解析X.509证书

简介

X.509是密码学里公钥证书的格式标准。**X.509**证书已应用在包括TLS/SSL在内的众多Intenet协议里。同时它也用在很多非在线应用场景里，比如电子签名服务。**X.509**证书里含有公钥、身份信息和签名信息。对于一份经由可信的证书签发机构签名或者可以通过其它方式验证的证书，证书的拥有者就可以用证书及相应的私钥来创建安全的通信，对文档进行数字签名。

证书组成结构标准用**ASN.1语言**来进行描述。**X.509** 数字**证书结构**如下：

- 证书内容 `TBSCertificate`
  - 版本号 `EXPLICIT Version DEFAULT v1`
  - 序列号 `CertificateSerialNumber`

    - 签名 `AlgorithmIdentifier`
    - 颁发者 `Issuer Name`
    - 证书有效性 `Validity` （*有效日期*）
    - 主题 `Subject Name`
    - 主题公钥信息 `SubjectPublicKeyInfo`
    - 颁发者唯一身份信息 `IMPLICIT UniqueIdentifier OPTIONAL`
    - 主题唯一身份信息 `IMPLICIT UniqueIdentifier OPTIONAL`
    - 扩展信息 `EXPLICIT Extensions OPTIONAL`
- 签名算法 `AlgorithmIdentifier`
    - OID `OBJECT IDENTIFIER`
    - 参数 `ANY DEFINED BY algorithm OPTIONAL`
- 数字签名 `BIT STRING`

实现

数据结构

版本为整数格式，证书格式的版本只有v1、v2、v3，分别用整数0、1、2表示。

证书编码可以采用“TLV”方式，即依次对数据的类型（Type）、长度（Length）、值（Value）编码，一个基本的数据元就包括上面三个域，这样就可以完整地表示一个特定类型的数据。

```
struct TLV{
    TLV() {}
    char sig1[50],sig2[50];
};
```

签名算法给出了CA签发证书时所使用的数字签名算法，它的类型为AlgorithmIdentifier，签名算法中包含了签名算法和算法的参数。主题公钥信息给出了证书所绑定的加密算法和公钥：algorithm表示被绑定的、证书主体持有的公钥密码算法；subjectPublicKey是具体的公钥数据。证书的签发者和证书主体用**X.509** DN表示，DN是由RDN构成的序列，常用的属性类型名称以及简写如下：

属性类型名称	含义	简写
Common Name	通用名称	CN
Organizational Unit name	机构单元名称	OU
Organization name	机构名	O
Locality	地理位置	L
State or province name	州/省名	S
Country	国名	C

证书有效期给出证书的有效使用期，包含起、止两个时间值。签发者唯一标识符和主体唯一标识符给出了证书签发者和证书主体的唯一标识符。证书序列号为整数格式，证书序列号用来在某一个CA范围内唯一地标识一张证书。“签发者”和“证书序列号”配合起来就能唯一地标识一张数字证书。证书的签发者和证书主体分别标识了签发证书的CA实体和证书持有者实体，两者类型均为Name。

```
struct SignatureAlgorithm{
    TLV alg;
    TLV param;
};

struct SubjectPublicKey{
    TLV alg;
    TLV param;
    TLV2 pKey;
```

```
};

struct SignatureArray{
    char sig1[50],sig2[50];
}sA[7],is[6];

struct SignatureValue{
    TLV2 sigV;
};
```

证书的内容 `TbsCertificate` 为：

```
struct TbsCertificate{
    TLV version;
    TLV serialNumber;
    struct SignatureAlgorithm signature;
    struct SignatureArray issuer[6];
    TLV validity[2];
    struct SignatureArray subject[6];
    struct SubjectPublicKey SubjectPublicKeyInfo;
    TLV issuerUniqueID;
    TLV subjectUniqueID;
    TLV extensions;
};
```

证书最终的构成：

```
struct x509Cer{
    struct TbsCertificate cat;
    struct SignatureAlgorithm casa;
    struct SignatureValue casv;
}caCer;
```

函数

```
// bind OID
void sAfill(){
    strcpy(sA[0].sig1,"1.2.840.10040.4.1");
    strcpy(sA[0].sig2,"DSA");
    strcpy(sA[1].sig1,"1.2.840.10040.4.3");
    strcpy(sA[1].sig2,"sha1DSA");
    strcpy(sA[2].sig1,"1.2.840.113549.1.1.1");
    strcpy(sA[2].sig2,"RSA");
    strcpy(sA[3].sig1,"1.2.840.113549.1.1.2");
    strcpy(sA[3].sig2,"md2RSA");
    strcpy(sA[4].sig1,"1.2.840.113549.1.1.3");
    strcpy(sA[4].sig2,"md4RSA");
    strcpy(sA[5].sig1,"1.2.840.113549.1.1.4");
    strcpy(sA[5].sig2,"md5RSA");
    strcpy(sA[6].sig1,"1.2.840.113549.1.1.5");
    strcpy(sA[6].sig2,"sha1RSA");
}
```

```
// bind RDN
void isFill(){
    strcpy(is[0].sig1,"2.5.4.6");
    strcpy(is[0].sig2,"Country ");
    strcpy(is[1].sig1,"2.5.4.8");
    strcpy(is[1].sig2,"Sate or province name ");
    strcpy(is[2].sig1,"2.5.4.7");
    strcpy(is[2].sig2,"Locality ");
    strcpy(is[3].sig1,"2.5.4.10");
    strcpy(is[3].sig2,"Organization name ");
    strcpy(is[4].sig1,"2.5.4.11");
    strcpy(is[4].sig2,"Organizational Unit name
");
    strcpy(is[5].sig1,"2.5.4.3");
    strcpy(is[5].sig2,"Common Name ");
}
```

```
void fill(int){
    // The sequence number of each field of the certificate structure that invokes the TLV function is bound to the certificate structure content
    // Fill in the ca_cer structure
}
```

```
Len tlv(){
    // TLV matched recursion
}
```

