基于沙盒技术的企业移动应用安全平台①

马思峻, 肖　荣, 成江伟

(上海理想信息产业(集团)有限公司, 上海 201315)

摘　要: 随着智能手机的普及, 移动互联网正在成为发展主流, 大量的应用从 PC 向手机端转移. 移动应用给企业的业务和办公带来了便利, 同时也带来了各种安全风险和管理问题. 本文研究了企业移动应用沙盒技术, 它将企业移动应用及其数据和个人应用及其数据进行隔离, 保护企业应用和数据的安全. 经过实践证明, 基于沙盒技术开发出的产品能够满足企业的数据保护要求, 并且使用方便.

关键词: 沙盒技术; 移动互联网; 数据安全; 应用重打包; PBKDF2 加密算法

Enterprise Mobile Application Security Platform Based on Sandbox Technology

MA Si-Jun, XIAO Rong, CHENG Jiang-Wei

(Shanghai Ideal Information Industry (Group) Co., Ltd., Shanghai 201315, China)

Abstract: With the emergence of high-performance smart phones, mobile Internet is becoming mainstream, a large number of applications transfers from PC to mobile terminal, Convenient office also brings a variety of security risks and management issues. This paper studies the enterprise mobile application sandbox Technology, it realizes enterprise mobile application and data and personal applications and data isolation, in order to protect the security of enterprise applications and data. It has been proved through practice, based on the sandbox technology to develop products to meet the requirements of enterprise data protection, and it is convenient to use.

Key words: Sandbox technology; mobile internet; data security; app wrapper; PBKDF2

移动互联网的飞速发展, 改变了企业传统的业务模式, 提高了工作效率. 但同时也给企业的数据安全带来了巨大的挑战, 我们面对各种攻击的可能性会大大增加, 面临潜在的风险:

① 移动设备中的不安全应用程序会危及到企业网络的安全.

② 移动设备可能在不安全的网络(如公共 WiFi 热点)中使用, 容易造成恶意软件感染和数据泄漏.

③ “越狱”或“获取 Root 权限”移动设备会带来更多的风险.

④ 移动设备被盗, 在未经授权的情况下访问公司的网络或者泄漏敏感数据.

所以, 我们需要使用企业移动应用安全平台来保证企业数据的安全, 本文介绍了企业移动应用安全平台架构及其关键技术点本地存储、通信数据加密算法的实现.

1　企业移动应用安全平台架构

企业移动应用安全平台检测企业移动应用运行环境是否满足要求, 远程安装升级删除企业移动应用, 阻止外部应用打开容器内应用和服务并读取其中的数据, 如果手机丢失, 企业管理员可以在服务平台锁定应用和删除企业应用数据. 安全容器中的企业移动应用运行时产生的数据加密保存在手机本地, 外部应用不能打开修改, 除此之外, 安全容器还实现了企业应用和服务器之间通信数据的加密, 保证通信数据不被窃取, 支持常用的数据通信方式 Socket 和 Https.

企业移动应用安全平台由四部分功能组成, 安装在手机上的安全容器、安全加固 SDK, Web 管理门户和通信代理, 各部分的功能及其相互间的关系如下所示.

① 收稿时间:2015-12-30;收到修改稿时间:2016-02-02　[doi:10.15888/j.cnki.csa.005309]

