PKI数据格式

目录

简介

本文描述最普通的公共密钥基础设施(PKI)数据格式和编码。

先决条件

要求

Cisco 建议您了解以下主题：

- 公钥加密(基本概念)。
- 公钥基础架构(基本概念)。

使用的组件

本文档不限于特定的软件和硬件版本。

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您使用的是真实网络，请确保您已经了解所有命令的潜在影响。

规则

有关文档规则的信息，请参阅 Cisco 技术提示规则。

ASN.1符号

抽象语法标记(ASN.1)是数据类型和值的定义的一种规范语言，并且那些数据类型和值如何使用并且被结合以多种数据结构。标准的目标是定义信息抽象语法没有限制条件信息如何为发射编码。

这是摘自*X.509 RFC*部分的示例：

```
   Version  ::=  INTEGER  {  v1(0), v2(1), v3(2)  }
CertificateSerialNumber ::= INTEGER
Validity ::= SEQUENCE {
notBefore Time,
notAfter Time }
Time ::= CHOICE {
utcTime UTCTime,
generalTime GeneralizedTime }
```

参考从国际电信联盟(ITU-T)标准站点的这些文档：

- X.680 ASN.1 ：基本符号的规格
- X.681 ASN.1 ：信息对象规格
- X.682 ASN.1 ：限制条件规格
- X.683 ASN.1 ：ASN.1规格的参数化

ITU-T建议搜索- **X.509的**搜索在**Rec。或者标准语言**设置为**ASN.1**。

BER/CER/DER编码

ITU-T定义编码数据结构一个标准的方式在ASN.1描述的到二进制数据。X.690定义了基本编码规则(BER)和其两子集、规范编码规则(CER)和著名的编码规则(DER)。全部三根据在一层次结构包装的**类型长度值**数据域，从**顺序**、**集**和**选择**被建立，与这些差异：

- BER提供编码同一个数据多种方式，没有适用与crypto操作。
- CER提供毫不含糊的编码并且以一数据结尾标记在特定情况下使用不确定长度数据。
- DER提供毫不含糊的编码并且在特定情况下使用明确长度标记。
- 在三中， DER是通常遇到，当交易与PKI和crypto有效载荷时的那个。

示例：在DER， 20位值`1010 1011 1100 1101 1110`编码如下：

- **标记**：`0x03` (bitstring)
- **长度**：`0x04` (字节)
- **值**：`0x04ABCDE 0`
- **完整DER编码**： `0x030404ABCDE0`

导致的`04`意味着必须丢弃最后4个位(等于落后的`0`位)编码的值，因为编码的值在字节边界不结束。

参考从TU-T标准站点的这些文档：

- X.690 ASN.1编码规则：基本编码规则(BER)，规范编码规则(CER)和著名的编码的规格规定(DER)

从维基百科站点，参考这些文档：

- 基本编码规则
- 规范编码规则
- 著名的编码规则

DER HEX转储

Cisco IOS、可适应安全工具(ASA)和其它设备显示DER内容作为**HEX转储**用**show running-config命令**。 这是输出：

```
crypto pki certificate chain root
certificate ca 01
30820213 3082017C A0030201 02020101 300D0609 2A864886 F70D0101 04050030
1D310C30 0A060355 040B1303 54414331 0D300B06 03550403 1304726F 6F74301E
170D3039 30373235 31313436 33325A17 0D313230 37323431 31343633 325A301D
...
```

这种HEX转储可以转换回到DER以多种方式。例如，您能删除空格符和管道传送它到**xxd程序**：

```
$ cat ca.hex | tr -d ' ' | xxd -r -p -c 32 | openssl x509 -inform der -text -noout
```

另一简单的方法是使用此Perl脚本：

```
#!/usr/bin/perl
foreach (<>) {
s/[^a-fA-F0-9]//g;
print join("", pack("H*", $_));
} $ perl hex2der.pl < hex-file.txt > der-file.der
```

另外，一个紧凑方式转换**cert转存**，每一个以前手工复制对有分机的**.hex**一个文件，从**bash** line命令如显示此处：

```
for hex in *.hex; do
b="${hex%.hex}"
hex2der.pl < "$hex" > "$b".der
openssl x509 -inform der -in "$b".der > "$b".pem
openssl x509 -in "$b".pem -text -noout > "$b".txt
done
```

每个文件导致：

- **file.hex -**原始文件(必须包含仅六角形的位)。
- **file.der -**在DER (二进制)格式的**证书**。
- **file.pem -**在PEM (Base64 +页眉/页脚)格式的**证书**。
- **file.txt -**证书的用户友好，可读的版本。