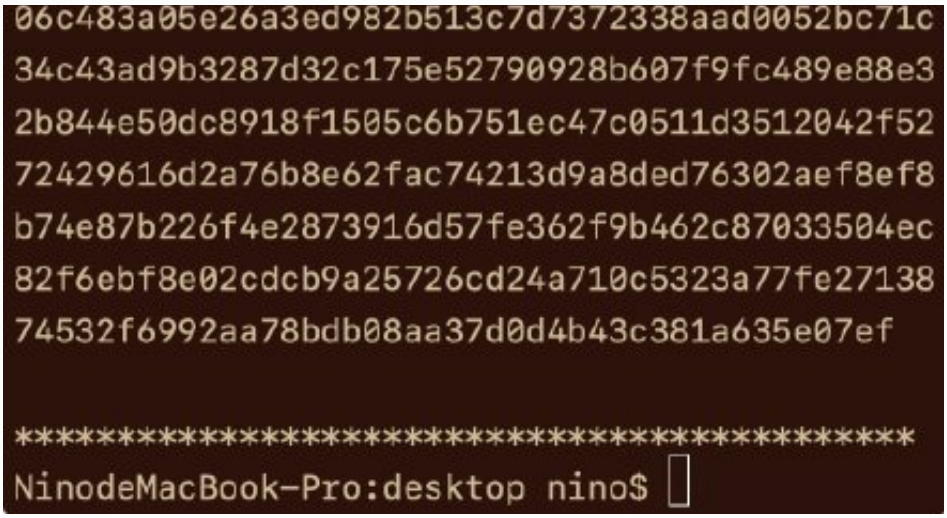
```
// Gets the contiguous bytecode (string) from the file and assigns it to the string s
void bitFill(int dd){
    strcpy(s,"");
    for(int i=0;i<dd;i++){
        unsigned char tl=fgetc(filePointer);
        int d=tl;
        char tsig2[10];
        sprintf(tsig2,"%02x",d);
        strcat(s,tsig2);
    }
}
```

结果

```
Last login: Sun Dec 16 17:41:04 on ttys000
[cd NinodeMacBook-Pro:~ nino$ cd desktop
[NinodeMacBook-Pro:desktop nino$ g++ X509.c -w
[NinodeMacBook-Pro:desktop nino$ ./a.out

******* Certificate Athority Parsing *******
```

```
version:        v3

-----------------------------------------

serialNumber:   2b85f2fe98d176994f38bfab9da62
d5f

-----------------------------------------

name of alg of signature:       sha1RSA

-----------------------------------------

param of signature:     NULL

-----------------------------------------

validity 150523034331Z-200523035214Z

--------------- ISSUER ---------------------
Country of issuer:      CN
Sate or province name of issuer:        SC
Locality of issuer:     CD
Organization name of issuer:    UESTC
Organizational Unit name of issuer:     CS
Common Name of issuer:  testCA

-----------------------------------------

--------------- SUBJECT --------------------
Country of subject:     CN
Sate or province name of subject:       SC
Locality of subject:    CD
Organization name of subject:   UESTC
Organizational Unit name of subject:    CS
Common Name of subject: testCA
```

```
-----------------------------------------

name of alg of SubjectPublicKey:        RSA

-----------------------------------------

param of alg of SubjectPublicKey:       NULL

-----------------------------------------

SubjectPublicKey:       003082010a0282010100d
49f7db04dd0136c7663ede566d0a6f7b14227326544bf
96cbbc5f7a0c5762fdcdae250aad8dc97cbba6a4cbc36
5c5c76fa856932dc92a24b63092f72db4215407a20532
cc9167a58259442253dcfd38e83438e524f24965cb1fa
c9621069baed0858b71b178d602db9e9ce4a800953c17
589bad04b0c8112084a5d1cb4b47d3900f7a1686e3182
89445408e0126c5fe973029eb35a74d8ceaaba57a1091
93ad6073f20078e4566796065e6ddf6157c5dd7d86693
c24fa983420616731e673f0ae0bd866fc148daceb3e03
b2aef1cc091f5b707e5d745a5f310e44c5531432ac4b9
9c47df54acec9d070203010001

-----------------------------------------

name of signatureAlgorithm:     sha1RSA

-----------------------------------------

param of signatureAlgorithm:    NULL

-----------------------------------------

signiture value:        00689be24fac4b849bbf3
763b7046561d97f79e5996f2836efd272364152489bf1
6131ae7f3031f751066ad84c61a8912ececb453465d27
9c12bcf97030e0205bfd78d059d6b864e6a888fc599e0
f5b7336d04a3b6127a714699fc022161f348b48e3137b
```



如上图，可以看到，**版本、序列号、算法、标识信息、有效期、签发者信息、主题信息、公钥加密算法、公钥数据、签名算法和签名结果**依次显示！

参考

喝水不忘挖井人，在此感谢为我提供思路的资料：

- x.509数字证书编码详解
- X.509证书的编码及解析
- X.509数字证书的编码
