

Androidアプリへの署名

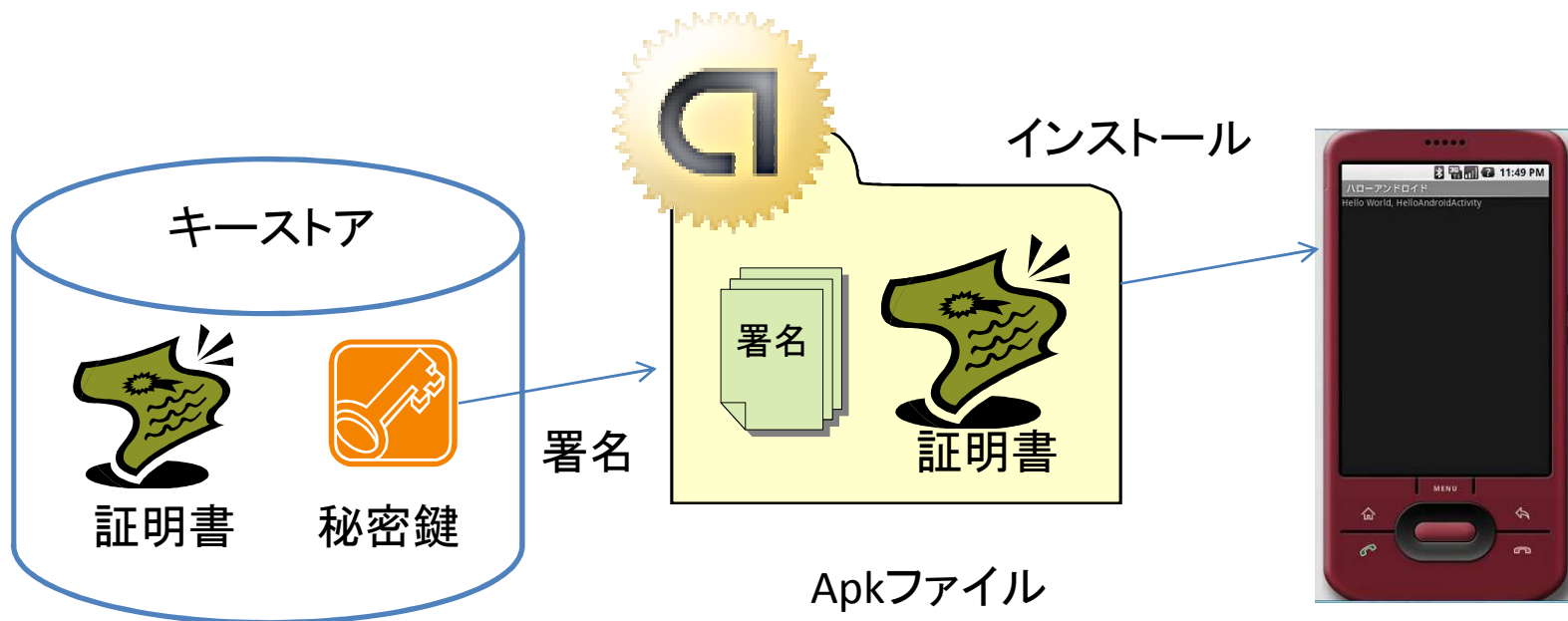
2008/11/10 (V1.01)

