合作开发文档

# 总体方案

中国科学院信息工程研究所和深圳市同洲电子股份有限公司就“研制基于国产自主操作系统的海云安全智能手机”进行技术合作，共同开发手机软硬件资源管控子系统。其中，同洲电子负责手机端管控应用的开发，信工所负责子系统的功能、架构设计和管控云平台的开发并负责向同洲电子开放手机端管控应用所需的云接口。

手机端管控应用和服务器端的通信采用webservice接口进行通信，具体方式分为POST和GET。POST为手机端管控应用主动向服务器端发送数据，GET为手机端管控应用接受服务器端发送的指令或数据。服务器端会通过推送服务器向手机端发布消息（具体流程详见附件4 消息推送），手机端收到消息后，再调用相关GET方式的接口接收具体的指令或数据。

手机软硬件资源管控子系统整体功能架构及所需接口如下图、表所示。



图1 手机软硬件资源管控子系统架构

表1 手机软硬件资源管控子系统功能和接口列表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **一级功能** | | **二级功能** | **功能描述** | **接口** | |
| **登录** | | 用户登录 | 手机端管控应用输入用户名和密码，实现登录工作模式 | POST /a/wp/user/login | |
| 注销登录 | 点击注销登录/退出按钮，实现退出功能 | POST /a/wp/user/logout | |
| 修改密码 | 输入旧密码及符合规则的新密码，以修改密码 | POST /a/wp/user/set\_pw | |
| **设备管控** | | 发送设备信息 | 用来发送设备的静态信息 | POST /a/wp/user/dev\_static\_info | |
| 发送位置信息 | 定时向服务器端发送手机当前的位置信息 | POST /a/wp/user/loc | |
| wifi管控 | 获取到wifi的管控指令，控制手机的wifi功能禁用/非禁用 | GET /a/wp/user/get\_cmds  POST /a/wp/user/complete\_cmd  POST /a/wp/user/set\_cmd\_respons | |
| 蓝牙管控 | 获取到蓝牙的管控指令，控制手机的蓝牙功能禁用/非禁用 |
| 摄像头管控 | 获取到摄像头的管控指令，控制手机的摄像功能禁用/非禁用 |
| GPS管控 | 获取到GPS的管控指令，控制手机的GPS禁用/非禁用 |
| 录音管控 | 获取到录音的管控指令，控制手机的录音功能禁用/非禁用 |
| 数据流量管控 | 获取到数据流量的管控指令，控制手机的数据流量功能禁用/非禁用 |
| USB调试管控 | 获取到USB调试的管控指令，控制手机的USB调试功能禁用/非禁用 |
| USB连接管控 | 获取到USB连接的管控指令，控制手机的USB连接功能禁用/非禁用 |
| 推送通知 | 接收服务器端推送过来的通知，并在手机上显示 | GET /a/wp/user/get\_cmds  POST /a/wp/user/complete\_cmd | |
| 屏幕锁定 | 接收锁定命令，强制进入锁屏状态，重新输入密码解锁 |
| 锁屏密码修改 | 用来修改用户的锁屏密码。（现有功能是从web端修改和手机端不通过服务器自己修改） |
| 远程遥毁 | 收到指令后，清除所有的隐私信息和工作数据 |
| **安全策略** | | 获取策略 | 通过策略id，从数据库中取需要执行的策略 | GET /a/wp/user/get\_strategy\_by\_id1 | |
| 完成策略 | 当执行完策略时，向服务器端反馈执行结果 | POST/a/wp/user/complete\_strategy | |
| 删除过期策略 | 删除已过期的策略 | POST /a/wp/user/del\_user\_strategy | |
| 获取当前策略信息列表 | 通过设备的id，获得相应设备的当前策略列表 | GET /a/wp/user/get\_user\_strategys | |
| 获取删除策略 | 当服务器对策略进行删除操作后，会通知手机端删除相应的策略 | GET /a/wp/user/get\_user\_need\_del\_strategys | |
| **同步联系人** | | 获取同步联系人 | 手机端从服务器获取要同步的联系人（群组） | GET /a/wp/user/get\_contacts | |
| **应用管理** | 提交APP信息 | 获取手机端现有的APP信息，并上传至云端 | POST /a/wp/user/dev\_app\_info |
| 获取应用列表 | 手机端通过应用id获取列表中所有应用的信息 | GET /a/was/appslist |
| 获取应用地址 | 手机端获取应用地址，从而下载该应用，并自动提示用户安装 | GET /a/was/get\_app\_addr |
| **安全审计** | | 提交审计日志 | 监测手机端安全行为，生成审计日志并提交给服务器 | POST /a/wp/user/log | |
| **数据防护** | | 数据备份 | 将手机本地数据上传到服务器端 | 提交表单 POST http://<server-ip>:8083/f\_backup | |
| 数据获取 | 将服务器端备份的数据取回本地 | GET /a/wp/user/get\_dev\_data | |

