# 总体框架图

Java 接口

Java钩子代码

C++钩子代码

C ++ 接口

图 1

1. Java接口。Java接口为java层代码提供规则存储和查询服务。它通过content provider打开文件，实现文件的写操作，从而存储和删除规则。通过 C ++接口查询是否禁用当前功能。
2. Java content provider目前是为java接口和C ++接口提供文件创建和打开接口， 并不实现数据库的操作，只利用数据库接口实现文件操作。
3. C ++ 接口为C ++钩子代码和Java接口提供服务。它通过content provider打开文件，实现文件的只读操作，从而查询是否禁用当前功能。

# Java 接口为AEMM client提供下面规则存储功能。

* AEMM components can’t be deleted
* Block and unblock install for not enterprise application
* Call block and unblock
* Camera block and unblock
* Video recorder block and unblock
* Audio recorder block and unblock
* MMS block and unblock
* Not enterprise application block and unblock
* SMS block and unblock
* AEMM Wifi profile can’t be deleted by Third-party applications
* AEMM VPN profile can’t be deleted by Third-party applications
* AEMM APN profile can’t be deleted by Third-party applications
* EMail

# java接口提供给AEMM client使用的函数的定义和描述。

|  |  |
| --- | --- |
| 函数定义 | 函数描述 |
| static void addProtectedAEMMComponent(ContentProvider resolve, String component); | 添加受保护的AEMM模块。输入参数是模块名称。 |
| static void deleteProtectedAEMMComponent(ContentProvider resolve, String component); | 删除受保护的AEMM模块。输入参数就是模块名称。 |
| static void addProhibitedApk(ContentProvider resolve, String Apk); | 添加禁止安装的安装包名字。输入参数是安装包名称。 |
| static void deleteProhibitedApk(ContentProvider resolve, String Apk); | 删除禁止安装的安装包名字。输入参数是安装包名称。 |
| static void enableDial(ContentProvider resolve, bool enabled); | 启用和禁用拨号。输入参数true- 允许，false-禁用。 |
| static void enableCamera(ContentProvider resolve, bool enabled); | 启用和禁用相机拍照和录像。输入参数true- 允许，false-禁用。 |
| static void enableAudio(ContentProvider resolve, bool enabled); | 启用和禁用声音。输入参数true- 允许，false-禁用。 |
| static void enableVideo(ContentProvider resolve, bool enabled); | 启用和禁用视频 。输入参数true- 允许，false-禁用。 |
| static void enableSMS(ContentProvider resolve, bool enabled); | 启用和禁用短信。输入参数true- 允许，false-禁用。 |
| static void enableMMS(ContentProvider resolve, bool enabled); | 启用和禁用彩信。输入参数true- 允许，false-禁用。 |
| static void addProhibitedApplication(ContentProvider resolve, String application); | 添加禁止启动的程序。输入参数是程序名称。 |
| static void deleteProhibitedApplication(ContentProvider resolve, String application); | 删除禁止启动的程序。输入参数是程序名称。 |
| static void addProhibitedActivity(ContentProvider resolve, String activity); | 添加禁止启动的Activity。输入参数是Activity名称。 |
| static void deleteProhibitedActivity(ContentProvider resolve, String activity); | 删除禁止启动的Activity。输入参数是Activity名称。 |
| static void addAemmWifi(ContentProvider resolve, String key); | 添加要保护wifi配置。输入参数wifi配置名字。 |
| static void deleteAemmWifi(ContentProvider resolve, String key); | 删除要保护的wifi配置。输入参数wifi配置名字。 |
| static void addAemmVPN(ContentProvider resolve, String key); | 添加要保护VPN配置。输入参数VPN配置名字。 |
| static void deleteAemmVPN(ContentProvider resolve, String key); | 删除要保护的VPN配置。输入参数VPN配置名字。 |
| static void addAemmAPN(ContentProvider resolve, String key); | 添加要保护APN配置。输入参数APN配置名字。 |
| static void deleteAemmApVPN(ContentProvider resolve, String key); | 删除要保护的APN配置。输入参数APN配置名字。 |
| static void addAemmEmail(ContentProvider resolve, String key); | 添加要保护Email配置。输入参数Email配置名字。 |
| static void deleteAemmEmail(ContentProvider resolve, String key); | 删除要保护的Email配置。输入参数Email配置名字。 |

表 1

# Java 接口提供给Java钩子代码的接口函数描述如下。

|  |  |
| --- | --- |
| 函数定义 | 函数描述 |
| static boolean isProtectedAEMMComponent(String component); | 查询该是否是受保护的模块。输入参数是模块名字，返回参数true-是，false-不是。 |
| static boolean canInstallApk(String Apk); | 查询式否是禁止安装的包。输入参数是安装包名字，返回参数true-是，false-不是。 |
| static boolean isDialEnabled(); | 查询是否禁用拨号。输入参数无。返回参数true-是，false-不是。 |
| static boolean isSMSEnabled (); | 查询是否禁用短信。输入参数无。返回参数true-是，false-不是。 |
| static boolean isMMSEnabled (); | 查询是否禁用彩信。输入参数无。返回参数true-是，false-不是。 |
| static boolean canStartApplication(String application); | 查询式否是禁止启动的程序。输入参数是程序名字，返回参数true-是，false-不是。 |
| static boolean isProhibitedActivity(String activity); | 查询式否是禁止启动的Activity。输入参数是Activity名字，返回参数true-是，false-不是。 |
| static boolean canChangeWifi (String key); | 查询是否允许修改或删除wifi配置。输入参数wifi配置名字。返回参数true-允许，false-不允许。 |
| static boolean canChangeVPN(String key); | 查询是否允许修改或删除vpn配置。输入参数wifi配置名字。返回参数true-允许，false-不允许。 |
| static boolean canChangeAPN(String key); | 查询是否允许修改或删除APN配置。输入参数wifi配置名字。返回参数true-允许，false-不允许。 |
| static boolean canChangeEmail(String key); | 查询是否允许修改或删除Email配置。输入参数Email配置名字。返回参数true-允许，false-不允许。 |

表2

Java 钩子代码过这些接口函数判断这些功能是否已禁用。

# C ++ 接口提供给 C++钩子代码的接口

|  |  |
| --- | --- |
| 函数定义 | 函数描述 |
| static bool isTakePictureEnabled(); | 查询是否禁用拍照。输入参数无。返回参数true-是，false-不是。 |
| Static bool isVideoRecorderEnabled(); | 查询是否禁用视频录制。输入参数无。返回参数true-是，false-不是。 |
| Static bool isAudioRecorderEnabled(); | 查询是否禁用声音录制。输入参数无。返回参数true-是，false-不是。 |

C ++ 钩子代码过这些接口函数判断这些功能是否已禁用。

# 规则的添加,修改和查询

每一个功能对应一个规则文件，所有的规则文件都要保存到content provider私有目录，文件访问权限控制通过content provider 来控制。每一个规则文件必须存在，如果规则文件不存在，对应的所有功能默认认为是要禁掉。

1. AEMM保护模块规则

创建文本文件aemmcomnent，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行就是一个要保护的模块名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. 安装包禁止安装规则

创建文本文件aemmapkname，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行就是一个要禁止安装的程序包的模块名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. 拨号规则

创建二进文件aemmdial,文件大小是一个字节,内容为0则表示允许打电话,内容为非0的任何数表示禁止打电话。

1. 相机拍照规则

创建二进文件aemmcamera,文件大小是一个字节,内容为0则表示允许拍照,内容为非0的任何数表示禁止拍照。

1. 禁止声音规则

创建二进文件aemmaudio,文件大小是一个字节,内容为0则表示允许录音,内容为非0的任何数表示禁止声音。

1. 禁止视频规则

创建二进文件aemmvideo,文件大小是一个字节,内容为0则表示允许视频,内容为非0的任何数表示禁止视频。

1. SMS规则

创建二进文件aemmsms,文件大小是一个字节,内容为0则表示允许发短信,内容为非0的任何数表示禁止发短信。

1. MMS规则

创建二进文件aemmmms,文件大小是一个字节,内容为0则表示允许发彩信,内容为非0的任何数表示禁止发彩信。

1. 应用程序禁止启动规则

创建文本文件aemmappname，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行就是一个要禁止启动的程序名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. Activity禁止启动规则

创建文本文件aemmactivity，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行就是一个要禁止启动的Activity名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. wifi规则

创建文本文件aemmwifi，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行是wifi的配置名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. VPN规则

创建文本文件aemmvpn，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行是vpn的配置名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. APN规则

创建文本文件aemmapn，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行是apn的配置名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

1. EMail规则

创建文本文件aemmemail，文件的每一行以换行(’\n’)结尾，不能用回车(‘\r’)。每一行是email的配置名字。添加规则的时候要先匹配是不是该规则已经存在，不能重复添加。

# 钩子代码

所谓钩子代码，是指在原有的android的代码里面嵌入AEMM的代码。这部分代码通过调用c ++的接口函数来确定是否中止原来android代码的继续执行或者传入一个错误参数使后续代码执行发生错误从而达到禁用该功能的目的。

1. 禁止拨号举例，用户打开拨号程序，输入对方手机号码，然后按拨号键，

在程序内部正常流程应该是，程序最后会执行ril.java里面的dial函数。dial函数的定义为void dial(String address, ….); dial函数会继续往下调用，直到完成拨号。

嵌入AEMM代码后，dial函数就变成：

void dial(String address, …) {

if(!isEnabledDial()) { // 通过content provider的接口函数判断拨号功能是否禁用

address = “Not permitted”; // 传入一个错误参数

}

…

}

1. 发短信的例子

正常流程代码一定要走到SmsManager.java里sendTextMessage函数，函数定义为void sendTextMessage(…);

嵌入AEMM代码后函数变为

void sendTextMessage(…) {

if(!isEnabledSMS()) {{ // 通过content provider的接口函数判断拨号功能是否禁用

throw new IllegalArgumentException("Not permitted");// 中止执行  
 }

}

1. 相机拍照举例

用户按下相机快门，这条指令通过系统传到相机，相机获取照片之后又通过系统往回返，这是通过一个回调函数来实现的。函数定义为void JNICameraContext::copyAndPost(JNIEnv\* env, const sp<IMemory>& dataPtr, int msgType);

在这个函数内部嵌入代码

static bool isEnableCamera() {

unsigned char benable = 0;

String16 url16("content://com.xuye.TestContentProvider/camera");

int fd = android::openContentProviderFile(url16);

if (fd >= 0) {

lseek(fd, 0, SEEK\_SET);

read(fd, (char \*)&benable, sizeof(benable));

close(fd);

}

return (benable == 0);

}

JNICameraContext::copyAndPost(JNIEnv\* env, const sp<IMemory>& dataPtr, int msgType) {

…

If(!isEnableCamera()) {

//修改原来数据传回一个禁止拍照照片

}

…

}

如上所示，红色部分就是我们的钩子代码。