

Android保存私密信息—强大的keyStore（译）



wutongke (/u/0e0821e94979) + 关注

2017.02.25 00:14* 字数 1118 阅读 4778 评论 2 喜欢 55

(/u/0e0821e94979)



这里讨论下如何使用**Android Keystore** (<https://link.jianshu.com?t=https://developer.android.com/training/articles/keystore.html>)保存密码等敏感信息，如何加密、解密数据。

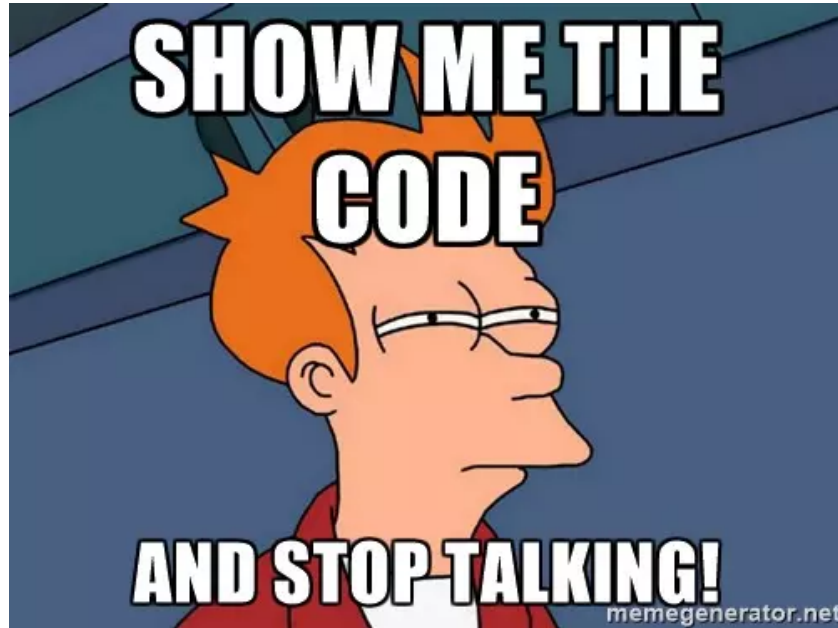
在开始讨论之前我们先搞清楚一些基础知识。**Keystore**不只是可以保存密码，还可以保存敏感数据，而且它的实现方式使得黑客或者恶意程序很难破信息。

Android的Keystore系统可以把密钥保持在一个难以从设备中取出数据的容器中。当密钥保存到**Keystore**之后，可以在不取出密钥的状态下进行私密操作。此外，它提供了限制何时以何种方式使用密钥的方法，比如使用密钥时需要用户认证或限制密钥只能在加密模式下使用。

一个应用程式只能编辑、保存、取出自己的密钥。这个概念很简单，但是功能很强大。**App**可以生成或者接收一个公私密钥对，并存储在**Android的Keystore**系统中。公钥可以用于在应用数据放置到特定文件夹前对数据进行加密，私钥可以在需要的时候解密相应的数据。



如果你只是想看代码，可以直接点击[这里 \(https://link.jianshu.com?t=https://github.com/wutongke/KeyStoreDemo\)](https://link.jianshu.com?t=https://github.com/wutongke/KeyStoreDemo)。



(/apps/
utm_sc
banner

简单起见，我写了一个demo演示如何使用Android Keystore保存密码，加密、显示加密形式以及解密。

这里我就不写xml了，都是一些简单的东西，我在文后贴出所有的代码。

我这里新建了2个类文件。一个是EnCryptor，另一个是Decryptor。通过名字很容易知道其功能。

创建新密钥

在开始编码之前，我们需要给加密/解密数据的别名进行命名，名字可以是任意字符串，但是不可以是空字符串。别名是显示在Android Keystore中生成的密钥的名字。

首先我们需要获取Android KeyGenerator (<https://link.jianshu.com?t=https://developer.android.com/reference/javax/crypto/KeyGenerator.html>)的实例：

```
final KeyGenerator keyGenerator = KeyGenerator
    .getInstance(KeyProperties.KEY_ALGORITHM_AES, "AndroidKeyStore");
```

这里我们设置使用KeyGenerator的生成的密钥加密算法是AES，我们将在AndroidKeyStore中保存密钥 / 数据。

接下来我们能使用KeyGenParameterSpec.Builder (<https://link.jianshu.com?t=https://developer.android.com/reference/android/security/keystore/KeyGenParameterSpec.Builder.html>) 创建KeyGenParameterSpec (<https://link.jianshu.com?t=https://developer.android.com/reference/android/security/keystore/KeyGenParameterSpec.html>)，传递给KeyGenerators的init方法。



