ANVA 移动应用开发者签名验签工具

使用说明

二〇一八年一月三日

1、引言

1.1、编写目的

本文档的编写目的是指导移动应用开发者使用安卓代码签名证书对公钥证书进行签名的过程。

1.2、参考资料

《CNCERT 移动开发者第三方认证方案》

《安卓代码签名证书使用简介》

1.3、术语与缩写词

CA：电子商务认证授权机构（CA, Certificate Authority），也称为电子商务认证中心，是负责发放和管理数字证书的权威机构，并作为电子商务交易中受信任的第三方，承担公钥体系中公钥的合法性检验的责任。

X.509：X.509 是被广泛使用的数字证书标准，是由国际电联电信委员会（ITU-T）为单点登录和授权管理基础设施制定的 PKI 标准。

PKCS#7：是描述数字证书的语法和其他加密消息，是数据加密和数字签名的方法，也包含了算法。当使用 PKCS#7 进行数字签名时，结果包含签名证书和已证明路径上任何其他证书。如果使用 PKCS#7 加密数据，通常包含发行者的参考消息和证书的序列号，它与用于解密已加密数据的公共密钥相关。

2、软件概述

2.1、软件用途

【签名软件】实现了《CNCERT 移动开发者第三方认证方案》所设计的 Android 开发者

第三方签名。使用从 CA 机构获得的合法数字证书对 Android 应用程序进行版权保护。

本软件使用场景适合所有的 Android 应用程序开发过程，但是如果您的 Android 应用程序尚未大范围的推广发行过，建议使用 CA 机构签发的安卓代码签名证书直接对 Android 应用程序进行签名打包（具体可参考《安卓代码签名证书使用简介》）。

2.2、环境配置

软件需要在安装有 jdk1.6 以上版本的计算机上运行。

2.3、软件使用

本软件的运行分为签名和验签两个过程，主要是对公钥证书文件进行签名，以及对签名文件进行签名验签。

2.3.1、签名

2.3.1.1、说明

签名过程是对 Android 应用程序原始数字证书的指纹特征值，使用 CA 机构颁发的 Android 代码签名证书进行签名。签名结果存储在指定位置。

2.3.1.2、准备、输入、处理、输出

（1） 准备

- 原始数字证书

由于是对原始数字证书的指纹特征值进行提取，所以需要预先从原始的 keystore 证书中导出目标证书的公钥数字证书。

命令格式：keytool -export -alias 【证书别名】 -keystore 【keystore 文件】 -file 【导出后的证书文件名（扩展名为 crt）】

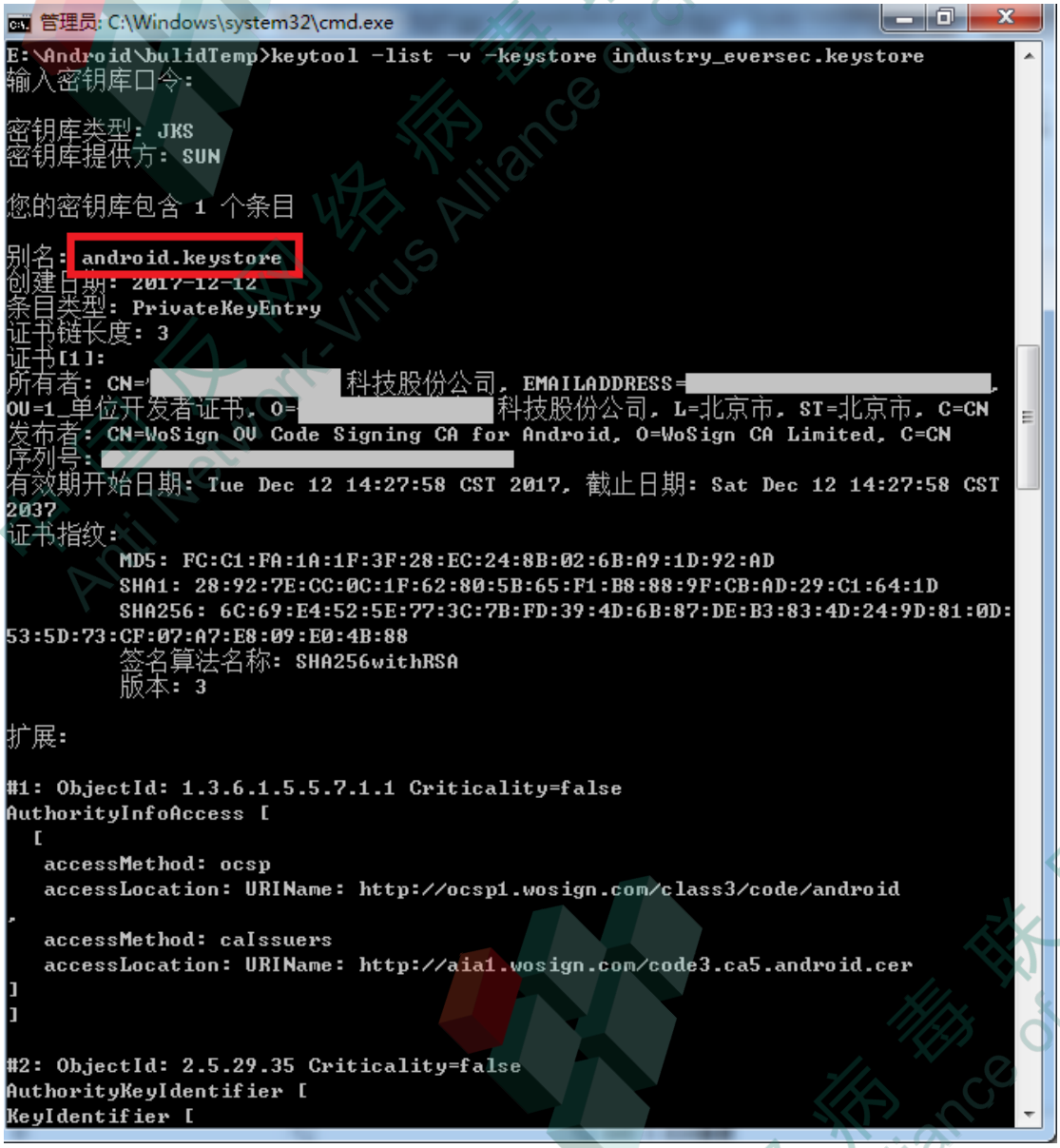
示例：keytool -export -alias android.keystore -keystore industry_eversec.keystore -file industry_eversec.crt