日本Androidの会

木南英夫

アプリには署名が必要

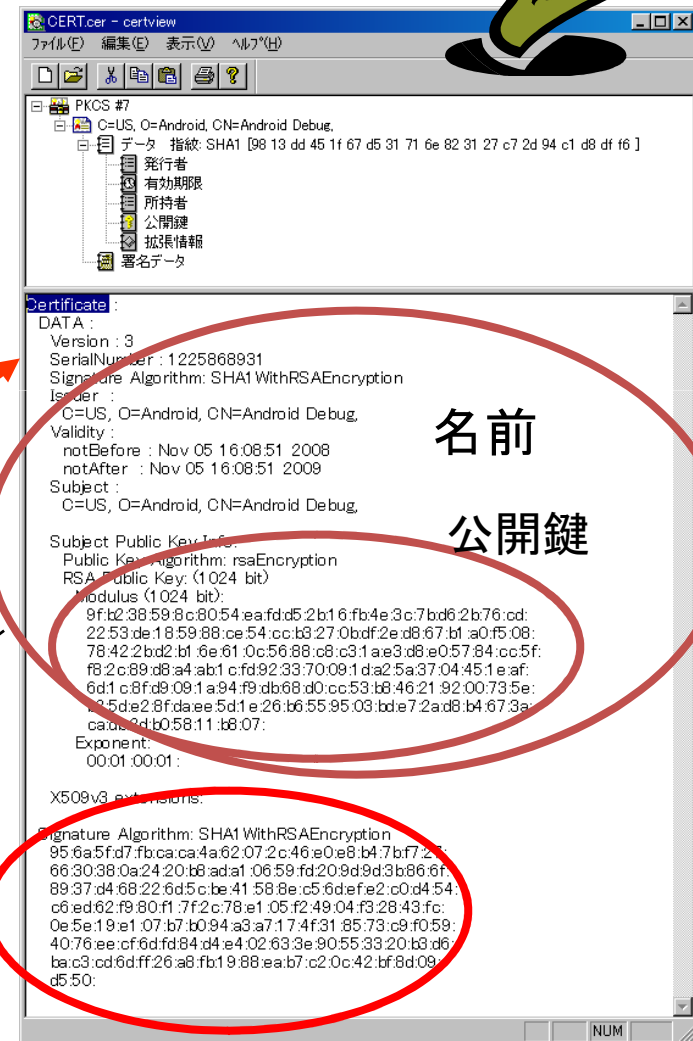
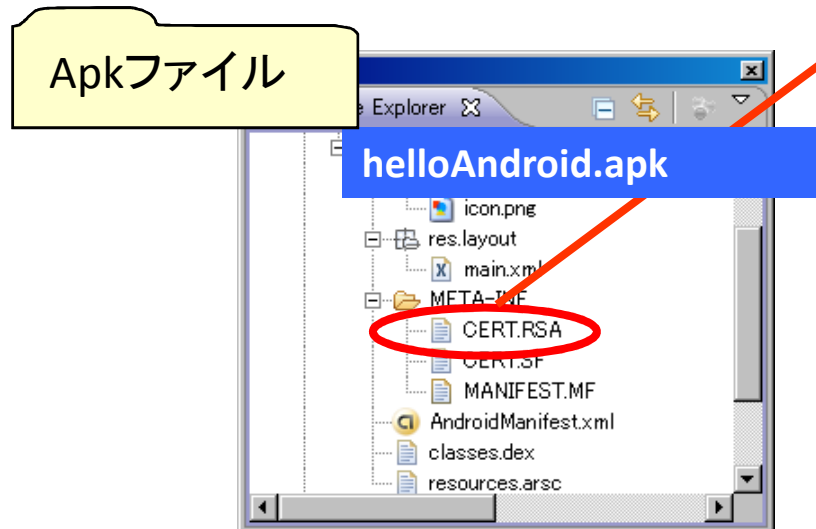
- アプリケーションパッケージに、秘密鍵と公開鍵を用いた電子署名をしないとインストールできない
- キーストアには、証明書と秘密鍵が格納されている
- Apkファイルは、jar形式で、ファイルごとの署名が添付される