# 登录接口

## 用户登录

调用端：

手机端管控应用

请求：

POST /a/wp/user/login HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

uid: “用户的电话号码”

pw: “用户的登陆密码”

dev\_id:”用户的设备ID

返回值：

Content-Type: application/json

{

rt: error code, 详见附件3：错误码列表

sid: session id服务器端生成

need\_mod\_pw: 如果为0表示不需要修改密码，如果为1则需要修改密码

}

备注：

❖用户每次登陆验证成功后，后台会为其分配一个唯一的sessionid，移动端的sid是由uid和dev\_id以及一个随机产生的数组成，web端的sid由管理员账号和admin字符串外加一个随机数产生。

❖sid作为客户端和websocket之间连接的channel\_id，只要客户端不重新调用login接口，sid将一直有效。

## 用户登出

调用端：

手机端管控应用

请求：

POST /a/wp/user/logout HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖正常登出会将服务器端存储的此次会话的sessionid删除，下次登录会重新生成一个不同的sessionid

## 修改登录密码

调用端：

手机端管控应用

请求：

POST /a/wp/user/set\_pw HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

oldpw: 旧的密码

newpw: 新的密码

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

# 设备信息上传接口

为了方便对于用户手机的管理，云平台应能够查看用户手机的设备信息，包括：系统版本，设备型号，IMEI号，网络活动，位置信息等，手机端管控应用应能够获取这些信息，并通过调用设备信息上传接口，将相关信息发送至服务器端。

## 3.1 提交设备静态信息（首次登陆提交+信息更新提交）

调用端：

手机端管控应用

请求：

POST /a/wp/user/dev\_static\_info HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

key1: value1

key2: value2

...

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

count: 成功更新的key个数

}

设备静态信息key详见附件2：设备静态信息key

## 3.2 提交设备位置信息（定时提交+位置快速变动提交）

调用端：

手机端管控应用

请求：

POST /a/wp/user/loc HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: sessionid

lon: 经度

lat: 纬度

currenttime: 客户端当前提交时间

offlinetime: 移动端最近一次掉线时间

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖Currenttime/offlinetime为可选参数，默认情况下，currenttime为服务器当前时间，无offlinetime则默认当前状态的前一状态是在线

# 设备资源管控接口

设备资源管控主要包括控制设备敏感资源、获取设备信息、远程锁屏（修改锁屏密码）、远程遥毁等功。

为了保护场所安全，应能够在系统底层对手机中的敏感资源进行控制，需要管控的敏感资源应包含但不限于：**摄像头、录音、蓝牙、WIFI、数据连接、GPS定位、录音、USB调试模式、USB连接等**，管控内容主要是对上述四项敏感资源进行“禁用/非禁用”控制，拦截上层应用的调用请求，并给用户提示。

系统应开放敏感资源控制接口，供设备管控系统的服务器端调用，同时提供本地接口，允许用户通过手机端的管控软件控制本机敏感资源，但应确保服务器端的管控权限高于手机端。

当某项设备资源处于“禁用”状态时，应确保任何上层应用程序都无法打开该资源。当某项设备资源处于“非禁用”状态时，系统应能够对上层应用的调用请求进行拦截，并给用户提示，经用户许可后才能通过该调用请求。同时系统应维护敏感资源调用的黑、白名单，当用户信任某应用时，允许该应用在调用“非禁用”状态的设备资源时，无需经由用户许可（即不弹出用户提示）；反之，当用户将某应用拉入黑名单时，禁止该应用调用“非禁用”状态的设备资源。需要注意的是，黑、白名单仅对“非禁用”状态的设备资源生效，对于“禁用”状态的设备资源，仍应确保任何应用都无法打开。黑、白名单应提供添加和修改的本地接口，允许用户通过手机端的管控软件进行查看和修改。

