保护智能生活

**梆梆安全**

**应用OTA安全解决方案**

**贵阳银行**

北京洋浦伟业科技发展有限公司

（梆梆安全）

**目录**

[1 方案背景 4](#_Toc474510449)

[2 方案目标 5](#_Toc474510450)

[3 方案设计 6](#_Toc474510451)

[3.1 功能简介 6](#_Toc474510452)

[3.2 详细方案设计 6](#_Toc474510453)

[3.2.1 Android OTA热更新 6](#_Toc474510454)

[3.2.2 IOS OTA热更新 8](#_Toc474510455)

[3.3 方案优势 12](#_Toc474510456)

[3.4 部分产品截图 13](#_Toc474510457)

[4 梆梆安全简介 15](#_Toc474510458)

1. **分发控制**

|  |  |  |
| --- | --- | --- |
| 读者 | 文档权限 | 说明 |
| 公司内部员工 | 可修改 | 仅限项目相关人员 |
| 客户（贵阳银行） | 只读 | 仅限贵阳银行相关人员 |

1. **更新日志**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文档版本 | 变更描述 | 日期 | 作者 | 审核 |
| V1.0 | 正式版本 | 2017.2.10 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **文件版本信息说明**

文件版本信息记录本文件提交时的当前有效的版本控制信息，当前版本文件有效期将在新版本文档生效时自动结束。文件版本小于1.0时，表示该版本文件为草案，仅可作为参照资料之目的。

# 方案背景

随着智能手机以及智能设备的飞速发展，iOS、Android应用体积的不断增大，以及应用版本发布的不断更迭，如何实现用户应用版本的静默、快速升级的急切性已经显露。在金融行业，几乎每款移动金融APP都需要定期更新，用于兼容性Bug修复、安全漏洞修复、新功能发布等。因此急需一套轻便简洁稳定安全的产品来解决Android、iOS应用静默、快速升级的相关问题。

在Android开发领域，不仅系统版本众多，机型众多，而且各个市场都各有各的政策和审核速度，每次发布一个版本对于开发者来讲都是一种漫长的煎熬。用户难以第一时间获取到最新版本，同时大部分用户在面对更新，由于老版本使用习惯的养成，倾向于不更新或延迟更新，因此部分金融机构会选择强制升级。根据业界的经验数据表明，对于一款手机银行应用来说，一个强制升级的版本会导致近5%-10%的用户流失，部分用户黏性低的应用用户流失率高达近20%。如果不强制升级，大多数用户又会选择不升级，这会导致安全漏洞无法及时修复，运营强度成指数级上升，版本兼容越来越痛苦，同时新的版本的业务功能又不能得以体现并第一时间进行推广，因此android应用的差量、静默更新已经成为很多金融机构关注的重点。

在IOS开发领域，由于Apple严格的审核标准和低效率，IOS应用的发版速度极慢，稍微大型的app发版基本上都在一个月以上，严重阻碍了版本迭代速度，也导致处理 bug 和投诉等也越来越麻烦。在金融行业内经常出现bug和安全漏洞修复后的版本无法及时传递到用户手中，导致用户体验下降、用户资金受损并最终引发用户的流失，所以应用热更新对于IOS应用来说就显得尤其重要。

# 方案目标

通过应用OTA升级安全解决方案（热更新），使用户不需要去应用市场上面重新下载安装包，或者重新安装升级包，通过后台的管理控制平台实现APP静默、快速升级，达到以下目标：

1. 实现应用的差量更新，通过高效的差分技术减小更新包的体积，降低用户流量消耗；
2. 实现应用静默更新，无需用户重新下载安装包及主动点击安装更新，提高用户使用体验；
3. 多种灰度更新策略。通过管理平台添加的升级包，针对区域、系统版本、自定义检签等特性进行可控更新。
4. 完美兼容应用安全加固，不影响现有的安全加固功能
5. 提供较高的安全性保证，对升级包进行高强度加密，确保升级不被篡改。

