LOGIC

iOS应用签名原理

QQ : 1900009932

@Hank

LOGIC

CONTENTS

代码签名

代码签名是对可执行文件或脚本进行**数字签名**.用来确认软件在签名后未被修改或损坏的措施。和数字签名原理一样,只不过签名的数据是代码而已.

简单的代码签名

在iOS出来之前,以前的主流操作系统(Mac/Windows)软件随便从哪里下载都能运行,系统安全存在隐患,盗版软件,病毒入侵,静默安装等等.那么苹果希望解决这样的问题,要保证每一个安装到 iOS 上的 APP 都是经过苹果官方允许的,怎样保证呢?就是通过**代码签名**。

如果要实现验证.其实最简单的方式就是通过苹果官方生成非对称加密的一对公私钥.在iOS的系统中内置一个公钥,私钥由苹果后台保存,我们传APP到AppStore时,苹果后台用私钥对APP数据进行签名,iOS系统下载这个APP后,用公钥验证这个签名,若签名正确,这个APP肯定是由苹果后台认证的,并且没有被修改过,也就达到了苹果的需求:保证安装的每一个APP都是经过苹果官方允许的.

如果我们iOS设备安装APP只从App Store这一个入口这件事就简单解决了,没有任何复杂的东西,一个数字签名搞定.

但是实际上iOS安装APP还有其他渠道.比如对于我们开发者iOSER而言,我们是需要在开发APP时直接真机调试的.而且苹果还开放了企业内部分发的渠道,企业证书签名的APP也是需要顺利安装的.

苹果需要开放这些方式安装APP,这些需求就无法通过简单的代码签名来办到了。

苹果的需求

那么我们来分析一下,它有些什么需求：

- 安装包不需要上传到App Store,可以直接安装到手机上.
- 苹果为了保证系统的安全性,又必须对安装的APP有绝对的控制权
  - 经过苹果允许才可以安装
  - 不能被滥用导致非开发APP也能被安装

为了实现这些需求,iOS签名的复杂度也就开始增加了,苹果这里给出的方案是**双层签名.**

双层代码签名

为了实现苹果验证应用的一些需求,iOS签名的复杂度也就开始增加了,苹果给出的方案是**双层签名.**

双层代码签名

iOS的双层代码签名流程这里简单梳理一下,这也不是最终的iOS签名原理.iOS的最终签名在这个基础上还要稍微加点东西.

首先这里有两个角色.一个是iOS系统 还有一个就是我们的Mac系统.因为iOS的APP开发环境在Mac系统下.所以这个依赖关系成为了苹果双层签名的基础.

1. 在Mac系统中生成非对称加密算法的一对公钥\私钥(你的Xcode帮你代办了).这里称为公钥M 私钥M . M = Mac

2. 苹果自己有固定的一对公私钥,跟之前App Store原理一样,私钥在苹果后台,公钥在每个iOS系统中.这里称为公钥A , 私钥A. A=Apple

3. 把公钥M 以及一些你开发者的信息,传到苹果后台(这个就是CSR文件)，用苹果后台里的私钥 A 去签名公钥M。得到一份数据包含了公钥M 以及其签名，把这份数据称为证书。