- CA 证书信息

为了使用 CA 颁发证书进行签名，需要获得 CA 颁发的数字证书的基本信息：证书保护密码、证书别名、证书别名密码。

命令格式：keytool -list -v -keystore 【安卓代码签名证书文件名称】

运行示例：



注：证书保护密码是用户在 ca.anva.org.cn 申请数字证书时所填写的口令，从这里获取的 CA 证书的证书别名密码与证书保护密码相同。

（2） 输入

用户可以运行签名程序，根据签名程序的逐步提示，输入签名所需的参数，完成签名过程。

参数说明：

- 证书绝对路径：CA 证书在磁盘中的位置
- 证书保护密码：CA 证书的证书保护密码
- 证书别名：keystore 文件中标识不同证书的名称，默认情况下 CA 颁发的 keystore 中只有一个证书
- 证书别名密码：同证书保护密码
- 导出后的证书公钥文件的绝对路径：原始证书中导出的公钥证书在磁盘中的位置
- 生成 PKCS7 签名文件的绝对路径：生成签名文件在磁盘中的位置

（3） 处理

软件的处理过程如下：

- 提取指纹特征值

  对原始证书的公钥证书提取 sha1 指纹值。
- 从 keystore 中提取证书

  利用证书别名从 keystore 文件中取出数字证书。
- 从证书中提取签名算法

  获取数字证书中约定的签名算法。
- 利用签名算法对指纹特征值签名

  用签名算法进行签名。
- 输出签名文件

  将 CA 证书的公钥证书和签名结果组合成 PKCS#7 格式的文件。

（4） 输出

输出的 PKCS#7 文件以 ANVA.RSA 命名，该文件是用 CA 证书对原始证书的拥有者的证明。需要用户在 APK 源码的根目录中建立 ANVA-INF 文件夹，并将 ANVA.RSA 文件放置在此文件夹中。

注：如果有加固需求，在 APK 的加固处理的时候，不要对 ANVA-INF 文件夹下的文件进行加固。

2.3.2、验签

2.3.2.1、说明

所有获得 APK 的用户都可以对 APK 进行验签，验证 APK 开发者的信息和原始签名证书的信息是完整一致的。这有利于 APK 的版权识别、APK 问题追溯、APK 的实名推广等。

2.3.2.2、准备、输入、处理、输出

（1） 准备

- 依照《CNCERT 移动开发者第三方认证方案》进行签名的 APK
- 从 APK 中提取 ANVA.RSA 和 APK 根目录下的 META-INF 文件夹下的 RSA 文件

（2） 输入

用户运行签名程序，根据签名程序的提示，逐步输入签名所需的参数，完成验签过程。

参数说明：

- 安卓签名文件：用原始证书对 APK 进行签名打包后生成的文件，位置在 APK 根目录 META-INF 文件夹下，以 RSA 为扩展名。
- ANVA.RSA 的绝对路径：ANVA.RSA 在磁盘中的位置，该文件可以从 APK 根目录下的 ANVA-INF 文件夹中提取。

（3） 处理

软件的处理过程如下：

- 提取安卓签名文件的证书

从安卓签名文件中提取原始签名证书的公钥证书。

- 提取安卓签名文件证书的指纹特征值

从原始证书的公钥证书中计算原始证书的 sha1 指纹特征。

- 验证签名结果

验证 ANVA.RSA 文件中的签名是对原始证书的 sha1 指纹特征的签名。

（4） 输出

当签名验证成功之后，软件会输出“证书验证成功！”，否则会提示“证书验证失败！”。