# 方案设计

## 功能简介

梆梆安全应用OTA系统云平台提供了升级包后台托管，版本管理，保证传输安全等功能，开发者自身无需专门搭建一套更新平台，无需关心部署操作，只需在客户端集成应用OTA的SDK（详细说明见集成文档），调用初始化及检测更新方法。并通过云平台进行统一的管理和新版本的分发升级。

梆梆安全应用OTA系统也可提供本地部署方式。

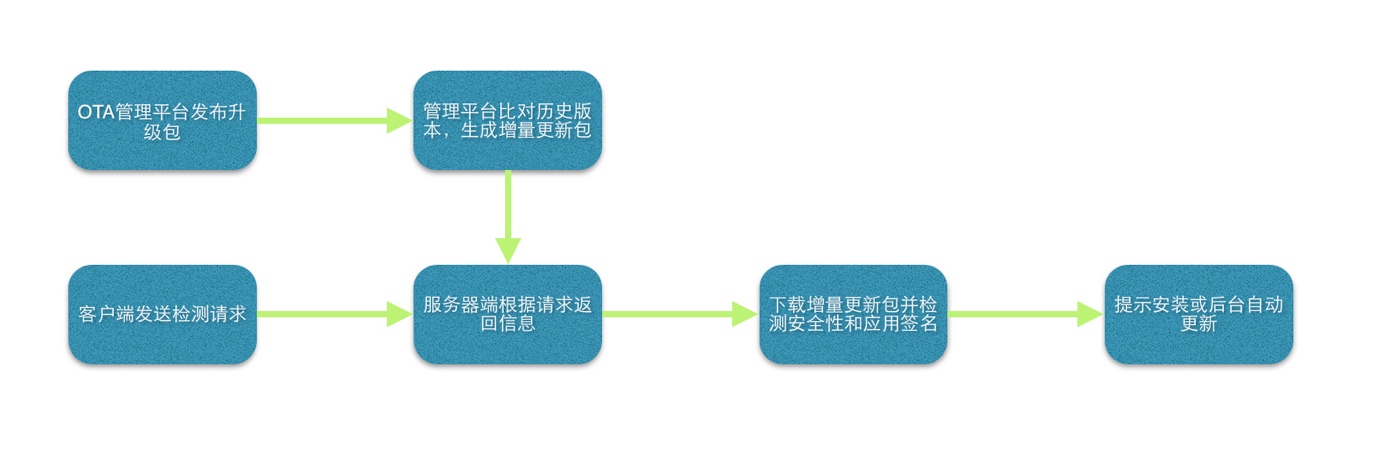
## 详细方案设计

### Android OTA热更新

#### 实现流程

Android自动更新SDK首先将用户终端上的应用版本信息（version name和version code）与应用OTA服务器上开发者上传的应用版本信息进行比较。如果服务器的应用版本更高，获取更新版本之间的差异包，自动更新SDK开始在Android后台下载差异包，并比对文件MD5，下载完成并验证成功后，则会提示用户有新的应用版本。用户选择更新之后，然后安装此应用。

**Android自动更新流程图**



|  |  |  |
| --- | --- | --- |
| 名词 | 解释 | 备注 |
| 比对历史版本 | 对管理平台同一个APP仓库内已经上传的所有apk文件进行比对 | 每一个apk版本不同，生成的增量更新包也不同 |
| 校测安全性 | 对增量更新包做MD5校验 |  |

#### 技术原理

Android在运行时加载对应的类是通过ClassLoader来实现的，ClassLoader本身是一个抽象来，Android中使用PathClassLoader类作为Android的默认的类加载器,PathClassLoader其实实现的就是简单的从文件系统中加载类文件。PathClassLoade本身继承自BaseDexClassLoader，BaseDexClassLoader重写了findClass方法，该方法是ClassLoader的核心。