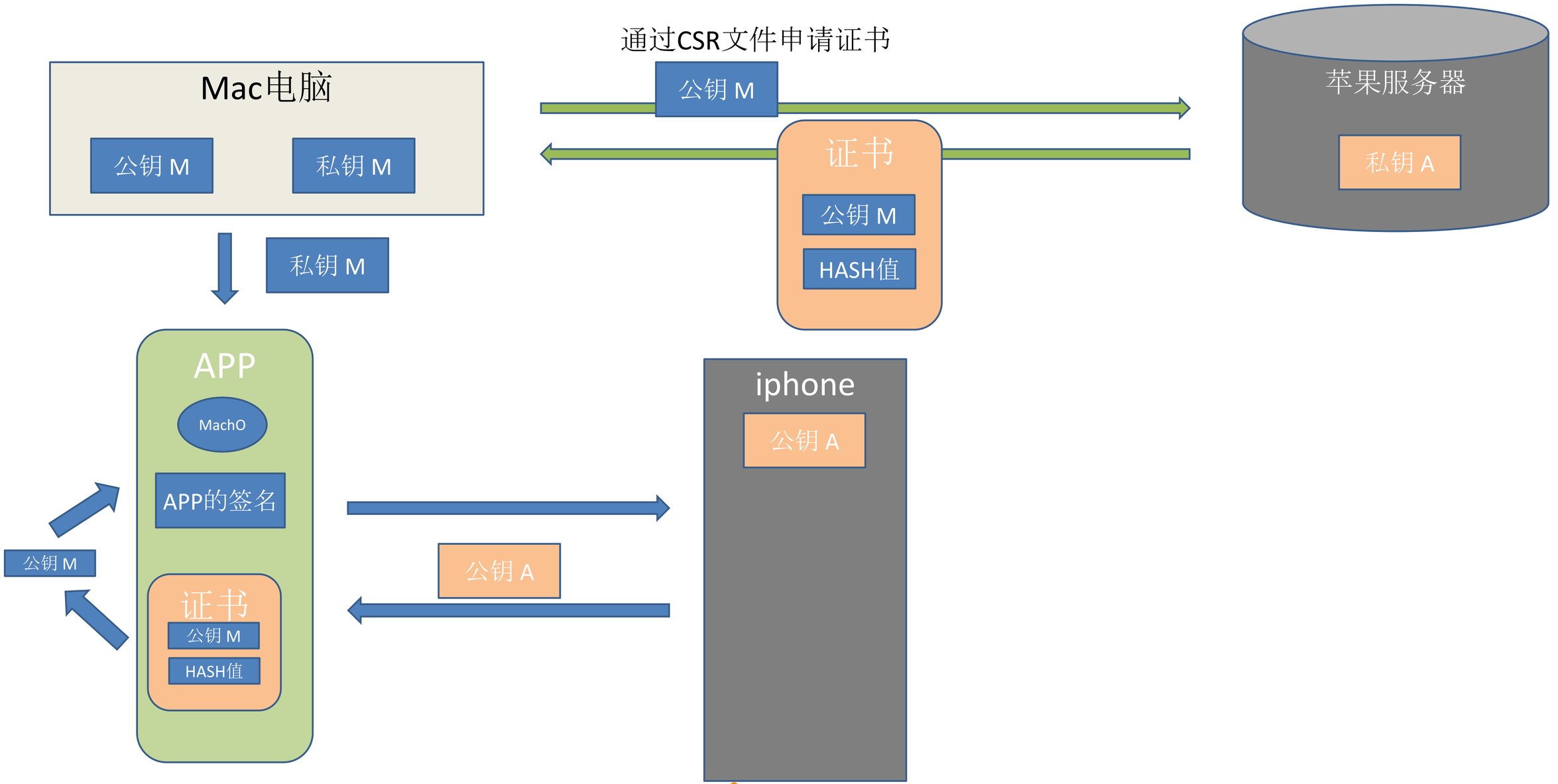
双层代码签名

4. 在开发时，编译完一个 APP 后，用本地的私钥 M(今后你导出的P12) 对这个 APP 进行签名，同时把第三步得到的证书一起打包进 APP 里，安装到手机上。

5. 在安装时，iOS 系统取得证书，通过系统内置的公钥 A，去验证证书的数字签名是否正确。

6. 验证证书后确保了钥 M 是苹果认证过的，再用公钥 M 去验证 APP 的签名，这里就间接验证了这个 APP 安装行为是否经过苹果官方允许。（这里只验证安装行为，不验证APP 是否被改动，因为开发阶段 APP 内容总是不断变化的，苹果不需要管。）

有了上面的过程，已经可以保证开发者的认证，和程序的安全性了。 但是，你要知道iOS的程序，主要渠道是要通过APP Store才能分发到用户设备的，如果只有上述的过程，那岂不是只要申请了一个证书，就可以安装到所有iOS设备了？

