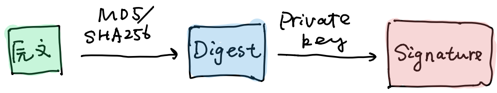
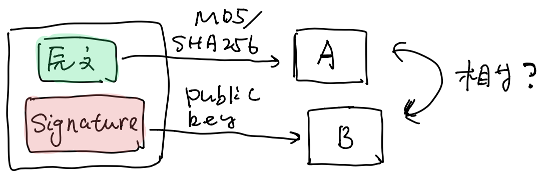
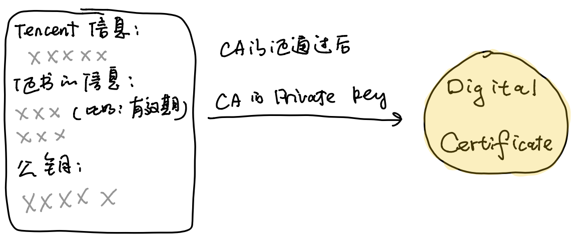
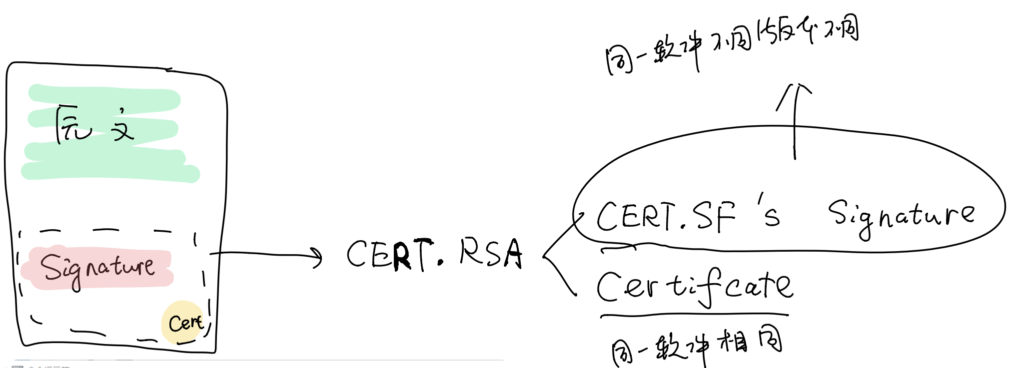
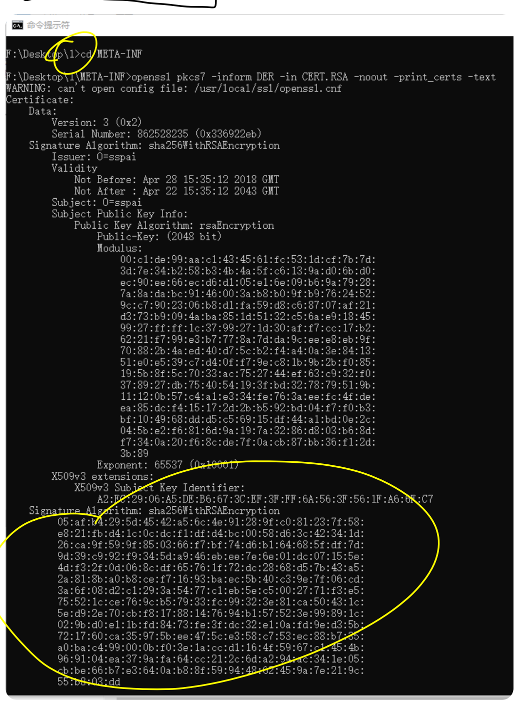