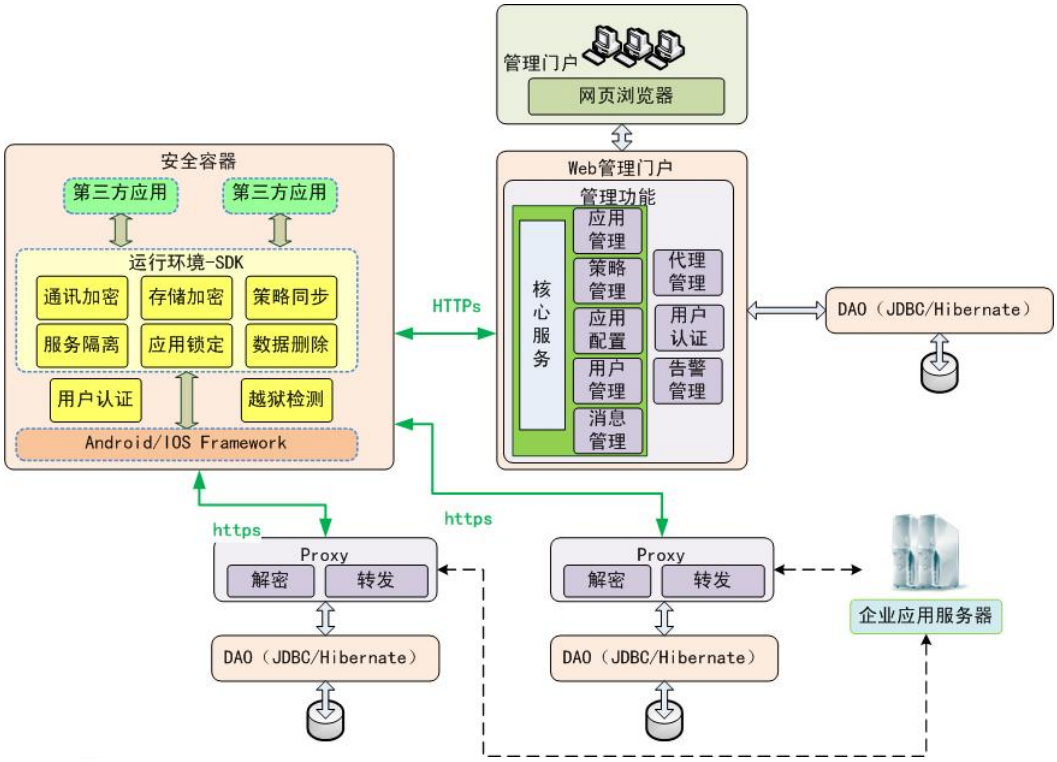

图 1 功能架构图

1.1 安全容器

安全容器在手机上作为企业应用启动入口, 实现用户认证、通信密钥加解密、数据存储密钥加解密、服务隔离、密码策略校验、超时锁容器、管理平台各种下发指令执行等功能. 安全容器中包含企业应用商店, 可以下载安装升级企业发布的各种应用, 安装完成的企业应用只能在安全容器内才能运行启动, 不能在外部启动, 如果通过其他方式强制从外部应用启动安全容器内应用, 应用启动时会报没有授权, 禁止运行.

用户每次输入密码通过认证后, 管理平台会返回数据通信加解密的key, 数据传输采用https, 而且每次认证通过后加解密 key 都会发生变化, 进入安全容器应用后数据交互使用key进行AES256位对称加密. 加解密key分享给使用安全加固SDK应用的数据通信使用.

安全容器激活后输入的用户密码和保存在本地的salt用来解密保存在本地的应用数据加解密key, 加解密key获得后分享给使用安全加固SDK应用的本地存储加解密使用.

Android 平台数据和通信加解密 key 分享通过AIDL[1]方法实现进程间通信, 在安全容器中先建立一个 interface 定义外部进程调用的接口, 然后在 Service 中具体实现接口返回加解密 key. IOS 平台加解密 key 分享通过 IPC[2]方法实现进程间通信, 也就是在已加固应用的 info.plist 中添加 URL Schema 键值对, 安全容器通过传递参数 URL 打开已加固 IOS 应用.

安全容器实时响应管理平台下发的各种指令和策略, 加解锁容器, 加解锁应用, 擦除容器内应用数据, 越狱或 root 后的相应操作, 密码策略更新.

1.2 安全加固 SDK

安全加固 SDK 必须注入到每个企业移动应用中才能实现文件读写加解密、Http、Socket 的加密传输. 本地文件存储除了文件操作和数据库存储之外, Android 和 IOS 平台还有各自特有的本地数据存储方式需要实现加密保存数据, 例如 Android 平台的SharedPreferences, IOS平台的NSUserDefaults. SDK在Android平台上的表现形式是jar包和so文件, 在IOS平台上是framework. Android SDK中包含连接安全容器中Service的接口定义, Service连接成功后就可以取到安全容器的加解密key.

Android 应用替换手机原有 SDK 的 file、http、

Socket 类有两种方式, 如果有加固应用的源代码, 可以把 SDK 加入开发环境中, 然后修改程序代码把原来程序中使用的文件操作、http、Socket 类替换为安全加固 SDK 中对应的类, 如果没有加固应用的源代码, 只能采取 App Wrapper[3]的方式处理加固应用的安装包, 首先把安装包脱壳解开为 smali[4]文件, 其次处理 smali 文件, 再次重新打包生成加固后的安装包, 最后用企业安全证书签名发布到企业应用商店. 安全容器和重新打包生成安装包的签名证书必须一致.