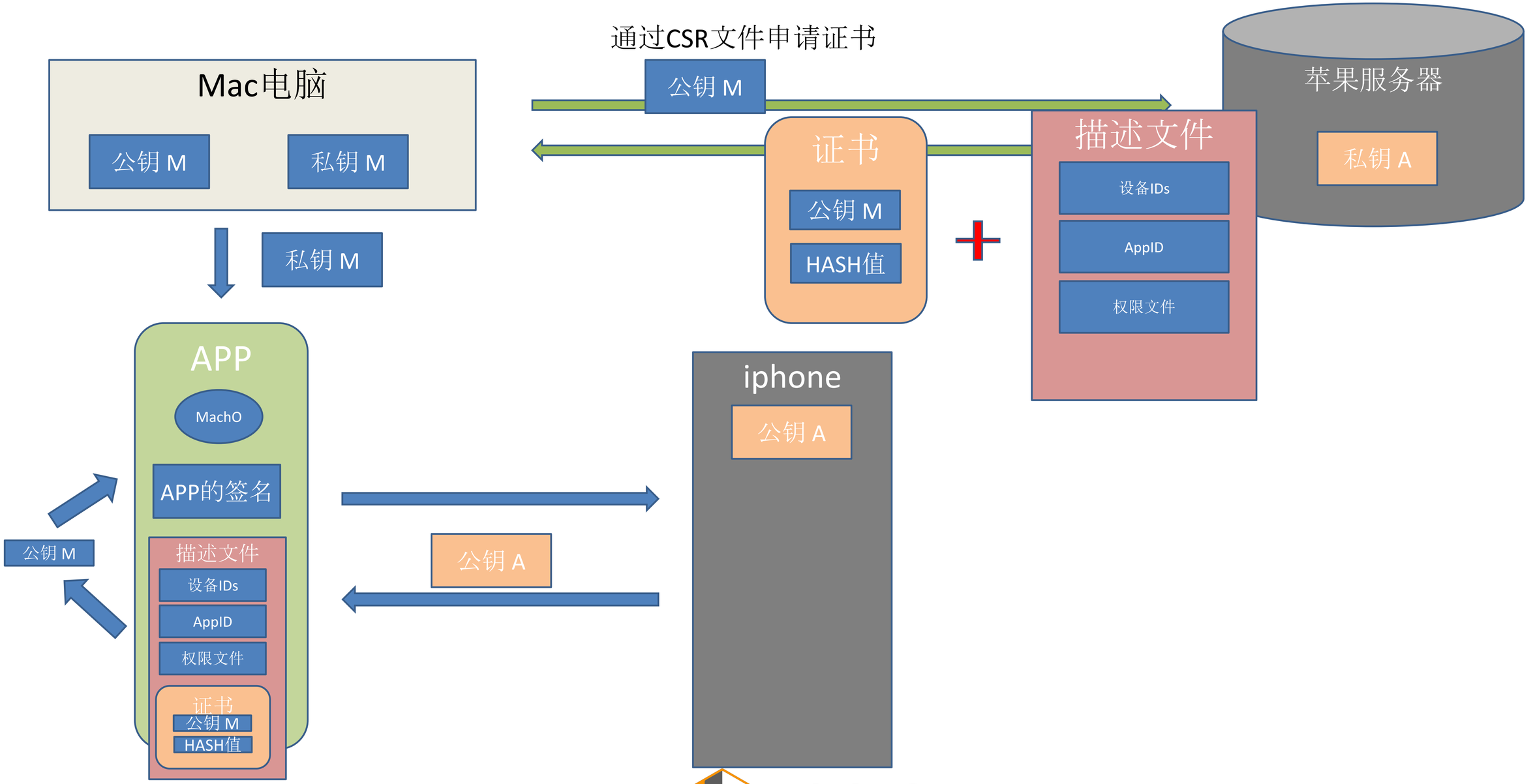

通过CSR文件申请证书
Mac电脑
公钥 M
私钥 M
公钥 M
证书
公钥 M
HASH值
苹果服务器
私钥 A
私钥 M
APP
MachO
APP的签名
公钥 M
证书
公钥 M
HASH值
公钥 A
iphone
公钥 A

描述文件

描述文件（Provisioning profile）一般包括三样东西：证书、App ID、设备。当我们在真机运行或者打包一个项目的时候，证书用来证明我们程序的安全性和合法性。

描述文件

苹果为了解决应用滥用的问题,所以苹果又加了两个限制.
第一限制在苹果后台注册过的设备才可以安装.
第二限制签名只能针对某一个具体的APP.
并且苹果还想控制App里面的iCloud/PUSH/后台运行/调试器附加这些权限,所以苹果把这些权限开关统一称为**Entitlements**(授权文件).并将这个文件放在了一个叫做**Provisioning Profile(描述文件)**文件中.
描述文件是在AppleDevelop网站创建的(在Xcode中填上AppleID它会代办创建),Xcode运行时会打包进入APP内.
所以我们使用CSR申请证书时,我们还要申请一个东西!! 就是描述文件!
在开发时,编译完一个 APP 后,用本地的私钥M对这个APP进行签名,同时把从苹果服务器得到的 Provisioning Profile 文件打包进APP里,文件名为embedded.mobileprovision,把 APP 安装到手机上.最后系统进行验证。



LOGIC

THANK YOU

免费领取课后视频　QQ:1900009932