KeyGenParameterSpec是什么，可以把它当作我们要生成的密钥的参数。例如，我们需要给密钥设置一个特定的过期时间。

KeyGenParameterSpec的代码：

```
final KeyGenerator keyGenerator = KeyGenerator
    .getInstance(KeyProperties.KEY_ALGORITHM_AES, ANDROID_KEY_STORE);

final KeyGenParameterSpec keyGenParameterSpec = new
    KeyGenParameterSpec.Builder(alias,
        KeyProperties.PURPOSE_ENCRYPT | KeyProperties.PURPOSE_DECRYPT)
        .setBlockModes(KeyProperties.BLOCK_MODE_GCM)
        .setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_NONE)
        .build();
```

首先我们传递了一个别名，这个名字可以是任意的，之后我们设置意图，是加密还是解密数据。

setBlockMode保证了只有指定的block模式下可以加密，解密数据，如果使用其它的block模式，将会被拒绝。可以在这里 (<https://link.jianshu.com?t=https://developer.android.com/reference/android/security/keystore/KeyProperties.html>)查看不同的block模式。

我们使用了“AES/GCM/NoPadding”变换算法，还需要设置KeyGenParameterSpec的padding类型。

加密数据

以上的执行完之后，加密数据非常简单：

```
final KeyGenerator keyGenerator = KeyGenerator
    .getInstance(KeyProperties.KEY_ALGORITHM_AES, ANDROID_KEY_STORE);

final KeyGenParameterSpec keyGenParameterSpec = new
    KeyGenParameterSpec.Builder(alias,
        KeyProperties.PURPOSE_ENCRYPT | KeyProperties.PURPOSE_DECRYPT)
        .setBlockModes(KeyProperties.BLOCK_MODE_GCM)
        .setEncryptionPaddings(KeyProperties.ENCRYPTION_PADDING_NONE)
        .build();

keyGenerator.init(keyGenParameterSpec);
final SecretKey secretKey = keyGenerator.generateKey();

final Cipher cipher = Cipher.getInstance("AES/GCM/NoPadding");
cipher.init(Cipher.ENCRYPT_MODE, secretKey);
```

首先我们使用keyGenParameterSpec初始化KeyGenerator，之后我们生成了**SecretKey** (<https://link.jianshu.com?t=https://developer.android.com/reference/javax/crypto/SecretKey.html>)。

(/apps/
utm_sc
banner



现在我们有**secretkey**，我们可以初始化**Cipher** (<https://link.jianshu.com?t=https://developer.android.com/reference/javax/crypto/Cipher.html>) 对象，这将是实际的加密过程。我们需要设置Cipher编码类型：

(/apps/
utm_sc
banner

之后我们有了**ciphers initialization vector (IV)**的引用，可以用于解密。我们使用 `doFinal(textToEncrypt)` 拿到了最终的编码， `doFinal(textToEncrypt)` 返回的就是最重的加密数据。

解密

获取KeyStore实例：

我们用keyStore获取我们的secret key，我们还需要一个SecretKeyEntry：

之前KeyGenParameterSpecs中设置的block模式是 `KeyProperties.BLOCK_MODE_GCM`，所以这里只能使用这个模式解密数据。



我们需要为GCMParameterSpec (<https://link.jianshu.com?t=https://developer.android.com/reference/javax/crypto/spec/GCMParameterSpec.html>) 指定一个认证标签长度（可以是128、120、112、104、96这个例子中我们能使用最大的128），并且用到之前的加密过程中用到的IV。

(/apps/
utm_sc
banner

获取加密数据：

获取解密数据：

这就是整个过程了。



(/apps/
utm_sc
banner

源码地址：**HERE** (<https://link.jianshu.com?t=https://github.com/wutongke/KeyStoreDemo>)

原文地址：<https://medium.com/@josiassena/using-the-android-keystore-system-to-store-sensitive-information-3a56175a454b#.3lly5mk5i> (<https://link.jianshu.com?t=https://medium.com/@josiassena/using-the-android-keystore-system-to-store-sensitive-information-3a56175a454b#.3lly5mk5i>)

推荐阅读：

重要-作为Android开发者必须了解的Gradle知识
(<https://www.jianshu.com/p/c31513f5f550>)

编写高效的Android代码（译） (<https://www.jianshu.com/p/d8f2eab43e4a>)



Android中使用gradient的一条建议 (<https://www.jianshu.com/p/508d1cf8fb61>)

寻找卓越的（Android）软件工程师 (<https://www.jianshu.com/p/3615c18539bc>)

如果觉得我的文章对您有用，请随意打赏。您的支持将鼓励我继续创作！

赞赏支持

(/apps/
utm_sc
banner

 Android (/nb/6315168) 举报文章 © 著作权归作者所有



wutongke (/u/0e0821e94979) ♂

写了 61345 字，被 1785 人关注，获得了 3097 个喜欢
(/u/0e0821e94979)

