File IO项目实战—dex文件加密

T H A N K Y O U F O R W A T C H I N G

主讲老师Alvin：
2464061231

讲师简介

Alvin

华南理工大学 软件工程 工程硕士

三星中国研究院 5 years
项目经理

小米科技 2 years
技术总监

- 曾就业于三星中国研究院及小米旗下互联网公司担任android任软件工程师及项目经理
- 拥有扎实的C/Java 基础，深入研究android系统多年。
- 讲课形象生动，热情洋溢

目 录

CONTENTS

APK文件反编译

什么是反编译
如何防止反编译
Apk文件的基本构造

Apk加固的方案原理

Apk加固总体架构
Apk打包基本流程
Dex文件的意义
热修复原理解析

AES加密项目实战

Apk加固项目实战
Apk脱壳技术实战

课程总结

课程技术总结
交流互动

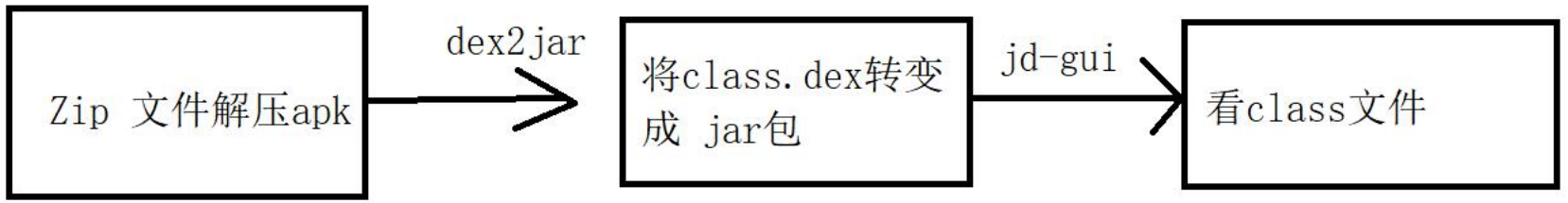
01 反编译

什么是反编译？（what ）

- 定义：利用编译程序从源语言编写的源程序产生目标程序的过程。

怎么进行反编译？（ hov

- 先了解apk的文件构造



01 加固方案的手段

反模拟器

模拟器运行apk，可以用模拟器监控到apk的各种行为，所以在实际的加固apk运行中，一旦发现模拟器在运行该APK，就停止核心代码的运行。

代码虚拟化

代码虚拟化在桌面平台应用保护中已经是非常的常见了，主要的思路是自建一个虚拟执行引擎，然后把原生的可执行代码转换成自定义的指令进行虚拟执行。

加密

样本的部分可执行代码是以压缩或者加密的形式存在的，比如，被保护过的代码被切割成多个小段，前面的一段代码先把后面的代码片段在内存中解密，然后再去执行解密之后的代码，如此一块块的迭代执行。

