# 签名工具

Android中签名的两个工具：jarsign和signapk

**区别：**

（1）jarsign是[Java](http://lib.csdn.net/base/17" \o "Java EE知识库" \t "_blank)本生自带的一个工具，他可以对jar进行签名的。

（2）signapk是后面专门为了Android应用程序apk进行签名的工具。

他们两的签名算法没什么区别，主要是签名时**使用的文件不一样**。

**（1）jarsign工具签名时使用的是keystore文件**

**（2）signapk工具签名时使用的是pk8,x509.pem文件**

Eclipse工具写程序的时候，出Debug包的时候，默认用的是jarsign工具进行签名的，Eclipse中有一个默认签名文件

# ****手动签名Apk包****

1. 使用keytool和jarsigne来进行签名
2. keytool工具生成一个keystore文件

**keytool -genkeypair -v -keyalg DSA -keysize 1024 -sigalg SHA1withDSA -validity 20000 -keystore D:\jiangwei.keystore -alias jiangwei -keypass jiangwei -storepass jiangwei**

**-alias是定义别名，这里为debug**

**-keyalg是规定签名算法，这里是DSA，这里的算法直接关系到后面apk中签名文件的后缀名，到后面会详细说明**

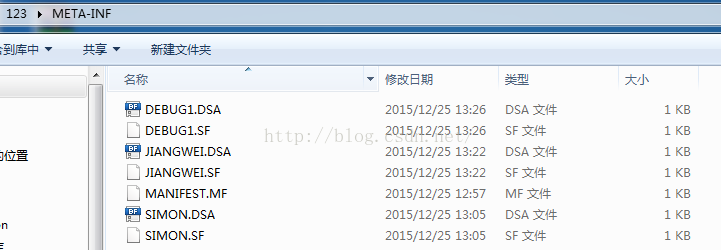
1. jarsign来对apk包进行签名

**jarsigner -verbose -sigalg SHA1withDSA -digestalg SHA1  -keystore D:\jiangwei.keystore -storepass jiangwei D:\123.apk jiangwei**

最后一个参数alias

**注意：Android中是允许使用多个keystore对apk进行签名的**

**这里我把签名之后的apk进行解压之后，发现有三个签名文件和证书(.SF/.DSA)**

****

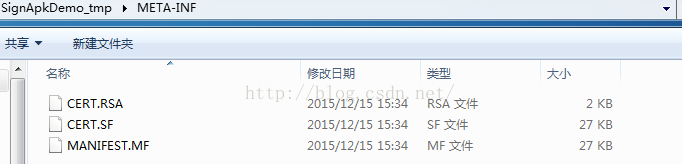
1. 使用signapk来进行签名

**java -jar signapk.jar .testkey.x509.pem testkey.pk8 debug.apk debug.sig.apk**

这里需要两个文件：.pk8和.x509.pem这两个文件

**pk8是私钥文件**

**x509.pem是含有公钥的文件**

**signapk签名之后的apk中的META-INF文件夹中的三个文件的名字是这样的，因为signapk在前面的时候不像jarsigner会自动使用别名来命名文件，这里就是写死了是CERT的名字**