## 4.1 获取指令列表(管控指令)

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_cmds HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

返回值：

Content-Type: application/json

{

rt: error code,

cmds: [

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”}

]

}

备注：

❖sid为必选参数，为当前设备的SessionID

❖dev\_id为可选参数，默认为当前SessionID所对应的设备ID

❖rt为返回值，详见附录3错误码列表

❖cmds为指令内容，可以是一条或多条，包括摄像头、蓝牙、WiFi等禁用命令（指令列表详见附录1），若与策略相冲突，则可提示强行执行指令。

❖cmds：id为指令执行的序列号，唯一标识一次指令执行

❖客户端读取完所有指令以后就将数据库中的存储指令集删除

## 4.2 完成一个指令

调用端：

手机端管控应用

请求：

POST /a/wp/user/complete\_cmd HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid:sessionID

id:”指令执行的序列号”

results:”0表示成功，1表示失败”

info:”一些说明信息”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖移动端执行完一条指令后，该指令即从数据库中删除。

## 4.3 提交管控指令的标识位值（对单个标识位设置）

调用端：

手机端管控应用

请求：

POST /a/wp/user/set\_cmd\_respons HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

info: wifi/bluetooth/camera

results: 0/1

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖当硬件实时管控与本地正在执行的策略相冲突时调用

❖info为管控指令所对应的键，如：指令为wjy，info就为wifi

❖results为移动端对管控指令的反馈，0表示不冲突，1表示冲突

# 安全策略

为了实现基于时间、场所的自动管控服务，满足用户手机进入安全保密区域时自动切换到960安全OS，系统应开发安全策略功能，由服务器端向手机端下发并由手机端执行。

当手机获取服务器端下发的安全策略后，应检测是否满足策略的条件，包括：策略作用基站、策略作用范围和策略作用时间。若满足策略条件，即手机在策略时间内位于策略区域中，手机端管控应用应该强制将手机从通用Android系统强制切换至960 OS，以实现对策略区域的安全保护。当出了安全区域后，960 OS应提示用户是否转到通用Android系统，并解除对手机端管控应用的管控。

## 5.1 通过strategy\_id获取单条策略（包括修改后的替换）

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_strategy\_by\_id1 HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

strategy\_id: 策略的ID号 时间戳（毫秒级）

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

strategy:{strategy\_id: ,…..}

}

备注：

❖strategy的键值如下：

|  |  |  |
| --- | --- | --- |
| 键 | 值 | 注释 |
| \_id | ObjectId("53bf45759c369a31c93ba77b") | Mongodb自动生成的每条记录的标识符 |
| strategy\_id | 1405044120489 | 时间戳（毫秒级）策略编号 |
| start | 2014-07-11 08:00 | 策略开始时间  格式为%YY-%mm-%dd %HH:%MM  如果为空，则以服务器当前时间为起始时间 |
| end | 2014-07-18 08:00 | 策略结束时间，格式同上  如果为空，则以起始时间后一个月为截止时间 |
| lon | 116.220686 | 策略有效中心点经度 |
| lat | 39.979471 | 策略有效中心点纬度 |
| desc | 123 | 对中心点的位置描述，如：中国科学院信工所 |
| *baseStationId* | plmn-46002-lac-41057-cid-5050519#plmn-46001-lac-41014-cid-4626131 | 策略做用的基站 |
| radius | 100 | 策略以中心点为圆心的有效半径，单位为米 |
| wifi | wfjy | 敏感资源管控指令，wifi，蓝牙，相机，GPS和录音的管控 |
| bluetooth | bfjy |
| camera | cfjy |
| gps | gfjy |
| tape | tfjy |

## 5.2 获取用户需要删除的策略ID集合

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_user\_need\_del\_strategys HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

dev\_id: 设备ID （可选）

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

ids:[{“strategy\_id”: id},{},…]

}

备注：

❖返回值ids为该用户需要在本地删除的策略ID集合

❖id为策略ID号，格式为时间戳（毫秒级）

## 5.3 完成一条策略

调用端：

手机端管控应用

请求：

POST /a/wp/user/complete\_strategy HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid:sessionid

id:"strategy call id"

results: 0表示成功，1表示失败

info：一些描述信息

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖将该条策略ID标记为已读，当客户端实时读取一条策略后调用