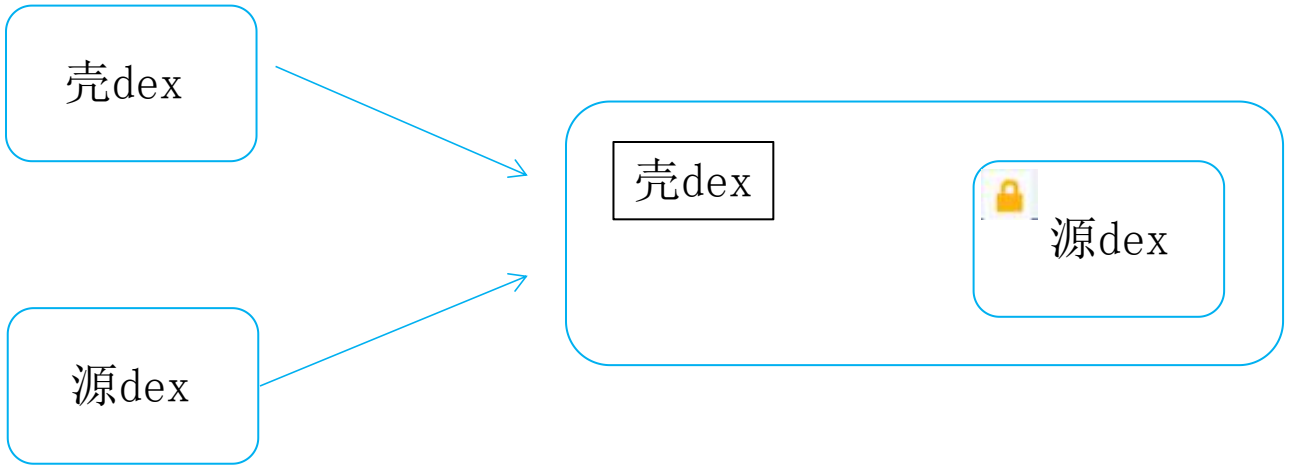
01 加固方案总体思想

一个程序员的故事：

辛辛苦苦找到一个对象，结婚后发现是个母夜叉。不给管钱就闹，晚上睡觉她趴着睡，导致这程序员无法去洗脚了。然而这个程序员很努力，平时除了上班，还能够做点外包，赚点外快。所以他就想到了把工资卡上交，而把赚到的外快放到了自己的小金库。从此过上了性福生活。

