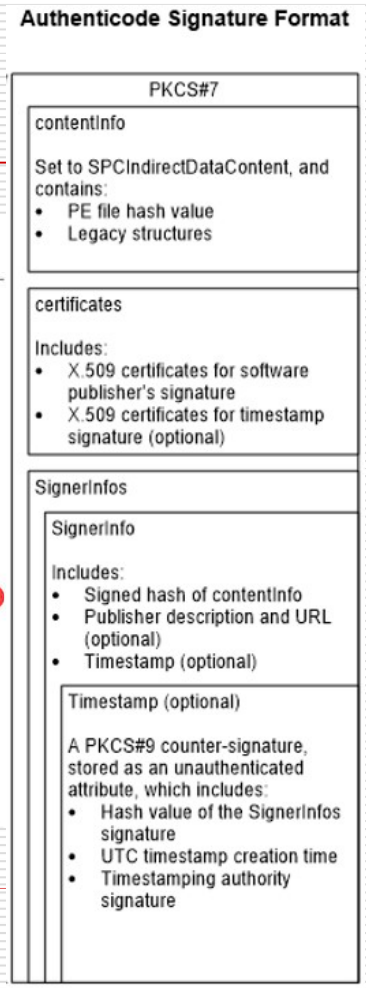
  SHA256 2.16.840.1.101.3.4.2.1
- ③签名属性

  SPC 1.3.6.1.4.1.311.2.1.4

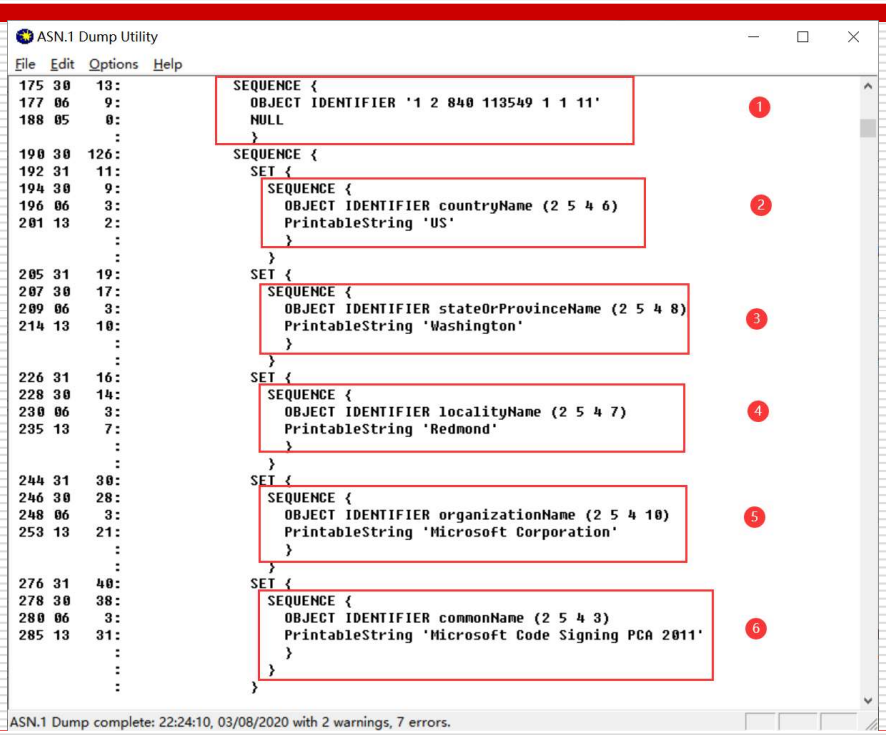
ASN.1 Dump Utility

File Edit Options Help

```
File: C:\Users\xiuzhang\Desktop\ZoomIt02.dat
Time: 22:24:10, 03/08/2020
-------------------------------------------------------------------------------
   0 30 9067: SEQUENCE {
   4 06    9:   OBJECT IDENTIFIER signedData (1 2 840 113549 1 7 2)
  15 A0 9052:   [0] {
  19 30 9048:     SEQUENCE {
  23 02    1:       INTEGER 1
  26 31   15:       SET {
  28 30   13:         SEQUENCE {
  30 06    9:           OBJECT IDENTIFIER '2 16 840 1 101 3 4 2 1'
  41 05    0:           NULL
                        }
                      }
  43 30   92:       SEQUENCE {
  45 06   10:         OBJECT IDENTIFIER
                        spcIndirectDataContext (1 3 6 1 4 1 311 2 1 4)
  57 A0   78:         [0] {
  59 30   76:           SEQUENCE {
  61 30   23:             SEQUENCE {
  63 06   10:               OBJECT IDENTIFIER
                              spcPeImageData (1 3 6 1 4 1 311 2 1 15)
  75 30    9:             SEQUENCE {
  77 03    1:               BIT STRING 0 unused bits
                              Error: Object has zero length.
  80 A0    4:               [0] {
  82 A2    2:                 [2] {
```