## 5.4 用户删除过期策略

调用端：

手机端管控应用

请求：

POST /a/wp/user/del\_user\_strategy HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid:sessionid

id:"strategy id

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖当本地策略到期后，客户端调用来维护dev表，即从dev表中删除该策略ID

## 5.5 获取设备未读的策略集（备用）

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_user\_strategysHTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

dev\_id: 设备ID

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

strategys:[{},{},…]

}

备注：

❖strategys中{}的结构格式同获取单条策略的strategy格式

❖为了节省移动端的数据流量和减少冗余无用信息，strategy中没有users和userdesc两个键值对

❖参数dev\_id为可选参数

❖返回的strategys只包含该用户未读的策略信息

❖调用完毕后，会将这些策略标记为已读，下次获取时如果服务器端没有对其做修改，将不会再返回该策略信息，反之则会再次返回

# 同步联系人接口

为了方便群组用户使用，可以通过云平台同步的方式，对群组内用户手机的联系人进行增、删、改，具体包括：群组联系人同步和个体联系人同步。群组同步时，应对整个群组联系人列表进行更新，包括：新增、替换和删除。个体同步时，应对进行同步的联系人信息进行更新，包括：新增和替换。

## 6.1 获取同步联系人

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_contacts HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: Sessionid

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

contacts: [{},{},{},…]

is\_group: 0/1

}

备注：

❖contacts为返回的联系人信息

具体格式为：{

email: 注册邮箱 如：[\*\*\*\*\*\*\*\*\*@iie.ac.cn](mailto:*********@iie.ac.cn)

name:姓名 中英文都可

job:职位/职称 如：组长/工程师

department：所在部门 如：中科院信工所，四室，无线组

pnumber:联系人移动电话

}

❖is\_group为群组同步/个体同步的标识，其中0表示个体同步，1表示群组同步。

# 数据防护接口

为了保障用户隐私信息和工作数据的安全，应对手机中的相关数据进行防护。

1. 960 OS下的数据：包括密联系人、密短信、密通话记录和在960 OS下生成的数据（下载的邮件文件、处理的工作文件以及用户在960 OS下收发的明短信，生成的明通话记录等），在用户退出960 OS时应建议用户将数据上传到服务器端保存（**在连接wifi网络的前提下**），本地进行销毁，防止手机端的信息泄露。在下次用户登入960 OS时，直接访问安全存储在服务器端的数据或将数据同步到本地（**在连接wifi网络的前提下**）。在上传和同步过程中，应对数据进行加密处理，防止网络侧的攻击。如果用户选择存储在本地（**或在无法连接wifi网络的前提下**），则应对960 OS中的数据进行隔离加密存储，严禁生活模式中的应用访问。
2. 通用Android下的数据：包括明联系人，在通用Android下收发的明短信、生成的明通话记录和文件（照片等）应提供云备份功能。

为了防止因手机遗失造成的失、泄密事件发生，系统应开发数据销毁功能（擦除并重写，防止通用的数据恢复手段），并提供远程调用接口（webservice和短信），供服务器端调用。

## 数据备份接口

调用端：

手机端管控应用

请求：

POST表单到地址：

http://<server-ip>:8083/f\_backup

Content-Type: multipart/form-data

参数：

sid：SessionID

dev\_id:”目标设备ID”

backuptime:”备份时间戳（int型）”

backupfile:”提交的文件”（文件格式无要求）

data\_verify:”用于校验的数组”，如：filename:filemd5

返回值：

由推送服务器推送的消息，详见附录4，其中推送的内容为：

text：“backup\_success”/”backup\_failed”

分别表示：备份成功/备份失败

备注：

* 数据备份过程中，当POST表单时，提交的备份文件应打包成一个压缩文件（按需求加密），云端只安全的存储备份数据，将不对备份数据的内容进行读取。
* 备份时间戳为数据恢复的依据之一，类似的备份信息将由手机端管控应用本地维护，云端只负责文件的存储与校验。
* 在提交的数据中，数组data\_verify：”filename,filemd5”用于文件的校验，其中filename为提交的压缩文件名（由手机端按类别统一命名），filemd5为该文件的md5校验码。
* 当云端接受数据完成时，根据md5校验的结果来判断当次备份是否成功，并生成消息由推送服务器推送给手机端，若当次备份失败，手机端根据需要可再次向云端申请备份。

## 数据获取接口

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_dev\_data HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

filename:”待恢复的数据的压缩文件名”

backuptime:”待恢复的数据的备份时间戳”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

PATH: [filepath,filemd5]