Base64编码

Base64编码类似只代表与64个可印字符(`A-Za-z0-9+/`)的二进制数据于**UUENCODE**。在转换从二进制到Base64，原始数据的每6位块编码到与转换表的一个8位可打印的ASCII字符。所以，数据的大小，在编码由33百分比后(数据增加计时6个位分开的8，等于1.333)。

24位缓冲区使用三(3)八组的转换(8)位到六(6)位的四(4)组里。所以一(1)或两(2)填充字节也许要求在输入数据流结束时。填充符表示在Base64-encoded数据结束时，由一个等于(=)每八组(8)填充位的符号被添加到输入在编码期间。

参考从维基百科的此示例。

参考在RFC 4648的多数最近信息：Base16、Base32和Base64数据编码。

PEM编码

增强加密邮件(PEM)是一个全双工互联网工程任务组(IETF) PKI标准为了交换安全消息。它同样地不再用途广泛，但是其封装语法广泛被借用为了格式化和交换Base64-encoded Pki相关数据。

PEM RFC 1421，第4.4部分：封装机制，定义了PEM消息如分隔由封装限定范围(EBs)，根据RFC 934，与此格式：

```
-----BEGIN PRIVACY-ENHANCED MESSAGE-----
Header: value
Header: value
...

Base64-encoded data
...
-----END PRIVACY-ENHANCED MESSAGE-----
```

实际上，当分配时PEM格式化的数据，此边界格式使用今天：

```
-----BEGIN type-----
...
-----END type-----
```

类型可以是其他密钥或证书例如：

- RSA
-
-
-
- X509 CRL

注意：虽然RFC不做此必须，编号导致和落后在EBs飞奔(-)是重大的，并且应该总是五(5)。否则，一些应用程序，例如Openssl，在输入堵塞。另一方面，其他应用程序，例如Cisco IOS，根本不要求EBs。

欲知详情参考这些最最近的RFC ：

- RFC 1421 ：PEM第I部分：消息加密和认证程序
- RFC 1422 ：PEM第II部分：基于认证的密钥管理
- RFC 1423 ：PEM Part III ：算法、模式和标识符
- RFC 1424 ：PEM Part IV ：关键证明和相关服务

X.509证书和Crl

X.509是X.500的一子集，是关于开放式系统互联的一个延长的ITU规格。它特别地处理证书和公共密钥和适应作为互联网标准由IETF。X.509提供一个结构和语法，表示用RFC ASN.1符号，为了存储证书信息和证书撤销列表。

例如在X.509 PKI， CA问题绑定公共密钥的证书， ：Rivest Shamir Adelman (RSA)或数字签名算法(DSA)密钥对一个特定的特有名(DN)，或者对一代替名称例如电子邮件地址或完全合格的域名(FQDN)。DN跟随在X.500标准的结构。示例如下：

```
CN=common-name OU=organizational-unit O=organization L=location
```

C=country

由于ASN.1定义， X.509数据可以编码到DER为了交换以二进制形式，和或者，转换对文本基于通信方式的Base64/PEM，例如在终端的复制-粘贴。

- 参考此ITU-T标准文档X.509开放式系统互联-目录：公共密钥和属性证书框架。
- 参考的RFC 5280 ：X.509证书和证书撤销列表(CRL)配置文件欲知更多信息。

PKCS标准

公钥加密标准(PKCS)是从部分转变了成业界标准的RSA实验室的规格。经常遇到的那些，与这些主题的交易;然而，不是所有与数据格式的交易。

PKCS-1 (RFC 3347) -报道基于RSA的加密算法的实施方面(crypto原始，加密/签名策划， ASN.1语法)。

PKCS#5 (RFC 2898) -报道基于密码的密钥派生。

PKCS-7 (RFC 2315)和S/MIME RFC 3852 -定义了消息语法传送签字和已加密数据和涉及的证书。常用完全作为X.509证书的一个容器。

PKCS#8-定义了消息语法传输明文或已加密RSA密钥对。

PKCS#9 (RFC 2985) -定义了另外的对象类和标识属性。

PKCS-10 (RFC 2986) -定义了证书签名请求的(CSR)消息语法。CSR由实体发送对CA并且包含CA将签字的信息，例如公共密钥关键信息、标识和另外的属性。

PKCS-12 -定义了包的相关PKI数据的一个容器(典型地，**实体密钥对+实体cert +根和半成品CA证书**)在单个文件内。它是Microsoft的个人信息信息交换(PFX)格式的演变。

参考这些资源：
- 在PKCS的维基百科条款
- RSA在PKCS的实验室页

相关信息

- **技术支持和文档 - Cisco Systems**