PE文件数字签名解析



证书颁发者信息

①OID值

②国家名称

③州或省名称

④城市名称

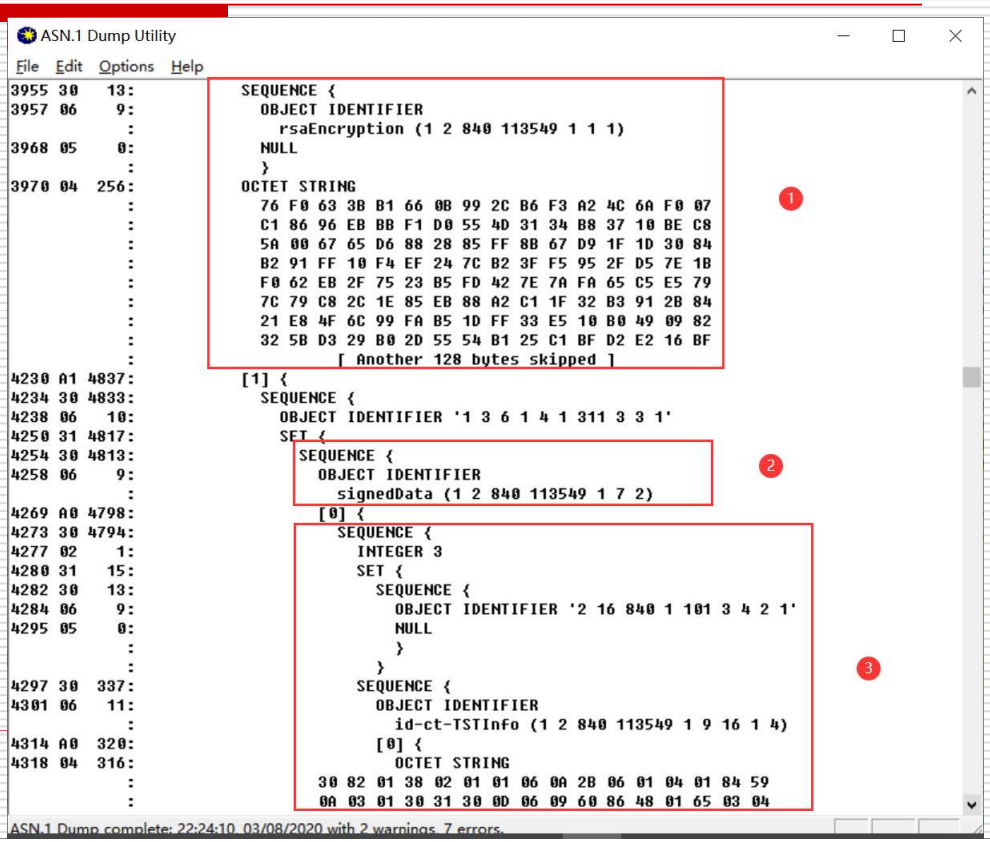
⑤组织名称

⑥通用名

PE文件数字签名解析

验证证书链相关字段：

- ①签名算法：RSA加密
- ②signedData采用PKCS#7格式
- ③SHA256散列算法



PE文件数字签名解析

- 散列算法：SHA256

```
SEQUENCE {
  OBJECT IDENTIFIER '2 16 840 1 101 3 4 2 1'
  NULL
  }
}
```

- 摘要数据

```
SEQUENCE {
  OBJECT IDENTIFIER
    messageDigest (1 2 840 113549 1 9 4)
  SET {
    OCTET STRING
      FF A8 29 9B 70 AD 5B 35 BD BE C9 5F 21 33 63 84
      95 80 DA 45 F6 22 F0 CD 8D 74 1A 7A 2B 44 08 A1
    }
  }
```

- 公钥数据

```
SEQUENCE {
  OBJECT IDENTIFIER
    rsaEncryption (1 2 840 113549 1 1 1)
  NULL
  }
BIT STRING 0 unused bits
  30 82 01 0A 02 82 01 01 00 B1 A7 89 D3 F6 A7 EE
  F4 ED 74 88 9C 9E C1 85 C7 94 DF 96 12 1E FD 36
  8A 2B F6 78 4C E9 E9 D4 35 B0 9B 85 7F 6F 4B EF
  1C FF 77 88 55 AC D2 62 75 41 4E A2 F3 5B BC 56
  37 56 C2 70 8B 43 6E C1 33 28 41 36 06 CC 12 BA
  7E C9 F7 10 9A 2F 07 06 1C A1 6B FA A5 94 97 3E
  E1 87 D4 5C 9A 36 12 90 E1 8D C9 B3 9F AE 08 F7
  B2 6D 03 4A DA 0A F2 58 F0 13 25 15 79 17 CC 1F
        [ Another 142 bytes skipped ]
}
```

- RSA签名后的数据

```
      SEQUENCE {
        OBJECT IDENTIFIER '1 2 840 113549 1 1 11'
        NULL
        }
      OCTET STRING
B0 DE FB 7B 7D 26 28 FE 5B A6 D9 A4 DB E7 F6 BE
17 C6 EC 7D E5 8B 1D 0D DD 6D BF 5E 2A 27 AA 64
64 28 CF B2 98 1F D8 21 24 3E 27 86 18 6A EC 3D
C0 09 23 4D 66 3B A4 57 03 9A 21 6D BD C6 A8 DB
8B 26 4D 2A 2A FD 4A 5C 0C 68 0A 9C 6F 9C 8B 53
78 AA 88 A2 65 6C 0A A2 F9 EA 50 EA 5C C4 4C 44
59 39 41 CC 71 94 29 FB 1E 5B 4F 65 6F B0 12 4C
D6 11 11 B2 EB E3 45 04 9B B5 29 08 44 6F 32 04
      [ Another 128 bytes skipped ]
```