IOS 应用在项目中加入 SDK, 修改代码后用企业证书打包签名生成 IPA 安装包发布到企业应用商店.

1.3 管理平台安全

管理平台保存着企业用户数据和应用安装包, 自身的安全尤为重要, 需要采取措施确保平台的安全, 应用开发时就已经增加了密码不能存在弱口令、登录用户身份敏感信息进行加密传输、登录时采用图片验证码策略、身份认证后会话不能被绕过和失效、长时间不用有超时机制、为不同权限用户设置不同账户级别、对用户行为进行日志记录等安全措施.

应用部署时也要采取安全措施, 首先管理平台和通信代理服务器放置在企业统一防火墙后, 内网和外网之间建立一个安全网关, 只有客户端指定端口的数据才能访问内网, 企业内部有网络监控服务, 持续监控应用服务器和数据库服务器的运行状态, 在发现异常情况时会及时报警提醒管理员采取措施, 其次使用通信代理服务器后, 在应用服务器侧设置 IP 白名单, 只有代理服务器才能访问应用服务器, 而不是手机直接访问应用服务器, 大大提高了企业应用的安全性, 最后 Web 管理门户只能从内网访问, 外网不能直接访问, 保证了平台安全, 网络架构图如下所示.

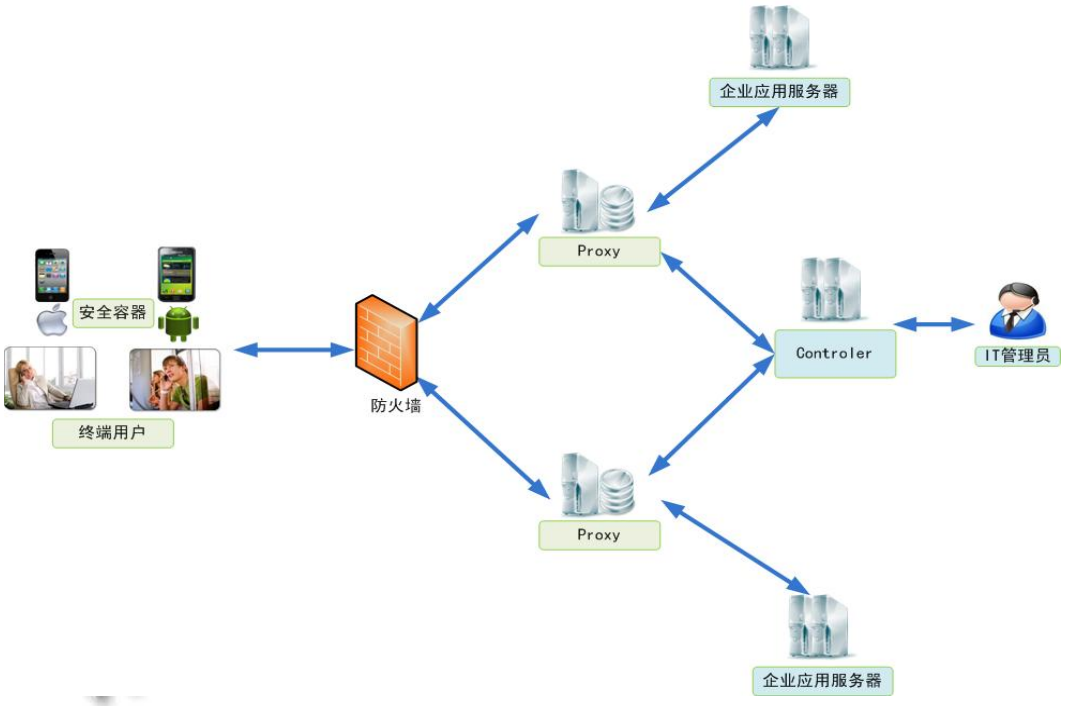

图 2 网络架构图

1.4 Web 管理门户功能

Web 管理门户实现企业员工及其部门维护、企业应用管理、沙箱安全策略配置、对员工手机上的安全容器下发指令(例如擦除加固应用的数据、锁定容器、解锁容器)、企业应用通信代理服务器配置以及统计报表、日志记录等功能.