01 加固的方案

加固



目 录

CONTENTS

APK文件反编译

什么是反编译
如何防止反编译
Apk文件的基本构造

Apk加固的方案原理

Apk加固总体架构
apk签名是什么
Dex文件加密的基本
方案框架

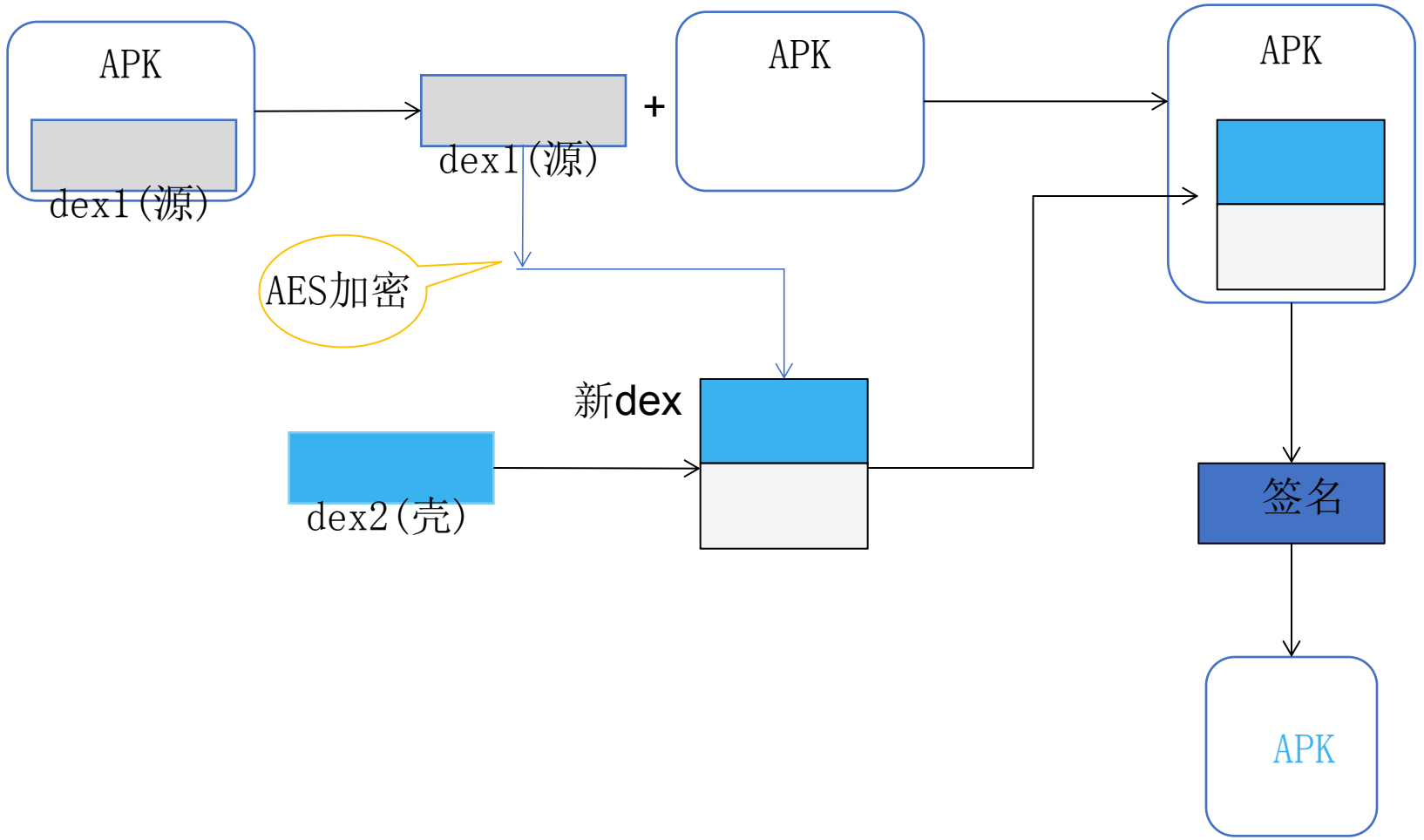
Apk加固项目实战

Dex文件基本结构
Dex文件的修改原则
Apk加固项目实战
Apk脱壳技术实战

课程总结

课程技术总结
交流互动

03 加固总体框架



04 问题

- dex文件可以随便拼凑吗？
- 壳dex 怎么来的
- 如何签名？
- 如何运行新apk（如何脱壳）？

02 基本原理

- Dex文件是什么？
- Apk打包流程
- Dex文件加载流程

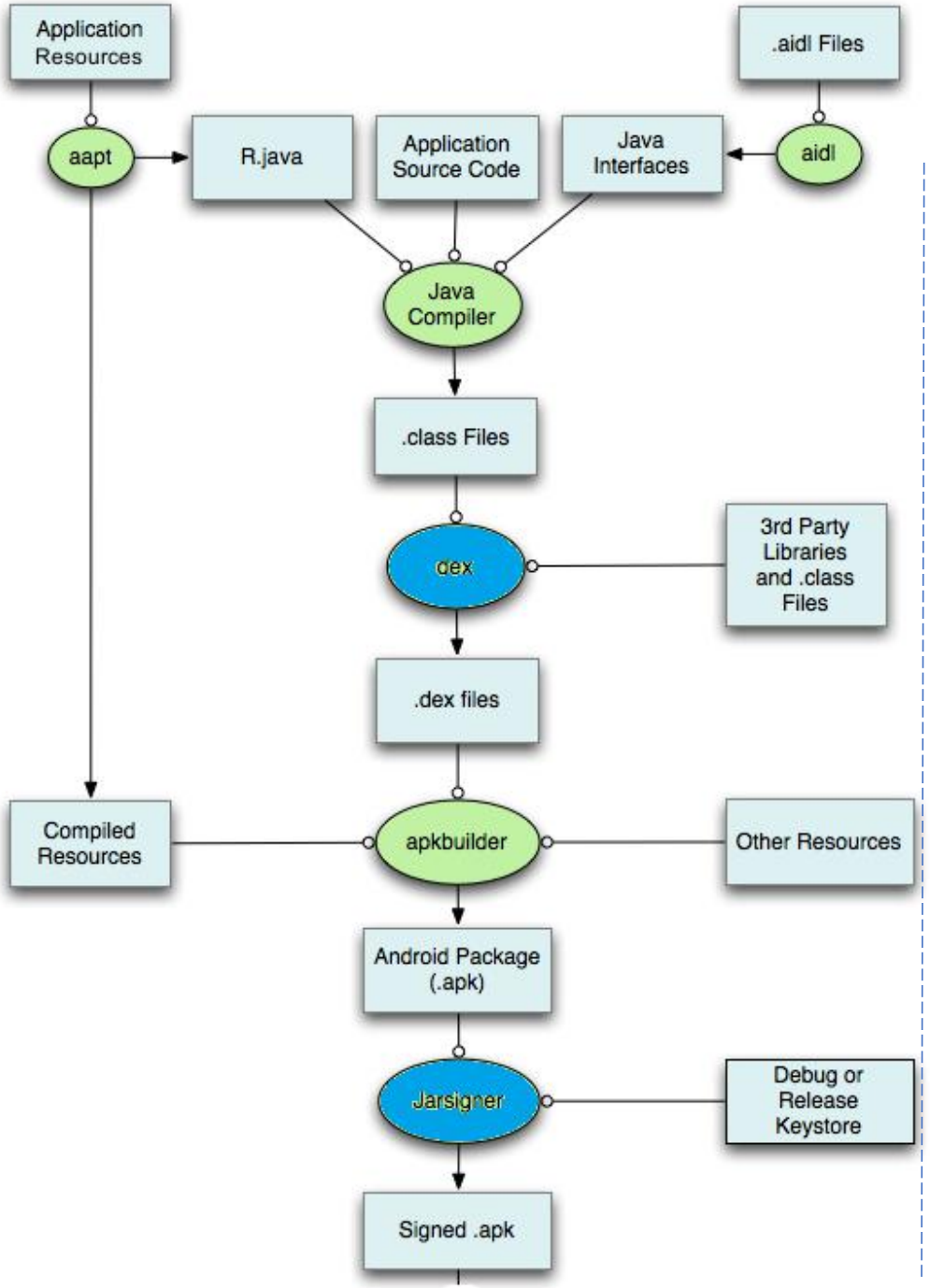
加固的目的是保护dex，直接而言就是对dex文件进行操作，对dex文件动刀子，必须知道dex文件是什么，能否直接动刀子。什么是源dex？什么是壳dex？