但，如果你拿到的 Public key 不是你想要的 PK，而是 Hacker 给你的 Public key
那么 Hacker 则可以伪装起来，向你发信息。
你如何知道你拿到的 Public key 是正确的呢？ CA








```
F:\Desktop\1>cd META-INF

F:\Desktop\1\META-INF>openssl pkcs7 -inform DER -in CERT.RSA -noout -print_certs -text
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 862528235 (0x336922eb)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=sspai
        Validity
            Not Before: Apr 28 15:35:12 2018 GMT
            Not After : Apr 22 15:35:12 2043 GMT
        Subject: O=sspai
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c1:de:99:aa:c1:43:45:61:fc:53:1d:cf:7b:7d:
                    3d:7e:34:b2:58:b3:4b:4a:5f:c6:13:9a:d0:6b:d0:
                    ec:90:ee:66:ec:d6:d1:05:e1:6e:09:b6:9a:79:28:
                    7a:8a:da:bc:91:46:00:3a:b8:b0:9f:b9:76:24:52:
                    9c:c7:90:23:06:b8:d1:fa:59:d8:c6:87:07:af:21:
                    d3:73:b9:09:4a:ba:85:1d:51:32:c5:6a:e9:18:45:
                    99:27:ff:ff:1c:37:99:27:1d:30:af:f7:cc:17:b2:
                    62:21:f7:99:e3:b7:77:8a:7d:da:9c:ee:e8:eb:9f:
                    70:88:2b:4a:ed:40:d7:5c:b2:f4:a4:0a:3e:84:13:
                    51:e0:e5:39:c7:d4:0f:f7:9e:c8:1b:9b:2b:f0:85:
                    19:5b:8f:5c:70:33:ac:75:27:44:ef:63:c9:32:f0:
                    37:89:27:db:75:40:54:19:3f:bd:32:78:79:51:9b:
                    11:12:0b:57:c4:a1:e3:34:fe:76:3a:ee:fc:4f:de:
                    ea:85:dc:f4:15:17:2d:2b:b5:92:bd:04:f7:f0:b3:
                    bf:10:49:68:dd:d5:c5:69:15:df:44:a1:bd:0e:2c:
                    04:5b:e2:f6:81:6d:9a:19:7a:32:86:d8:03:b6:8d:
                    f7:34:0a:20:f6:8c:de:7f:0a:cb:87:bb:36:f1:2d:
                    3b:89
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                A2:FC:29:06:A5:DE:B6:67:3C:EF:3F:FF:6A:56:3F:56:1F:A6:6F:C7
    Signature Algorithm: sha256WithRSAEncryption
         05:af:b4:29:5d:45:42:a5:6c:4e:91:28:9f:c0:81:23:7f:58:
         e8:21:fb:d4:1c:0c:dc:f1:df:d4:bc:00:58:d6:3c:42:34:1d:
         26:ca:9f:59:9f:85:03:66:f7:bf:74:d6:b1:64:68:5f:df:7d:
         9d:39:c9:92:f9:34:5d:a9:46:eb:ee:7e:6e:01:dc:07:15:5e:
         4d:f3:2f:0d:06:8c:df:65:76:1f:72:dc:28:68:d5:7b:43:a5:
         2a:81:8b:a0:b8:ce:f7:16:93:ba:ec:5b:40:c3:9e:7f:06:cd:
         3a:6f:08:d2:c1:29:3a:54:77:c1:eb:5e:c5:00:27:71:f3:e5:
         75:52:1c:ce:76:9c:b5:79:33:fc:99:32:3e:81:ca:50:43:1c:
         5e:d9:2e:70:cb:f8:17:88:14:76:94:b1:57:52:3e:99:89:1c:
         02:9b:d0:e1:1b:fd:84:73:fe:3f:dc:32:e1:0a:fd:9e:d3:5b:
         72:17:60:ca:35:97:5b:ee:47:5c:e3:58:c7:53:ec:88:b7:35:
         a0:ba:c4:99:00:0b:f0:3e:1a:cc:d1:16:4f:59:67:c1:45:4b:
         96:91:04:ea:37:9a:fa:64:cc:21:2c:6d:a2:94:ac:34:1e:05:
         cb:be:66:b7:e3:64:0a:b8:8f:59:94:48:62:45:9a:7e:21:9c:
         55:b8:03:dd
```


```
F:\Desktop\1_1\META-INF>openssl pkcs7 -inform DER -in CERT.RSA -noout -print_certs -text
WARNING: can't open config file: /usr/local/ssl/openssl.cnf
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 862528235 (0x336922eb)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: O=sspai
        Validity
            Not Before: Apr 28 15:35:12 2018 GMT
            Not After : Apr 22 15:35:12 2043 GMT
        Subject: O=sspai
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:c1:de:99:aa:c1:43:45:61:fc:53:1d:cf:7b:7d:
                    3d:7e:34:b2:58:b3:4b:4a:5f:c6:13:9a:d0:6b:d0:
                    ec:90:ee:66:ec:d6:d1:05:e1:6e:09:b6:9a:79:28:
                    7a:8a:da:bc:91:46:00:3a:b8:b0:9f:b9:76:24:52:
                    9c:c7:90:23:06:b8:d1:fa:59:d8:c6:87:07:af:21:
                    d3:73:b9:09:4a:ba:85:1d:51:32:c5:6a:e9:18:45:
                    99:27:ff:ff:1c:37:99:27:1d:30:af:f7:cc:17:b2:
                    62:21:f7:99:e3:b7:77:8a:7d:da:9c:ee:e8:eb:9f:
                    70:88:2b:4a:ed:40:d7:5c:b2:f4:a4:0a:3e:84:13:
                    51:e0:e5:39:c7:d4:0f:f7:9e:c8:1b:9b:2b:f0:85:
                    19:5b:8f:5c:70:33:ac:75:27:44:ef:63:c9:32:f0:
                    37:89:27:db:75:40:54:19:3f:bd:32:78:79:51:9b:
                    11:12:0b:57:c4:a1:e3:34:fe:76:3a:ee:fc:4f:de:
                    ea:85:dc:f4:15:17:2d:2b:b5:92:bd:04:f7:f0:b3:
                    bf:10:49:68:dd:d5:c5:69:15:df:44:a1:bd:0e:2c:
                    04:5b:e2:f6:81:6d:9a:19:7a:32:86:d8:03:b6:8d:
                    f7:34:0a:20:f6:8c:de:7f:0a:cb:87:bb:36:f1:2d:
                    3b:89
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                A2:FC:29:06:A5:DE:B6:67:3C:EF:3F:FF:6A:56:3F:56:1F:A6:6F:C7
    Signature Algorithm: sha256WithRSAEncryption
         05:af:b4:29:5d:45:42:a5:6c:4e:91:28:9f:c0:81:23:7f:58:
         e8:21:fb:d4:1c:0c:dc:f1:df:d4:bc:00:58:d6:3c:42:34:1d:
         26:ca:9f:59:9f:85:03:66:f7:bf:74:d6:b1:64:68:5f:df:7d:
         9d:39:c9:92:f9:34:5d:a9:46:eb:ee:7e:6e:01:dc:07:15:5e:
         4d:f3:2f:0d:06:8c:df:65:76:1f:72:dc:28:68:d5:7b:43:a5:
         2a:81:8b:a0:b8:ce:f7:16:93:ba:ec:5b:40:c3:9e:7f:06:cd:
         3a:6f:08:d2:c1:29:3a:54:77:c1:eb:5e:c5:00:27:71:f3:e5:
         75:52:1c:ce:76:9c:b5:79:33:fc:99:32:3e:81:ca:50:43:1c:
         5e:d9:2e:70:cb:f8:17:88:14:76:94:b1:57:52:3e:99:89:1c:
         02:9b:d0:e1:1b:fd:84:73:fe:3f:dc:32:e1:0a:fd:9e:d3:5b:
         72:17:60:ca:35:97:5b:ee:47:5c:e3:58:c7:53:ec:88:b7:35:
         a0:ba:c4:99:00:0b:f0:3e:1a:cc:d1:16:4f:59:67:c1:45:4b:
         96:91:04:ea:37:9a:fa:64:cc:21:2c:6d:a2:94:ac:34:1e:05:
         cb:be:66:b7:e3:64:0a:b8:8f:59:94:48:62:45:9a:7e:21:9c:
         55:b8:03:dd
```

```
F:\Desktop\1_1\META-INF>keytool -printcert -file CERT.RSA
所有者: O=sspai
发布者: O=sspai
序列号: 336922eb
有效期为 Sat Apr 28 23:35:12 CST 2018 至 Wed Apr 22 23:35:12 CST 2043
证书指纹:
         MD5:  D6:EA:E7:F4:0C:85:99:6E:CA:31:D1:52:66:7B:52:93
         SHA1: 5F:3C:31:86:A9:97:4F:55:66:18:28:31:08:C9:E2:3E:C0:08:76:EA
         SHA256: F9:24:05:E5:9E:64:5F:88:FA:04:E6:D2:4C:ED:E1:21:94:42:8D:5B:8B:CE:A0:7E:A6:22:F0:31:FA:A0:32:8E
签名算法名称: SHA256withRSA
主体公共密钥算法: 2048 位 RSA 密钥
版本: 3

扩展:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: A2 FC 29 06 A5 DE B6 67   3C EF 3F FF 6A 56 3F 56  ..)....g<.?.jV?V
0010: 1F A6 6F C7                                        ..o.
]
]
```

```
F:\Desktop\1\META-INF>keytool -printcert -file CERT.RSA
所有者: O=sspai
发布者: O=sspai
序列号: 336922eb
有效期为 Sat Apr 28 23:35:12 CST 2018 至 Wed Apr 22 23:35:12 CST 2043
证书指纹:
         MD5:  D6:EA:E7:F4:0C:85:99:6E:CA:31:D1:52:66:7B:52:93
         SHA1: 5F:3C:31:86:A9:97:4F:55:66:18:28:31:08:C9:E2:3E:C0:08:76:EA
         SHA256: F9:24:05:E5:9E:64:5F:88:FA:04:E6:D2:4C:ED:E1:21:94:42:8D:5B:8B:CE:A0:7E:A6:22:F0:31:FA:A0:32:8E
签名算法名称: SHA256withRSA
主体公共密钥算法: 2048 位 RSA 密钥
版本: 3

扩展:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: A2 FC 29 06 A5 DE B6 67   3C EF 3F FF 6A 56 3F 56  ..)....g<.?.jV?V
0010: 1F A6 6F C7                                        ..o.
]
]
```