① 员工及其部门维护

维护多级部门

维护员工信息包括用户名、手机号、邮箱等, 用于登录企业移动应用安全容器

为员工生成容器激活码.

② 企业应用管理

应用 app wrapper(原有应用解包嵌入安全加固 SDK 后重新签名打包)

应用上传、应用升级

应用下载统计

应用授权给员工或组织.

③ 策略管理

密码强度策略、安全性检查策略、数据泄漏防护策略设置.

④ 设备管理

查看设备型号、是否 ROOT、系统版本、存储空间、已安装应用等信息

锁定应用、解锁应用、删除容器内所有应用数据.

1.5 通信加密

通信代理作为一个移动应用数据通信加解密的中继, 放置在企业防火墙后面, 来自安全容器中企业应用的数据在这里进行解密转发到应用服务器, 应用服务器返回的数据通过通信代理加密后返回给企业移动应用.

2 加密算法

加密算法是移动沙箱技术研究的重点, 在这里做详细说明, 主要有通信加密、本地存储加密两部分, 密钥 key 必须加密保存在手机本地或管理平台, 不容易被破解或获取, 而且要求定期改变.

为了保证加密算法安全不容易被反编译破解, Android 平台的所有加密算法都不使用 java 语言实现, Android 用 C 编写 so 文件封装加解密函数供 java 代码调用,IOS 用 C 语言编写加解密函数.

2.1 本地存储加密

激活码验证正确, 员工根据密码策略输入用户密码, 然后执行如下所示的流程生成本地数据加密 key 并加密保存在本地.

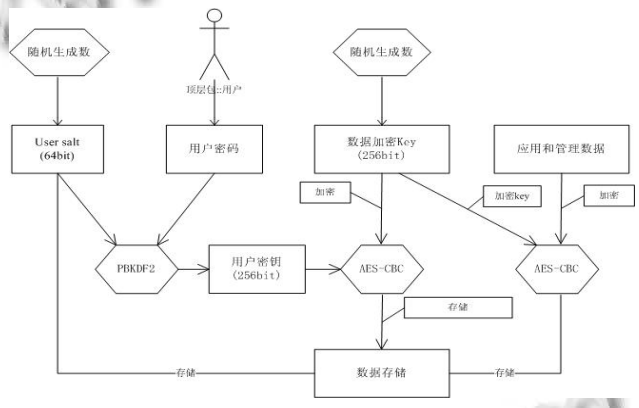

图 3 本地数据加密流程

本地存储数据加密流程使用了 PBKDF2[5]算法, PBKDF2 算法简单而言就是将 salted hash 进行多次重复计算, 重复次数是可选择的. 美国政府机构已经将这个方法标准化, 并且用于一些政府和军方的系统. PBKDF2 算法最大的优点是标准化, 实现容易并采用了久经考验的 SHA 算法. 本项目中上图所示的 256 位用户密钥就是使用保存在本地的 Salt 和员工自己设置的用户密码通过 PBKDF2 算法计算得到.

随机生成一个 256 位数据加密 key 用来加密安全容器中安装应用运行时生成的保存在手机本地的数据, 该加密 key 需要 AES256 加密后保存在本地, 对该 key 进行加密的 256 位密钥来自于用户密码和随机生成的 64 位 Salt. 随机生成的 64 位 Salt 和用户密码经过 PBKDF2 算法生成 256 位用户密钥用于加解密真正的加解密应用数据的随机 256 位 key.

这种设计保证了平台管理员也不知道加解密本地数据的 key. 用户密码保存在平台提供用户进入安全容器的验证, 而 Salt 保存在手机本地, 只有用户密码和本地存储的 salt 同时获得才能生成解密本地应用数据加解密的 key. 该 key 分享给使用安全加固 SDK 的企业移动应用的本地数据存储加密使用.

用户密码如果改变了, 不需要把所有已经加密过的应用数据解密后再次用新密码加密, 而只需要把保存在本地的数据加解密 key 用原有用户密码和随机数 salt 解开后再次用新用户密码和随机数 salt 加密后保存在本地.