+ 关注

Hi西木

喜欢 | 55



更多分享

开发10年
全记在这本Java进阶宝典了

Spring源码分析

分布式架构

微服务架构

JVM性能优化

高效DevOps


多线程并发编程


点击领取




(/p/428251ede1aa)

被以下专题收入，发现更多相似内容





- 

深入浅出And... (/c/3441588a59f5?
utm_source=desktop&utm_medium=notes-included-collection)
- 

密码学 (/c/2e88eebcc2c7?utm_source=desktop&utm_medium=notes-
included-collection)
- 

安卓资料汇总 (/c/aa7c48dea7fb?
utm_source=desktop&utm_medium=notes-included-collection)




-  Android知识 (/c/3fde3b545a35?utm_source=desktop&utm_medium=notes-included-collection)
-  Android开发 (/c/d1591c322c89?utm_source=desktop&utm_medium=notes-included-collection)
-  程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)
-  Android... (/c/fb738dba3952?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

(/apps/
utm_sc
banner


掘金 Android 文章精选合集 (/p/5ad013eb5364?utm_campaign=maleski...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？ 从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？ ... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...

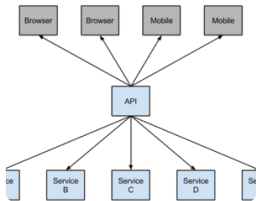
 掘金官方 (/u/5fc9b6410f4f?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommenc

Android - 收藏集 (/p/dad51f6c9c4d?utm_campaign=maleskine&utm_c...

用两张图告诉你，为什么你的 App 会卡顿？ - Android - 掘金 Cover 有什么料？ 从这篇文章中你能获得这些料：知道setContentView()之后发生了什么？ ... Android 获取 View 宽高的常用正确方式，避免为零 - 掘金...

 passiontim (/u/e946d18f163c?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommenc

(/p/46fd0faecac1?




utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommenc
Spring Cloud (/p/46fd0faecac1?utm_campaign=maleskine&utm_conte...

Spring Cloud为开发人员提供了快速构建分布式系统中一些常见模式的工具（例如配置管理，服务发现，断路器，智能路由，微代理，控制总线）。分布式系统的协调导致了样板模式，使用Spring Cloud开发人员可...

 卡卡罗2017 (/u/d90908cb0d85?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommenc

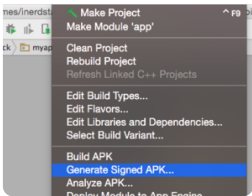
加解密详解 (/p/b3be35c1d424?utm_campaign=maleskine&utm_conten...

本文主要介绍移动端的加解密算法的分类、其优缺点特性及应用，帮助读者由浅入深地了解和选择加解密算法。文中会包含算法的关键代码，以利于读者理解使用。算法分类 根据加密结果是否可以被解密，算法可...

 voyagelab (/u/c2e1b9b2e28e?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommenc




(/p/644ddb6e3d9c?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner

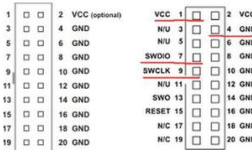
Android Keystore漫谈 (/p/644ddb6e3d9c?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner)

写在前面 今天使用高德地图为应用添加Key的时候，发现有一项需要用到安全码SHA1，而SHA1存在于Keystore中，遂简单地了解了一下Keystore。虽然之前实习开发中有用同事生成的Keystore对应用加过密...

 代码咖啡 (/u/9b87ba5fc959?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend_banner


(/p/73a2dc39684a?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner

Jlink接口的JTAG和SWD接口定义 (/p/73a2dc39684a?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner)

下面是标准的接口排列： Jlink仿真器接口：

 小狗乖乖汪汪叫 (/u/07a7541072c1?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend_banner


(/p/8aca535312ff?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner

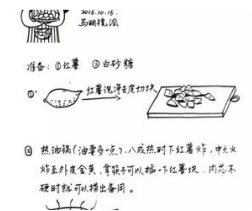
砖窑 (/p/8aca535312ff?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner)

老刘，56岁，陕西镇安人，老两口一起在砖窑干活，老板按计件，两个人每月各能拿到手2千多。白师傅，43岁，窑工，陕西镇安人，脸部有烫伤。主要在窑顶拉煤到每个窑孔处，由窑头根据火候，从窑孔往砖窑...

 琼钧 (/u/ac2453a75f36?)


utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend_banner

(/p/e95936acf869?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner

手绘：拔丝红薯步骤图 (/p/e95936acf869?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend_banner)

 明镜123 (/u/00ad92f7fb76?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend_banner


(/p/cd86831294d6?



(/apps/
utm_sc
banner

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommend
手机上那些相见恨晚的**APP** (/p/cd86831294d6?utm_campaign=maleskin...

超级好用的APP

 诗和远方_的田野 (/u/2f22124e9db6?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommend