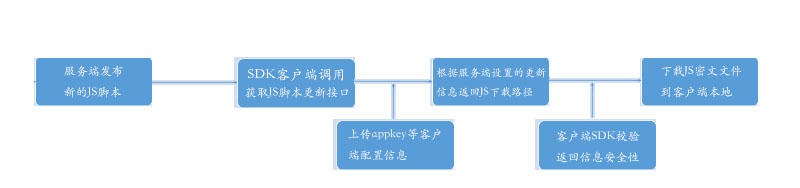
通过分析Android中的类的加载的流程，可以看出来DexPathList对象中的dexElements列表是类加载的一个核心，一个类如果能被成功加载，那么它的dex一定会出现在dexElements所对应的dex文件中，并且dexElements中出现的顺序也很重要，在dexElements前面出现的dex会被优先加载，一旦Class被加载成功，就会立即返回。

为了实现热更新，需要应用在运行时去更改PathClassLoader.pathList.dexElements，由于这些属性都是private的，因此需要通过反射来修改。另外，构造单独的dex文件所对应的dexElements数组的时候，也可以采取一个比较取巧的方式，就是通过构造一个DexClassLoader对象来加载相应的dex文件，并且调用一次dexClassLoader.loadClass(dummyClassName)方法，这样，dexClassLoader.pathList.dexElements中，就会包含相应的dex，通过把dexClassLoader.pathList.dexElements插入到系统默认的classLoader.pathList.dexElements列表前面，就可以让系统优先加载更新后的dex中的类，从而可以实现热更新了。

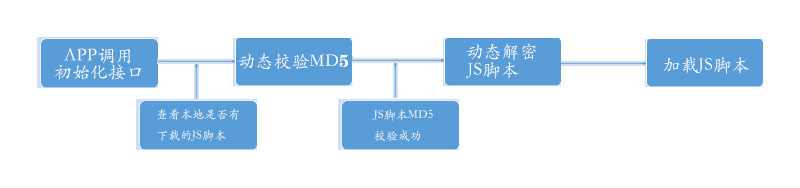
### IOS OTA热更新

#### 实现流程

应用OTA管理平台发布iOS JS脚本升级包流程图如下:



应用OTA iOS加载JS脚本升级包调用流程图如下:



|  |  |  |
| --- | --- | --- |
| 名词 | 解释 | 备注 |
| 动态校验MD5 | 每次加载js脚本的时，根据脚本内容计算这个文件的MD5，与后台返回的MD5是否一致 | 后台返回的MD5是以RSA私钥加密后的密文，SDK以公钥解密后再进行的比对 |
| 客户端SDK校验 | 对服务端请求返回的内容解密，对解密后明文做完整性MD5验证 | 防止传输内容中途被篡改 |

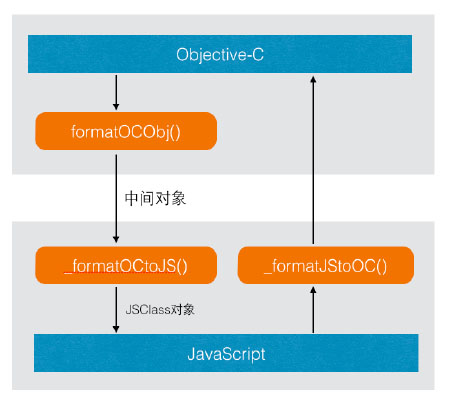
#### 技术原理

iOS自动更新SDK是使用iOS内置的JavaScriptCore.framework作为JS引擎，但没有用它JSExport的特性进行JS-OC函数互调，而是通过Objective-CRuntime，从JS传递要调用的类名函数名到Objective-C，再使用NSInvocation动态调用对应的OC方法；JS脚本补丁内容在后台服务器进行了AES白盒加密，热更新SDK每次下载的内容都加密后的文件，加载JS脚本时，每次会验证JS脚本补丁MD5值（MD5是RSA加密后密文，每次比较都是动态解密）和JS脚本内容解密操作。主要组成模块有：

* JS、OC的对象和数据转换
* 方法替换

##### JS、OC的对象和数据转换

OC对象可能会存在于 NSDictionary / NSArray 等容器里，所以需要遍历容器挑出OC对象进行格式化，OC需要把对象都替换成JS认得的格式，JS要把对象转成 JSClass 实例，JS实例回传给OC时需要把实例转为OC对象指针。所以OC流出数据时都会经过 formatOCObj() 方法处理，JS从OC得到数据时都会经过 \_formatOCToJS() 处理，JS传参数给OC时会经过 \_formatJSToOC() 处理，图示：



##### 方法替换

iOS自动更新SDK可以用 defineClass 接口任意替换一个类的方法。在OC上，每个类都是这样一个结构体：

struct objc\_class {

struct objc\_class \* isa;

const char \*name;

...

struct objc\_method\_list \*\*methodLists; /\*方法链表\*/

};

其中 methodList 方法链表里存储的是Method类型：

typedef struct objc\_method \*Method;

typedef struct objc\_ method {

SEL method\_name;

char \*method\_types;

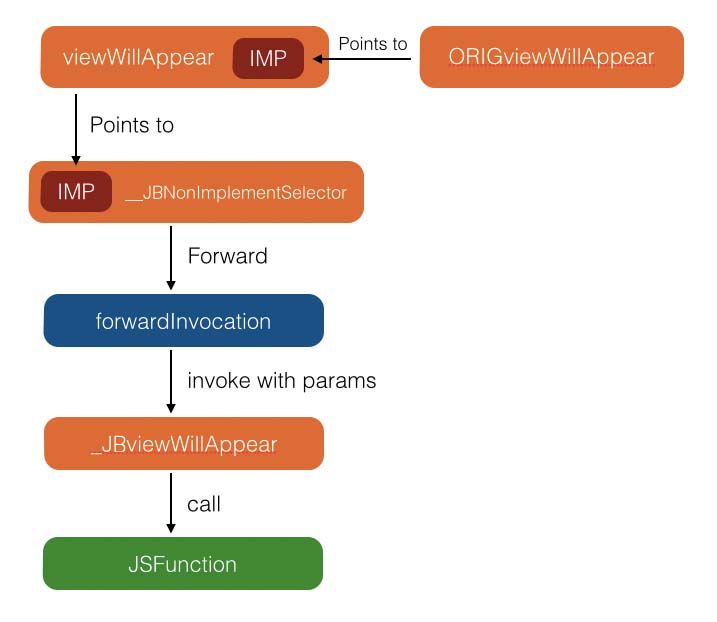
IMP method\_imp;

};

Method 保存了一个方法的全部信息，包括SEL方法名，type各参数和返回值类型，IMP该方法具体实现的函数指针。

通过 Selector 调用方法时，会从 methodList 链表里找到对应Method进行调用，这个 methodList 上的的元素是可以动态替换的，可以把某个 Selector 对应的函数指针IMP替换成新的，也可以拿到已有的某个 Selector 对应的函数指针IMP，让另一个 Selector 跟它对应。以替换 UIViewController 的 -viewWillAppear: 方法为例：

1. 把UIViewController的 -viewWillAppear: 方法通过 class\_replaceMethod() 接口指向一个不存在的IMP: class\_getMethodImplementation(cls, @selector(\_\_JBNONImplementSelector))，这样调用这个方法时就会走到 -forwardInvocation:。
2. 为 UIViewController 添加 -ORIGviewWillAppear: 和 -\_JBviewWillAppear: 两个方法，前者指向原来的IMP实现，后者是新的实现，稍后会在这个实现里回调JS函数。
3. 改写 UIViewController 的 -forwardInvocation: 方法为自定义实现。一旦OC里调用 UIViewController 的 -viewWillAppear: 方法，经过上面的处理会把这个调用转发到 -forwardInvocation: ，这时已经组装好了一个 NSInvocation，包含了这个调用的参数。在这里把参数从 NSInvocation 反解出来，带着参数调用上述新增加的方法 -JBviewWillAppear: ，在这个新方法里取到参数传给JS，调用JS的实现函数。整个调用过程就结束了，整个过程图示如下：



## 方案优势

梆梆应用OTA系统具有如下优点：

* 双平台支持（Android/iOS）

目前梆梆安全应用OTA安全解决方案可以提供对Android和iOS两大系统应用的OTA热更新，同时由于统一的后端管理控制平台进行更新控制，方便客户进行管理、运维。

* 灵活的灰度更新策略

目前梆梆安全应用OTA管理平台已经实现了针对用户区域、系统版本、自定义标签等特性的灰度更新控制。

用户区域：通过GPS信息、IP路由和基站信息实现当前当前用户位置确认，用户可在后端自定义里进行设置（如：city=北京）。

系统版本：前端SDK会自动采集用户手机的操作系统娄据，Android目前已经适配了最新的Android 7.1，iOS目前已经适配了最新的10.2。

自定义标签：客户根据自身业务发展需求，对用户进行自定义标签，如钻石用户、白金用户、普通用户。

* 完美兼容性

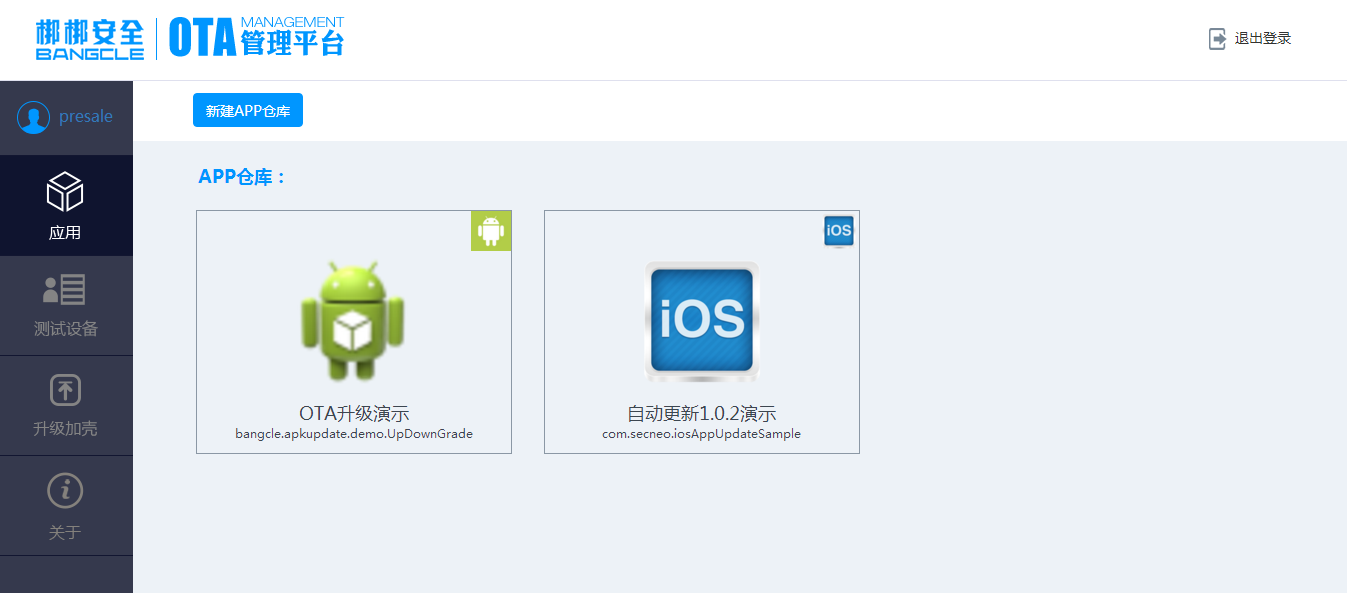
目前梆梆安全应用OTA已经成功兼容梆格应用安全加固，且不影响现有的安全加固功能。

* 高安全性

所有的升级包后台服务器进行了AES白盒加密，保证传输过程中不被窃取和篡改，同时在升级包客户端也是加密存储。

## 部分产品截图







# 梆梆安全简介

|  |  |  |
| --- | --- | --- |
| **类别** | **梆梆安全** | |
| **公司简介** | 梆梆安全成立于2005 年，注册资本1000万，是全球领先的移动应用安全服务提供商，目前拥有300余名员工，其中70%为研发和技术人员，在全国设有11个分公司和办事处。2016年获得IDG领投4轮近1.3亿美金的风险投资。 | |
| **从业年数** | 2005年成立至今拥有11年从业经验。 | |
| **科技实力** | 拥有国家级专利证书8项，软件著作权51项。 | |
| **公司资质** | 1. 国家高新技术企业 2. 中关村高新技术企业 3. 质量管理体系认证证书ISO9001 4. 信息技术服务管理体系认证证书ISO20000 5. 信息安全管理体系认证证书ISO27001 6. 软件产品登记证书 7. 软件企业认定证书 8. 企业资信等级（AAA）证书 9. CMMI3级资质证书 | |
| **行业准入资质** | 1. 国密局商用密码产品生产定点单位证书 2. 国密局商用密码产品销销许可证书 3. 信息安全服务资质认证证书-风险评估（二级） 4. 信息系统集成及服务资质证书（三级） 5. 国家信息安全漏洞库三级技术支撑单位 6. 增值电信业务经营许可证 7. 电信与信息服务业务经营许可证(ICP证) 8. 国家网络与信息安全信息通报机制技术支持单位 | |
| **产品资质** | 移动应用加固服务系统通过了：   1. 中国信息安全测评中心-评估技术报告（EAL2） 2. 中国信息安全测试中心-兼容性测试报告 3. 公安部信息安全等级保护评估中心-功能测试报告 4. 国家信息技术安全研究中心-安全检测报告 5. 中国软件评测中心-技术鉴定测试报告 | |
| **实施案例** | 截止2016年8月，与近150家金融机构(银行、基金、保险、证券、互联网金融)在移动安全领域有合作案例（此处只列举银行案例） | |
| 国有银行 | 3家五大国有银：中国银行、交通银行、建设银行 |
| 股份制银行 | 10家全国股份制银行：平安银行，光大银行，恒丰银行，浙商银行，渤海银行，招商银行，广发银行，华夏银行，兴业银行 |
| 城市商业银行 | 50多家城市商业银行：北京银行，江苏银行，南京银行，宁波银行，齐鲁银行，潍坊银行，齐商银行，临商银行，日照银行，泰安商行，德州银行，东营银行，莱商银行，烟台银行，济宁银行，威海商行，枣庄银行，甘肃银行，兰州银行，汉口银行，桂林银行，富滇银行，石嘴山银行，西安银行，华融湘江银行，洛阳银行，贵阳银行，攀枝花银行，吉林银行，上饶银行，辽阳银行，葫芦岛银行，邢台银行，澳门国际银行，唐山银行，德阳银行，晋城银行，南充商行，中原银行，乐山银行，徽商银行，江苏长江商业银行，民泰商行，鄂尔多斯银行…等等 |
| 农信社、农商行 | 近20家省级农合机构：甘肃省农信联社，湖北省农信联社，山东省农信联社，浙江省农信联社，福建省农信联社，深圳农商行，上海农商行，贵州省农信联社，江苏常熟农商行，江苏无锡农商行，黄河农商行，青海省农信联社，陕西省农信联社，湖南省农信联社，江西省农信联社，云南瑞丽南屏农商行…等等 |
| **部参公关能力** | 1. 是人行、银监会、工信部、公安部、铁路公司等监管单位及国家部委的移动应用安全技术支持商； 2. 银监会/公安部等级保护标准制定商； 3. 与国家信息技术安全研究中心、公安部信息安全等级保护评估中心、中国软件测评中心等多家权威安全机构进行长期的技术合作 | |
| **是否具有保险索赔服务** | 梆梆安全已向中国人保财险投保，因被成功脱壳、壳防护被攻破、核心技术人员离职从事非法行为等原因而直接造成贵行资金损失后的赔付问题将由人保财险负责承保。 | |