加壳是在原来apk的基础上加一层保护壳，dex文件修改了就需要重新打包，否则apk安装不了。这就需要我们详细学习apk如何打包的，

加壳后的文件是不能直接用的，dex文件是加密的，所以我们需要对他进行解密，解密后的dex文件如何加载？

02 Dex文件


```
struct header_item
{
    ubyte[8]    magic;
    unit        checksum;
    ubyte[20]   siganature;
    uint        file_size;
    uint        header_size;
    unit        endian_tag;
    uint        link_size;
    uint        link_off;
    uint        map_off;

    uint        string_ids_size;
    uint        string_ids_off;
    uint        type_ids_size;
    uint        type_ids_off;
    uint        proto_ids_size;
    uint        proto_ids_off;
    uint        method_ids_size;
    uint        method_ids_off;
    uint        class_defs_size;
    uint        class_defs_off;
    uint        data_size;
    uint        data_off;
}
```



dex 文件

jarsigner

Android APK的发布是必需要签名.

Jarsigner.exe 工具与证书 android.keystore 联合进行签名

目 录

CONTENTS

APK文件反编译

什么是反编译
如何防止反编译
Apk文件的基本构造

Apk加固的方案原理

Apk加固总体架构
apk签名是什么
Dex文件加密的基本
方案框架

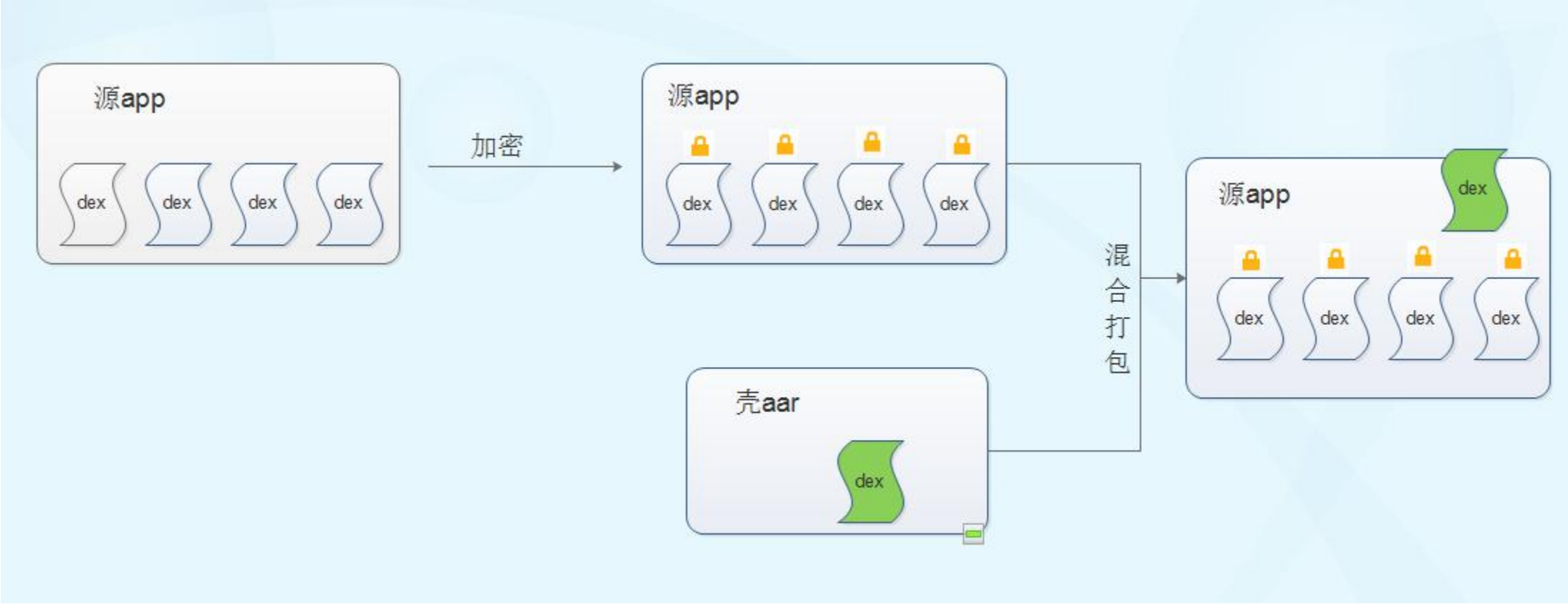
Apk加固项目实战

Dex文件基本结构
Dex文件的修改原则
Apk加固项目实战
Apk脱壳技术实战

课程总结

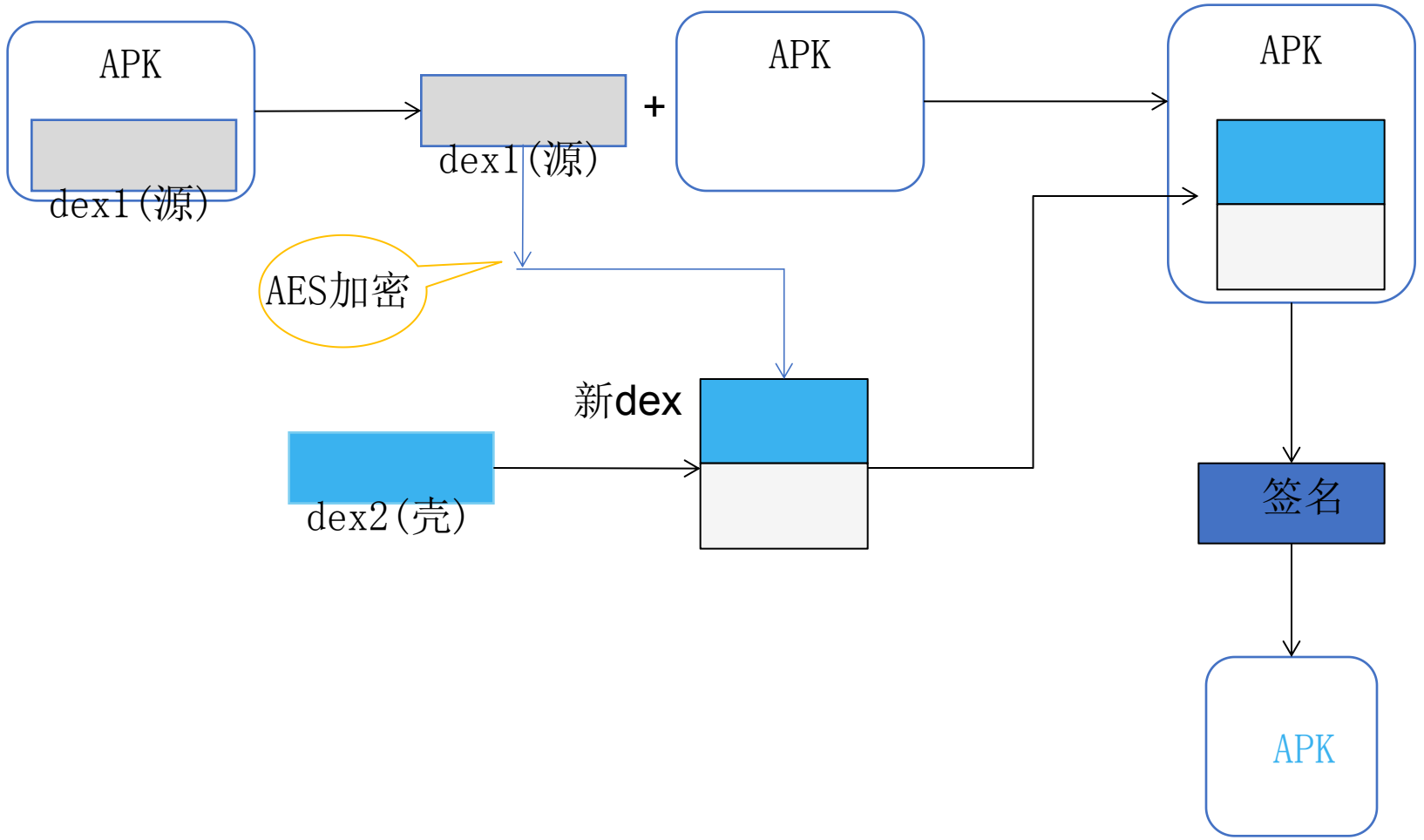
课程技术总结
交流互动

04 加密过程



这个里面都是什么操作？对Java那块知识要求最多？

03 加固总体框架



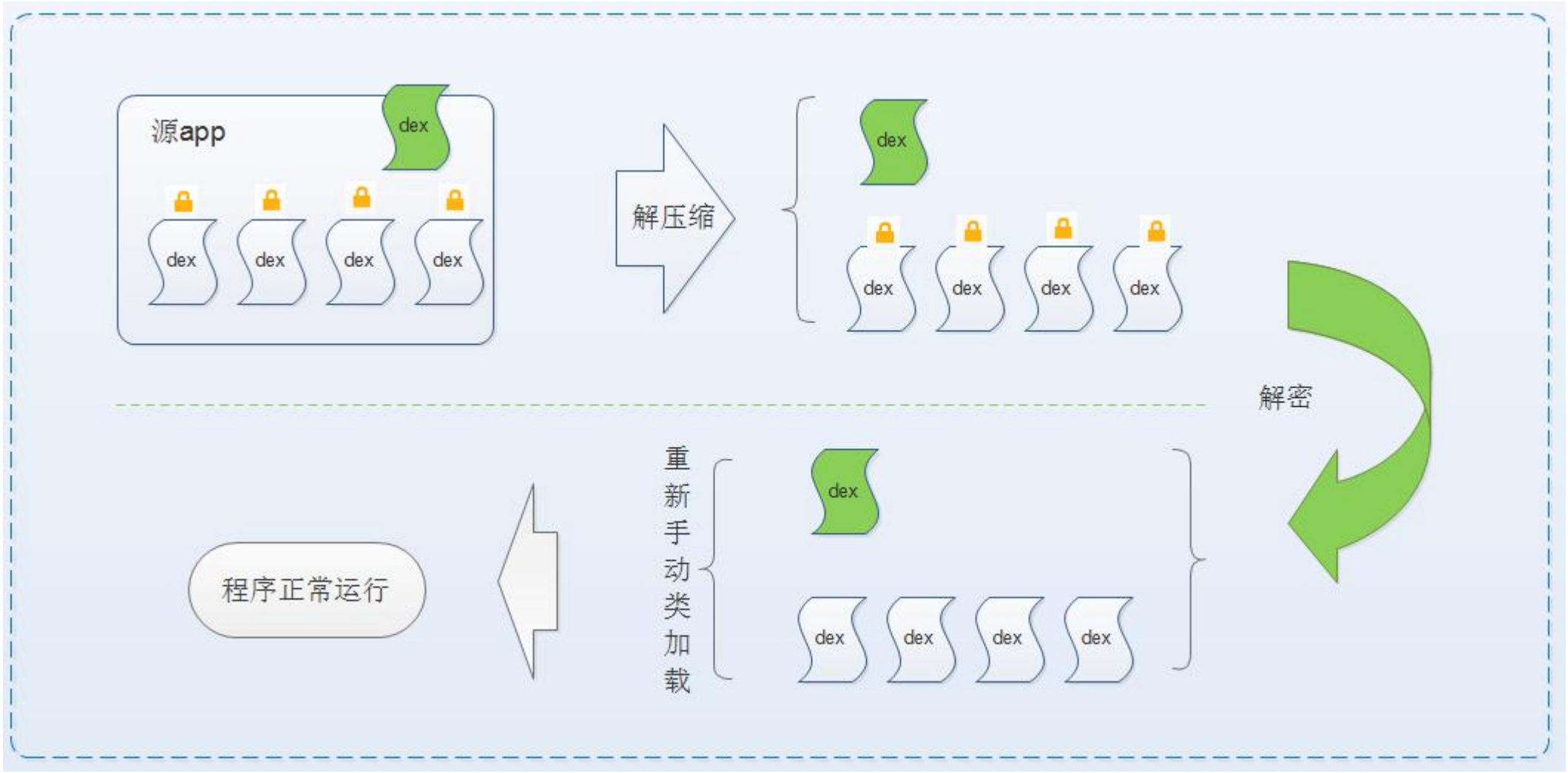
05 APK文件如何签名

jarsigner.exe	E:\software\android-studio-ide-182.5070326-windows\android-studio\jre\bin
jarsigner.exe	C:\Program Files\Android\Android Studio\jre\bin
