**1. android aosp 签名文件地址**

android AOSP 下保存的签名文件保存在 build/target/product/security/下面，如

~/workspace/aosp/build/target/product/security$ ls

Android.mk  media.x509.pem  platform.x509.pem  shared.pk8       testkey.pk8       verity\_key  verity.x509.pem

media.pk8   platform.pk8    README             shared.x509.pem  testkey.x509.pem  verity.pk8

可参见 <https://source.android.com/devices/tech/ota/sign_builds.html>

分别为

testkey, 如果没有特殊指明，这个是默认的key

platform, Test key for packages that are part of the core platform.

shared, Test key for things that are shared in the home/contacts process

media, Test key for packages that are part of the media/download system.

单独的package通过在Android.mk里指定LOCAL\_CERTIFICATE来指明一个特殊的key, 如果不指明的话，默认的是testkey, 你也可以指定一个你自己定义的key

device/yoyodyne/apps/SpecialApp/Android.mk

[...]

LOCAL\_CERTIFICATE := device/yoyodyne/security/special

Now the build uses the device/yoyodyne/security/**special.{x509.pem,pk8}** key to sign SpecialApp.apk. The build can use only private keys that are not password protected.

**Certificate 证书:** 是以  .x509.pem 结尾, 证书是用来验证一个package是否是用private key来签名的。

**Private key:** 私有的key是以 .pk8结尾    .pk8应该谨慎保管

Android 包含在 build/target/product/security的test key, 用make命令编译出来的OS会将所有的apk文件用test-kyes来签名，因为test-keys是公开的，因此任何人都可以用test-keys来签他们的apk, 这样在你build的系统里就可以用他们的apk来替换或hack你的系统，

**2. 在image里的每个apk必须签名, 签名的用处:**

    1). 当一个应用程序被replace的时候，新的应用程序必须和老的应用程序签名一致，这样才能访问老的应用程序的data.

    2). 如果两个或以上的应用程序想要分享一个user ID(这样他们就能共享数据)，那么他们必须用相同的签名。

**3. AOSP 签名与 reference code 签名**

   Reference产商和OEM产商会从AOSP里branch出属于自己的branch,然后在上面加feature, 这样也会拿到相同的testkey/platform/media/shared key, 因此在内部编译的时候都是默认的testkey, 而如果release出去的话，会进行产商自己的签名，这个签名只会让某个人或少数人知道，不会public出去，否则系统会被hack.

**但是 如果**

**LOCAL\_CERTIFICATE := PRESIGNED**

则表示 这个apk已经签过名了，系统不需要再次 签名；