Android 中 certificate 是如何生成的?

生成密钥和密钥库

您可以使用 Android Studio 生成应用签名或上传密钥，步骤如下:

1. 在菜单栏中，点击 **Build** > **Generate Signed APK**。
2. 从下拉菜单中选择一个模块，然后点击 **Next**。
3. 点击 **Create new** 以创建一个新密钥和密钥库。
4. 在 **New Key Store** 窗口上，为您的密钥库和密钥提供以下信息，如图 3 所示。

   密钥库

   - **Key store path**：选择创建密钥库的位置。
   - **Password**：为您的密钥库创建并确认一个安全的密码。

   密钥

   - **Alias**：为您的密钥输入一个标识名。
   - **Password**：为您的密钥创建并确认一个安全的密码。此密码应当与您为密钥库选择的密码不同
   - **Validity (years)**：以年为单位设置密钥的有效时长。密钥的有效期应至少为 25 年，以便您可以在应用的整个生命周期内使用相同的密钥签署应用更新。
   - **Certificate**：为证书输入一些关于您自己的信息。此信息不会显示在应用中，但会作为 APK 的一部分包含在您的证书中。

   填写完表单后，请点击 **OK**。

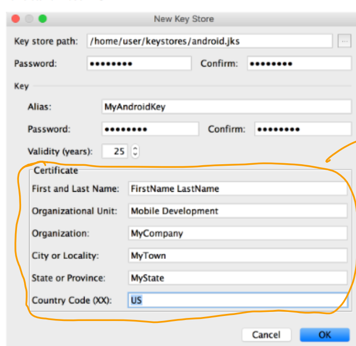


图 3. 在 Android Studio 中创建新的密钥库。

→ 证书是密钥库中的一把密钥对应

5. 如果您想生成一个使用您的新密钥签署的 APK，则继续转到手动签署 APK，如果您只想生成一个密钥和密钥库而不签署 APK，则点击 **Cancel**。
6. 如果您想选择使用 Google Play 应用签名，则转到管理您的应用签名密钥，并按照说明设置 Google Play 应用签名。

手动签署 APK

您可以使用 Android Studio 手动生成签署的 APK，既可以一次生成一个 APK，也可以一次为多个构建变体生成 APK。除了手动签署 APK 外，您还可以将 Gradle 构建设置配置为在构建流程期间自动处理签署。本部分将介绍手动签署流程。如需详细了解如何在构建流程中签署应用，请参阅配置构建流程以自动签署您的 APK。

要在 Android Studio 中手动签署用于发布的 APK，请按以下步骤操作：

1. 点击 **Build > Generate Signed APK** 以打开 **Generate Signed APK** 窗口。（如果您刚刚按上述说明生成了一个密钥和密钥库，则此窗口已处于打开状态。）
2. 在 **Generate Signed APK Wizard** 窗口上，选择一个密钥库和一个私钥，并为它们输入密码。（如果您刚刚在上一部分中创建密钥库，这些字段将自动填充。）然后，点击 **Next**。

> **注**：如果您使用 Google Play 应用签名，您应在此处指定您的上传密钥。如果改为自行管理您的应用签名密钥和密钥库，您应指定您的应用签名密钥。如需了解详细信息，请参阅上文的管理您的密钥。

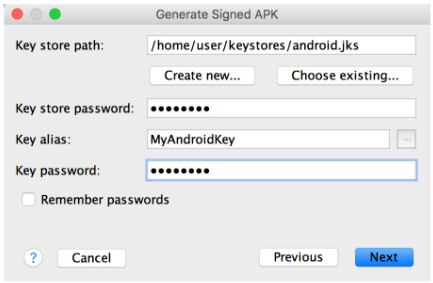


图 4. 在 Android Studio 中选择一个私钥。

3. 在下一个窗口上，为签署的 APK 选择一个目的地、选择构建类型、选择产品风味（如果适用），然后点击 **Finish**。

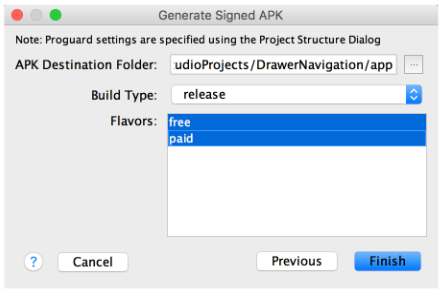


图 5. 为已选择的产品风味生成签署的 APK。

> **注**：如果您的项目使用多个产品风味，那么在 Windows/Linux 上按住 **Ctrl** 键或者在 Mac OS X 上按住 **Command** 键可以选择多个产品风味。Android Studio 将为选择的每个产品风味生成单独的 APK。

在流程完成后，您会在上面选择的目标文件夹中找到签署的 APK。现在，您可以通过像 Google Play 商店一样的应用市场或者您选择的机制分发签署的 APK。要详细了解如何将签署的 APK 发布到 Google Play 商店，请参阅开始发布。要详细了解其他分发选项，请阅读替代分发选项。

为了确保用户将更新成功安装到您的应用中，您需要在应用的整个生命周期内使用相同的证书签署 APK。要详细了解使用相同密钥签署所有应用的各种好处，请参阅下面的签署注意事项。如需了解有关保护您的私钥和密钥库的详细信息，请参阅下文的保护您的密钥。

签署注意事项

在应用的预期生命周期内，您应使用相同证书签署所有 APK。这么做的原因有多个：

- 应用升级：当系统安装应用的更新时，它会比较新版本和现有版本中的证书。如果证书匹配，则系统允许更新。如果您使用不同的证书签署新版本，则必须为应用分配另一个软件包名称 - 在此情况下，用户将新版本作为全新应用安装。
- 应用模块化：Android 允许通过相同证书签署的多个 APK 在同一个进程中运行（如果应用请求这样），以便系统将它们视为单个应用。通过此方式，您可以在模块中部署您的应用，且用户可以独立更新每个模块。
- 通过权限共享代码/数据：Android 提供基于签名的权限执行，以便应用可以将功能展示给使用指定证书签署的另一应用。通过使用同一个证书签署多个 APK 并使用基于签名的权限检查功能，您的应用可采用安全的方式共享代码和数据。

如果您计划支持升级应用，请确保您的应用签名密钥的有效期超出该应用的预期生命周期。建议有效期为 25 年或以上。当密钥有效期过期后，用户将不能再无缝升级到应用的新版本。

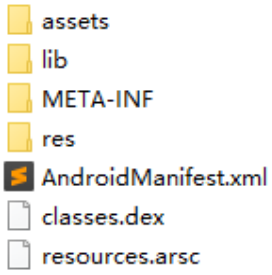
如果您计划在 Google Play 上发布您的应用，您用于签署这些 APK 的密钥的有效期必须在 2033 年 10 月 22 日以后结束。Google Play 强制执行此要求，以确保在新版本可用时，用户可以无缝升级应用。如果您使用 Google Play 应用签名，则 Google 可确保您的应用正确签署，并能够在它们的整个生命周期内接收更新。

Android应用签名验证过程中，满足以下条件才能安装应用：

1、SHA-1(除META-INF目录外的文件) == MANIFEST.MF中的各SHA-1值；

2、(SHA-1 + Base64)(MANIFEST.MF文件及各子项) == CERT.SF中各值

3、公钥（CERT.SF） == CERT.RSA/DSA对SF文件的签名



这里所说的 APK 签名文件是指解压 .apk 文件所得文件夹中的 META-INF 文件夹。

主要包含 3 个文件：

- MANIFEST.MF
- CERT.SF
- CERT.RSA

- MANIFEST.MF

```
Manifest-Version: 1.0
Built-By: Generated-by-ADT
Created-By: Android Gradle 2.3.3

Name: assets/xrays/highlightyellow.xray
SHA1-Digest: ln/ReK6pDSXc7uNlXbpz87dNh2g=

Name: res/drawable-xhdpi-v4/import_studio_white.png
SHA1-Digest: /GnG1Wa36srqJNzjhDk3pb75sv0=

Name: res/drawable-xhdpi-v4/tools_crop_4_3_selected.png
SHA1-Digest: MK1HkmkqQsmHSUJ/Xicozh9nLvc=

Name: lib/armeabi-v7a/libRSSupport.so
SHA1-Digest: r7TP5nKKXJMnDkqx5gdw0gX9Vcc=
```

APK 内所有源文件 SHA1 值的 base64 编码。

- CERT.SF

```
Signature-Version: 1.0
X-Android-APK-Signed: 2
SHA1-Digest-Manifest: MhhtfSZbIzKaUOfIrgdLdOvl9IY=
Created-By: 1.0 (Android)

Name: assets/xrays/highlightyellow.xray
SHA1-Digest: ln/ReK6pDSXc7uNlXbpz87dNh2g=

Name: res/drawable-xhdpi-v4/import_studio_white.png
SHA1-Digest: /GnG1Wa36srqJNzjhDk3pb75sv0=

Name: res/drawable-xhdpi-v4/tools_crop_4_3_selected.png
SHA1-Digest: MK1HkmkqQsmHSUJ/Xicozh9nLvc=

Name: lib/armeabi-v7a/libRSSupport.so
SHA1-Digest: r7TP5nKKXJMnDkqx5gdw0gX9Vcc=
```

SHA1-Digest-Manifest：MANIFEST.MF 文件的 SHA1 值的 base64 编码；

MANIFEST.MF 中每条内容的 SHA1 值的 base64 编码。

- CERT.RSA

CERT.RSA 存储 CERT.SF 的数字签名及其签名证书。