jarsigner.exe	C:\Program Files\Java\jdk1.8.0_131\bin

debug.keystore	C:\Users\Alvin\.android

```
String cmd[] = {"cmd.exe", "/C ","jarsigner",  "-sigalg", "MD5withRSA",
        "-digestalg", "SHA1",
        "-keystore", "C:/Users/Alvin/.android/debug.keystore",
        "-storepass", "android",
        "-keypass", "android",
        "-signedjar", signedApk.getAbsolutePath(),
        unsignedApk.getAbsolutePath(),
        "androiddebugkey"};
```

05 APK文件如何运行（脱壳）



双亲委托机制
ClassLoader

某个类加载器在加载类时，首先将加载任务委托给父类加载器，依次递归，如果父类加载器可以完成类加载任务，就成功返回；只有父类加载器无法完成此加载任务或者没有父类加载器时，才自己去加载。

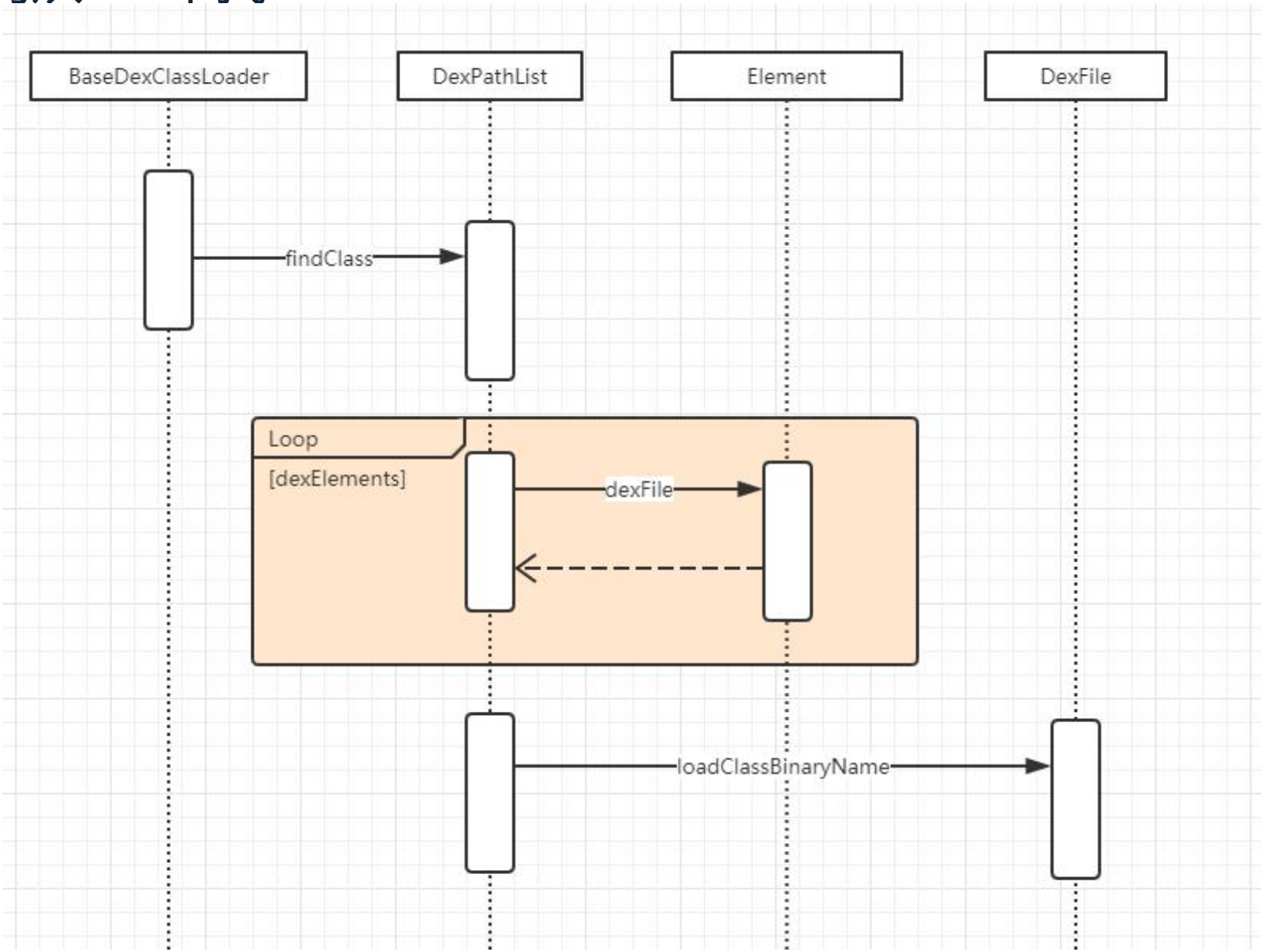
```
protected Class<?> loadClass(String name, boolean resolve) throws
ClassNotFoundException{

    // 检查class是否有被加载
    Class c = findLoadedClass(name);
    if (c == null) {
        long t0 = System.nanoTime();
        try {
            if (parent != null) {
                //如果parent不为null，则调用parent的loadClass进行加载
                c = parent.loadClass(name, false);
            } else {
                //parent为null，则调用BootClassLoader进行加载
                c = findBootstrapClassOrNull(name);
            }
        } catch (ClassNotFoundException e) {

        }

        if (c == null) {
            // 如果都找不到就自己查找
            long t1 = System.nanoTime();
            c = findClass(name);
        }
    }
    return c;
}
```

1、避免重复加载，当父加载器已经加载了该类的时候，就没有必要子ClassLoader再加载一次。

2、安全性考虑，防止核心API库被随意篡改。

在线源码阅读：
https://www.androidos.net.cn/
或
http://androidxref.com/

05 如何从dex中找class？



05 如何进行NDK 层加密

一般情况下我们没有必要使用NDK，官方也提到了使用native开发会增加开发过程的复杂性。但是对于一些计算密集型的应用，例如游戏、图像处理，使用NDK能提高运算性能。还有一些情况为了复用现有库或者跨平台库，也会选择NDK。

上面提到的一些核心算法和秘钥，大家选择放到native层，潜意识中也是默认native的破解难度比java高，还有欺负大部分Android程序员不会写C/C++代码:）

05 对称加密 &非对称加密

对称加密：加密和解密的秘钥使用的是同一个

例如：DES、3DES、Blowfish、IDEA、RC4、RC5、RC6 和 AES

非对称加密算法：公开密钥与私有密钥是一对，如果用公开密钥对数据进行加密，只有用对应的私有密钥才能解密；如果用私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。因为加密和解密使用的是两个不同的密钥，所以这种算法叫作非对称加密算法

目 录

CONTENTS

APK文件反编译

什么是反编译
如何防止反编译
Apk文件的基本构造

Apk加固的方案原理

Apk加固总体架构
apk签名是什么
Dex文件加密的基本
方案框架

Apk加固项目实战

Dex文件基本结构
Dex文件的修改原则
Apk加固项目实战
Apk脱壳技术实战

课程总结

课程技术总结
交流互动

课程总结

- 完成apk加固，我们需要具备的知识体系有哪些？

知识点	面试体系
熟练掌握Java IO相关代码	Java语言进阶
深入研究Android apk的启动流程	精通Android FrameWork层
精通Multidex文件加载机制，精通类加载机制	精通Android FrameWork层 JVM， DVM ClassLoader
明确dex文件的基本构造，了解dex文件相关源码	精通Android FrameWork层 dex
APK打包的基本流程需要理解	Gradle 工具熟练
掌握C/C++语言及NDK开发	精通C/C++ 及NDK开发
掌握Java 反射和动态代理	Java语言进阶基础