**Android中签名流程机制**

**使用signapk工具签名**

Android签名apk之后，会有一个META-INF文件夹，这里有三个文件：

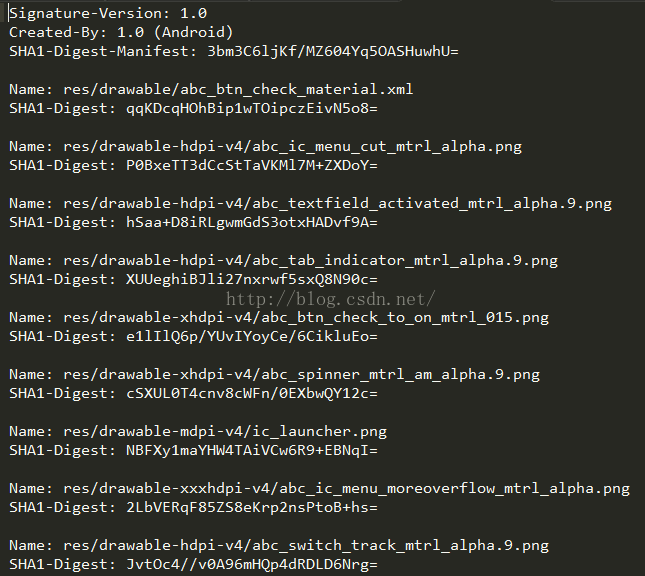
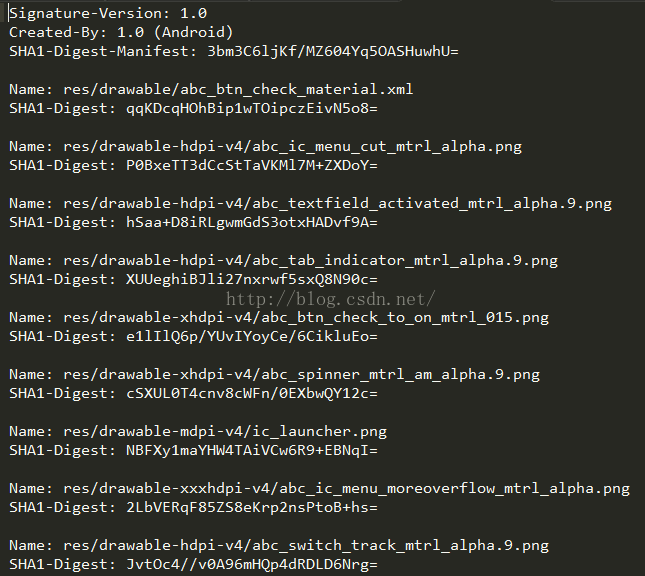
**MANIFEST.MF**

**CERT.RSA**

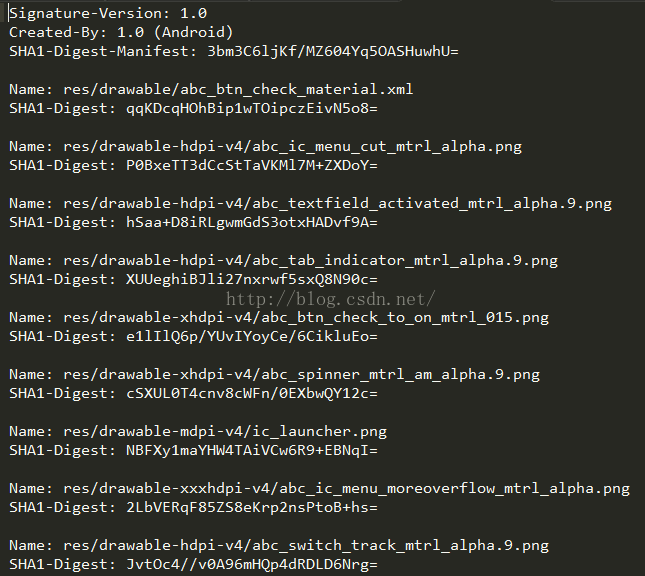
**CERT.SF**

## ****MANIFEST.MF****

对apk中的所有文件(MANIFEST.MF,CERT.RSA,CERT.SF除外) 的内容做一次SHA1算法就是计算出文件的摘要信息,然后用Base64进行编码



## 2. ****CERT.SF****

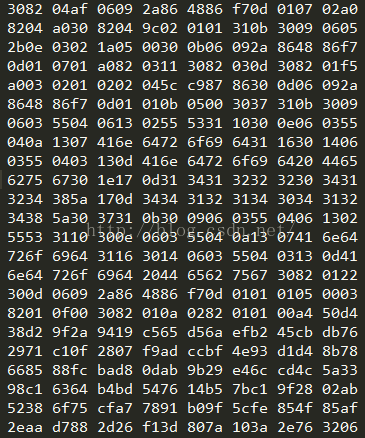


内容感觉和MANIFEST.MF的内容差不多，CERT.SF文件做了什么？

1》计算这个MANIFEST.MF文件的整体SHA1值，再经过BASE64编码后，记录在CERT.SF主属性块（在文件头上）的“SHA1-Digest-Manifest”属性值值下

2》逐条计算MANIFEST.MF文件中每一个块的SHA1，并经过BASE64编码后，记录在CERT.SF中的同名块中，属性的名字是“SHA1-Digest

## 3. CERT.RSA

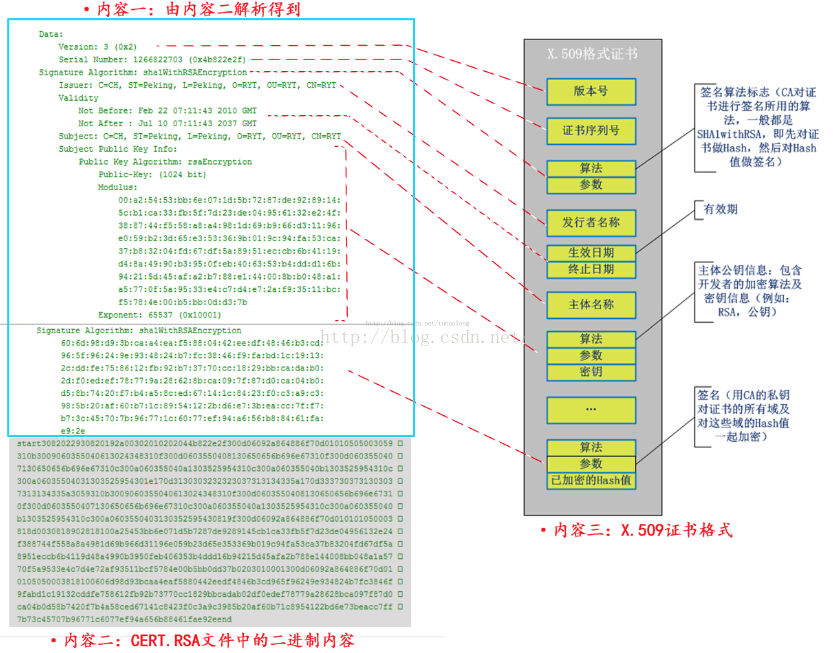


二进制文件，因为RSA文件加密，需要用openssl命令才能查看其内容

**openssl pkcs7 -inform DER -in CERT.RSA -noout -print\_certs –text**



信息解读如下：



CERT.RSA文件做了什么？

将之前生成的 CERT.SF文件， 用**私钥**计算出签名, 然后将签名以及包含公钥信息的数字证书一同写入  CERT.RSA  中保存。

# **签名的安全性分析**

首先，如果你改变了apk包中的任何文件，那么在apk安装校验时，改变后的文件摘要信息与MANIFEST.MF的检验信息不同，于是验证失败，程序就不能成功安装。  
其次，如果你对更改的过的文件相应的算出新的摘要值，然后更改MANIFEST.MF文件里面对应的属性值，那么必定与CERT.SF文件中算出的摘要值不一样，照样验证失败。  
最后，如果你还不死心，继续计算MANIFEST.MF的摘要值，相应的更改CERT.SF里面的值，那么数字签名值必定与CERT.RSA文件中记录的不一样，还是失败。  
那么能不能继续伪造数字签名呢？不可能，因为没有数字证书对应的私钥。

# ****知识点梳理****

1、数据指纹，签名文件，证书文件的含义

1》数据指纹就是对一个数据源做SHA/MD5算法，这个值是唯一的

2》签名文件技术就是：数据指纹+RSA算法

3》证书文件中包含了公钥信息和其他信息

4》在Android签名之后，其中SF就是签名文件，RSA就是证书文件我们可以使用openssl来查看RSA文件中的证书信息和公钥信息

2、我们了解了Android中的签名有两种方式：jarsigner和signapk 这两种方式的区别是：

1》jarsigner签名时，需要的是keystore文件，而signapk签名的时候是pk8,x509.pem文件

2》jarsigner签名之后的SF和RSA文件名默认是keystore的别名，而signapk签名之后文件名是固定的:CERT

3》Eclipse中我们在跑Debug程序的时候，默认用的是jarsigner方式签名的，用的也是系统默认的debug.keystore签名文件

4》keystore文件和pk8,x509.pem文件之间可以互相转化