**北京科技大学实验报告**

学院：计通学院 专业：信息安全 班级：信安182

姓名：王泽辰 学号：41824180 实验日期：2020年12月7日

**分析：**

1。首先生成selfcert

CertAndKeyGen cak = new CertAndKeyGen("RSA","MD5WithRSA",null);

//参数分别为 公钥算法 签名算法 providername（因为不知道确切的 只好使用null 既使用默认的provider）

cak.generate(1024);

//生成一对key 参数为key的长度 对于rsa不能小于512

X500Name subject = new X500Name("CN=simic,o=shanghai");

//subject name

X509Certificate certificate = cak.getSelfCertificate(subject,10);

// 后一个long型参数代表从现在开始的有效期 单位为秒（如果不想从现在开始算 可以在后面改这个域）

BASE64Encoder base64 = new BASE64Encoder();

FileOutputStream fos = new FileOutputStream(new File("d://test.crt"));

base64.encodeBuffer(certificate.getEncoded(), fos);

//生成cert文件 base64加密 当然也可以不加密

2。生成非自签的cert

首先按照1走一遍生成一个自签证书

byte certbytes[] = certificate.getEncoded();

X509CertImpl x509certimpl = new X509CertImpl(certbytes);

X509CertInfo x509certinfo = (X509CertInfo)x509certimpl.get("x509.info");

X500Name issuer = new X500Name("CN=fatal,o=shanghai");

x509certinfo.set("issuer.dname",issuer);

//设置issuer域

Date bdate = new Date();

Date edate = new Date();

edate.setTime(bdate.getTime() + validity \* 1000L \* 24L \* 60L \* 60L);

//validity为有效时间长度 单位为秒

CertificateValidity certificatevalidity = new CertificateValidity(bdate, edate);

x509certinfo.set("validity", certificatevalidity);

//设置有效期域（包含开始时间和到期时间）域名等同与x509certinfo.VALIDITY

x509certinfo.set("serialNumber", new CertificateSerialNumber((int)(date.getTime() / 1000L)));

//设置序列号域

CertificateVersion cv = new CertificateVersion(CertificateVersion.V3);

x509certinfo.set(X509CertInfo.VERSION,cv);

//设置版本号 只有v1 ,v2,v3这几个合法值

以上是证书的基本信息

ObjectIdentifier oid = new ObjectIdentifier(new int[]{1,22});

//生成扩展域的id 是个int数组 第1位最大2 第2位最大39 最多可以几位不明....

byte l = 0x11;//数据总长17位

byte f = 0x04;

String userData = "hohohohohahahahah";

byte[] bs = new byte[userData.length()+2];

bs[0] = f;

bs[1] = l;

for(int i=2;i<bs.length;i++)

{

  bs[i] = (byte)userData.charAt(i-2);

}

Extension ext = new Extension(oid,true,bs);

// 生成一个extension对象 参数分别为 oid，是否关键扩展，byte[]型的内容值

CertificateExtensions exts = new CertificateExtensions();

exts.set("aa",ext);

//如果有多个extension则都放入CertificateExtensions 类中，

x509certinfo.set(X509CertInfo.EXTENSIONS,exts);

//设置extensions域

X509CertImpl x509certimpl1 = new X509CertImpl(x509certinfo);

x509certimpl1.sign(cak1.getPrivateKey(), "MD5WithRSA");

//使用另一个证书的私钥来签名此证书 这里使用 md5散列 用rsa来加密

BASE64Encoder base64 = new BASE64Encoder();

FileOutputStream fos = new FileOutputStream(new File("d://test.crt"));

base64.encodeBuffer(x509certimpl1.getEncoded(), fos);

//生成文件

x509certimpl1.verify(cak.getPublicKey(),null);

//使用某个证书的公钥验证证书 如果验证不通过 则会抛错