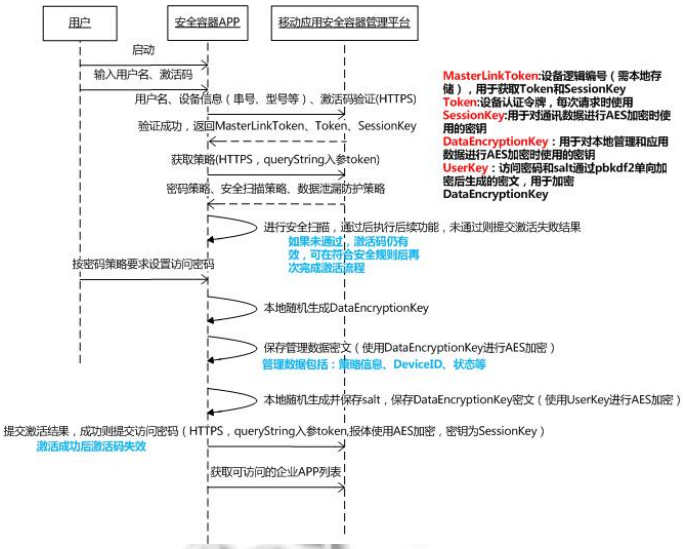

图 4　通信数据加密流程

2.2 通信数据加密

员工输入邮箱地址和激活码后, 通过 HTTPS 发送消息到移动应用沙箱管理平台进行验证, 验证消息中包含邮箱地址、激活码、设备的唯一标识信息(IMEI,IMSI),平台认证通过后返回 MasterLinkToken, Token, SessionKey. MasterLinkToken 作为用户认证成功的唯一标识, 用于再次进入安全容器 APP 向服务器获取 Token、SessionKey, Token、SessionKey 在每次进入容器后都会变化, 而且还会过期, 需要向服务器重新请求获得. SessionKey、Token 分享给安全加固 SDK 作为数据通信 AES256 加解密的 key 和身份认证 Token.

APP 再次向平台发送请求获取员工密码策略、安全扫描策略、放置数据泄漏策略, 服务器用 SessionKey AES256 位加密后返回 APP, APP 用上一步获取的 SessionKey 解密后执行策略. 如果员工手机运行环境存在问题, 提示员工只有满足安全策略要求才能继续激活. 手机环境满足安全策略要求后, 员工按照密码策略要求设置用户密码, 如果用户密码不符合密码策略设置要求, 提示员工重新设置.

员工设置符合密码策略要求的用户密码后, APP 随机生成本地应用数据加解密的 DataEncryptionKey, DataEncryptionKey 在手机本地的加密存储如图 3 所示, MasterLinkToken 和服务器下发的各种策略使用 DataEncryptionKey AES256 位加密后存储在手机本地, DataEncryptionKey、MasterLinkToken 和服务器下发的各种策略在手机本地保存成功后, 用户设置的密码使用 SessionKey AES256 位加密后发送给管理平台.

3　结语

通过以上研究, 明确了企业移动应用安全平台架构和技术难点, 解决了本地数据存储安全和数据通信安全问题. 安全平台除了通信和文件的加解密外, 还包括管理平台 Push 指令的加密推送、容器用户界面通过消息传递处理的方式进行切换更新等内容, 在此不再做详细论述. 平台将来还会增加加密联系人、加密短信、加密邮件、打开加密 pdf 和 office 格式文档等功能. 该产品的定位是企业移动应用市场, 为企业移动应用提供一个安全、统一的移动工作平台, 实现单点登录, 给 BYOD[6]模式提供后端管控能力.

参考文献

1 DiMarzio JF. Android. Tata McGraw-Hill Education, 2008.
2 Kivity A. Inter-process communication. Distributed Systems, 2011, 29(11): 975–975.
3 Blaisdell J, Vally JM. Securing and managing apps on a device: U.S. Patent 8,549,656. [2013-10-1].
4 Hoffmann J, Ussath M, Holz T, et al. Slicing droids: Program slicing for smali code. Proc. of the 28th Annual ACM Symposium on Applied Computing. ACM. 2013. 1844–1851.
5 于飞,李晓华,兰天,等.PBKDF2 函数的一种快速实现.信息安全与通信保密,2014,12(12):100–102.
6 钱煜明,董振江,吕达,王蔚.BYOD 企业移动设备管理技术.中兴通讯技术,2013,19(6):33–38.