自己署名の証明書

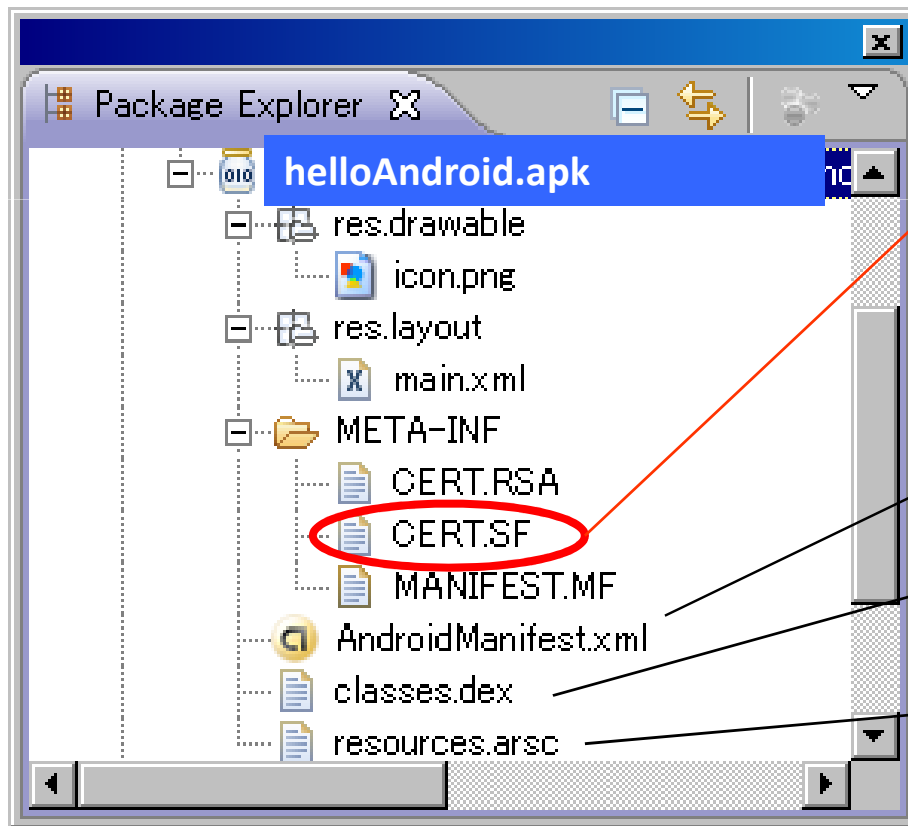


- 証明書は、アプリ配布用のapkファイルに添付される
- 証明書内には、名前のほかに、公開鍵が記述されている
- 認証機関(CA)の署名は不要
 - 名刺程度の信用力



署名

- Jarsignerというツールで、内部のファイルごとに署名



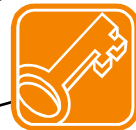
Name: res/drawable/icon.png
SHA1-Digest: N...

Name: res/layout/main.xml
SHA1-Digest: fTpg9iXHoCM...

Name: AndroidManifest.xml
SHA1-Digest: djW...

Name: classes.dex
SHA1-Digest: dbgeV/...

Name: resources.arsc
SHA1-Digest: 60l+...



証明書の作り方

- Java SDK付属のKeytoolで作成
 - 有効期間は25年上 (10000日)
- 設定する名前は自分の識別ができる名前にする

```
C:¥>keytool -genkey -keystore kinami.keystore -validity 10000 -alias hkinami
```

キーストアのパスワードを入力してください:

姓名を入力してください。

[Unknown]: Hideo Kinami

組織単位名を入力してください。

[Unknown]: Study working group

組織名を入力してください。

[Unknown]: Japan Android group

都市名または地域名を入力してください。

[Unknown]: Akihabara

州名または地方名を入力してください。

[Unknown]: Tokyo

この単位に該当する 2 文字の国番号を入力してください。

[Unknown]: jp

キーストア

kinami.keystore

エイリアス

hkinami



エイリアス

Yamada



apkファイルに署名する

- プロジェクトの右クリックからAndroid Toolを選ぶ
- Export unsigned application packageを選択
- jarsignerで署名する。キーストアとエイリアスを指定する

D:¥home>jarsigner -verbose -keystore c:¥kinami.keystore HelloAndroid.apk hkinami

キーストアのパスワードを入力してください:

追加中: META-INF/MANIFEST.MF

追加中: META-INF/HKINAMI.SF

追加中: META-INF/HKINAMI.DSA

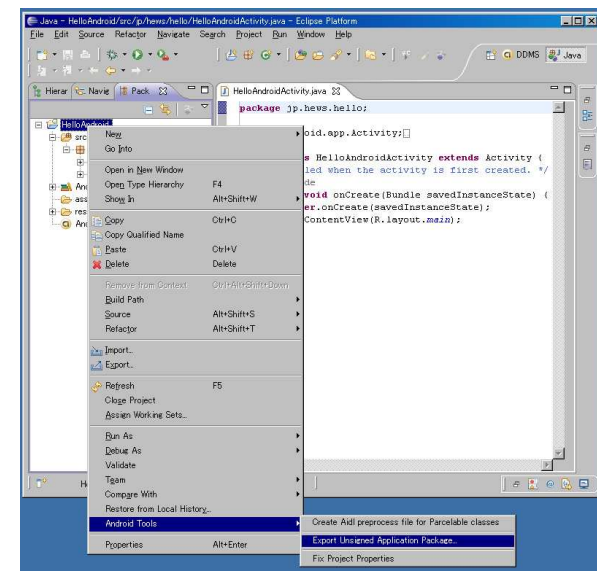
署名中: res/drawable/icon.png

署名中: res/layout/main.xml

署名中: AndroidManifest.xml

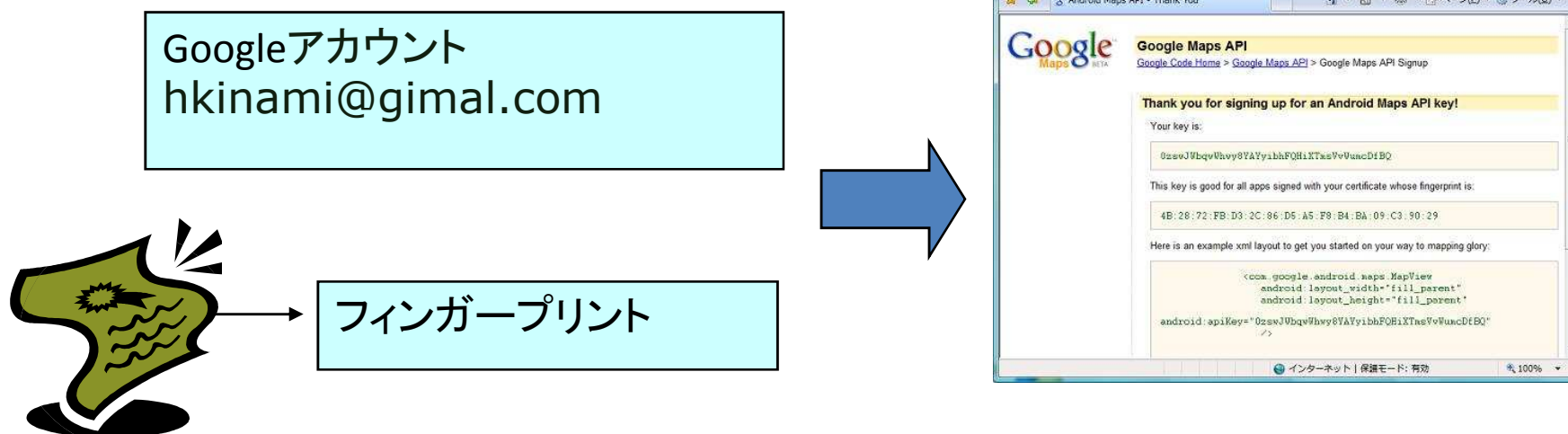
署名中: resources.arsc

署名中: classes.dex



Map Keyを取得する

- 使用する証明書のハッシュを、Google IDと関連付けて登録する
 - <http://code.google.com/android/maps-api-signup.html>
 - <http://www.android-group.jp/index.php?勉強会/MapsAPIKey>



取得したMapKeyを使用する

- MapViewの生成時に、XMLやプログラムで指定する

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <com.google.android.maps.MapView
        android:id="@+id/mapview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:enabled="true"
        android:clickable="true"
        android:apiKey=<取得したMaps API Key>
    />
</LinearLayout>
```


証明書のフィンガープリント

- デバック用と、リリース用は別のキーストアー
 - keytool -list キーストア
 - デバック用のPC毎に必要

キーストア
debug.keystore



- Keystore name — "debug.keystore"
- Keystore password — "android"
- Key alias — "androiddebugkey"
- Key password — "android"
- CN — "CN=Android Debug,O=Android,C=US"

OS X and Linux:

~/.android/debug.keystore

Windows Vista:

C:¥Users¥%USERNAME%¥AppData¥Local¥Android¥debug.keystore

Windows XP:

C:¥Documents and Settings¥%USERNAME%¥Local Settings¥Application Data¥Android¥debug.keystore

証明書の用途

- 証明書の有効期間の確認はインストール時だけ
- アプリケーションのバージョンアップのためには、同じ証明書が必要
 - 期限がきたら、別アプリにしなければならない
- 署名が同じ複数のアプリケーションを同じプロセスで実行できる
- 署名ベースのパーミッションによって、複数のアプリケーションでコードやデータを共有できる

証明書の利用範囲

- 開発者の立場では、原理だけ理解しておけば、あとは手順通りに署名するだけ
- 利用者の立場では、実際にアプリケーションを利用する場合には、署名が表していること(自己証明の証明書)を理解しておく必要がある。
- アプリを流通させる立場では、開発者とどのように信頼関係を構築していくかが重要。
 - Map keyの方式は、今後、開発者情報を管理する上で、応用が可能