APK 签名原理

1. 非常重要重要的东西（基础知识）

1.1、散列算法

散列函数（通常叫散列算法）可以把任意长度的数据映射成固定长度的数据，映射出来的数据称为散列值、哈希值、摘要。不同的散列值可以作为数据的唯一标识（类似指纹）。常见的散列算法有MD5 SHA-1 SHA-256，APK签名会使用SHA-256算法。

1.2、加密

加密=可读的明文数据->不可读的密文数据。只有通过对应的加密算法才可以解密。

常见的有：对称加密+不对称加密

对称加密：

简单的说，你有把锁A，只要用a钥匙就可以打开，那别人想开锁只要拿你的a钥匙就可用了。

非对称加密：

简单的说，你有把锁AB，有两把不同的钥匙a，b，用a锁上了 a就打不开了，你只能用b才可以打开；反之亦然。

1.3、数字签名

数字签名就是证明数据真实性的一种方式。

Apk中SHA-256加密就是数字签名，而且属于非对称加密

有了一把（钥匙）数字签名了，还需要一把钥匙。数字证书出来了。

1.4、数字证书

包括公钥信息公钥所有者信息、证书签发者信息。公钥证书由CA(证书颁发机构)保证。

总结：

APK签名和上述几个概念很有关系

数据 -------经过散列算法 ----散列值-----结果签名者私钥加密-----签名+ca认证----付给数据---有数字签名的数据。

签名数据-----公钥解密-------与数据直接散列对比-散列值-- 相同就允许

二、apk内部签名文件

Android studio 使用signapk.jar签名。

但是我们使用的时候需要keystore文件，这个是jarsigner签名方式，这是AS再内部做了转化。

主要会生成三个主要文件

1. MANIFEST.MF.MF清单文件

记录所有非目录文件的数值指纹（hash值）

在.MD中条目名称和条目属性组成

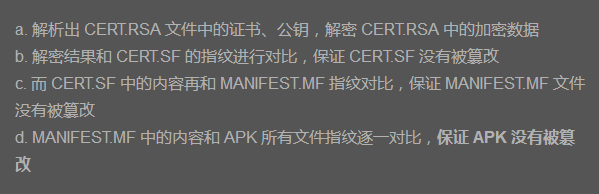
2、.SF待签名文件

针对.MF文件的数据指纹得到新的条目，简单的说就是.MF以及各个条目的hash记录。

3、.RSA签名结果文件

私钥算法得到签名算法，再使用签名算法对待待签名数据（.SF），最后签名数据和证书放在一起生成字节数组。

解析过程：



APK反编译：

1\ClassyShark.jar

直接 使用 java –jar ClassyShark.jar

编译jar包 出现图形界面

2\Apktool+dex2jar+jd-gui

3\Apktool+Smail2JavaUI