}

备注：

* 云端将根据发来的参数返回数组：PATH: [filepath,filemd5]。其中filepath为备份数据的下载路径，手机端需要根据此路径来获取要恢复的数据；filemd5为文件的md5校验码，用于手机端接收完数据后对文件完整性的判断。
* 手机获取用于恢复数据的路径为：

http://<server-ip>:8083/backup/(filepath)

## 远程遥毁接口

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_cmds HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

cmds: [

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”}

]

}

备注：

❖dev\_id为可选参数，默认为当前session所在设备

❖id为指令执行的序列号，唯一标识一次指令执行

❖客户端读取完所有指令以后就将数据库中的存储指令集删除

❖**cmd指令为reset**

# 安全审计接口

为了保障手机安全，960 OS应对应用程序访问手机敏感资源和数据进行审计，生成审计日志并上传到云平台。需要审计的行为包括但不限于：应用程序访问手机的摄像头、蓝牙、wifi、录音、GPS定位、电话、短信、通讯录、通话记录、互联网（网络行为）、SD卡中的用户工作文件（特定区域）等。审计日志的内容应包括：操作发生的时间，操作类型，操作者（应用程序包名），操作对象（敏感资源和数据）和操作结果。960 OS应调用审计日志上传接口，将审计日志上传至云端进行记录与备份。

## 8.1记录审计日志

调用端：

手机端管控应用

请求：

POST /a/wp/user/ log HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: “session id”,

logs:[{op\_user:”操作者”,op\_time:"操作时间",op\_type:"操作类型",op\_desc:"操作对象/操作描述",op\_result:"操作结果"},{}..]

返回值：

Content-Type: application/json

{

rt: error code, 详见附件：错误码列表

}

# 应用管理

## 提交APP信息接口

调用端：

手机端管控应用

请求：

POST /a/wp/user/dev\_app\_info HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: session id

apps: 应用列表的json字符串[{},{},{}...]

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

}

备注：

❖每个app信息的JSON结构如下：

{

"app\_id":"com.singuloid.sample",

"app\_name":"例子"，

"version\_name":"1.0",

"version\_code":100,

"icon":"fdsafdadfsasfadsf"，

"last\_update\_time":10000

}

❖具体键值含义如下表：

|  |  |  |
| --- | --- | --- |
| 1 | app\_id | 包名 |
| 2 | app\_name | 应用名 |
| 3 | version\_name | 版本名称 |
| 4 | version\_code | 版本代码（int型） |
| 5 | icon | base64编码的图片文件png |
| 6 | last\_update\_time | 最后更新的时间(long型)（Unixtime） |

## 获取应用列表接口

调用端：

手机端管控应用

请求：

GET /a/was/appslist HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sort: utime|name（排序方式）

is\_asc: yes|no（是否是正序）

start:开始的地方

count:获取数量

key: 查询关键字

返回值：

Content-Type: application/json

{

rt: error code, 详见附件：错误码列表

}

备注：

❖客户端获取应用列表

## 获取应用地址接口

调用端：

手机端管控应用

请求：

GET /a/was/get\_app\_addr HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid：身份验证

返回值：

Content-Type: application/json

{

rt: error code, 详见附件：错误码列表

apps:[{app\_id,versionCode,url}]

}

备注：

当应用推送和系统更新的命令发出后，手机端会从推送服务器得到内容为”apps”的消息，当解析到这个消息时，手机端应调用此接口，并用该接口的URL下载应用或系统应用。

❖app\_id为应用id

❖versionCode为版本代码（int型）

**❖当为推送应用时，url为:8083/download/apps/xxxx.apk**

**❖当为系统更新时，url为:8083/update/xxxx.apk**

# 消息推送与远程锁屏

## 消息推送接口

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_cmds HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

cmds: [

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”}

]

}

备注：

❖dev\_id为可选参数，默认为当前session所在设备

❖id为指令执行的序列号，唯一标识一次指令执行

❖客户端读取完所有指令以后就将数据库中的存储指令集删除

❖**cmd指令为url<URL>,<URL>为消息内容**

## 远程锁屏接口

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_cmds HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

cmds: [

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”}

]

}

备注：

❖dev\_id为可选参数，默认为当前session所在设备

❖id为指令执行的序列号，唯一标识一次指令执行

❖客户端读取完所有指令以后就将数据库中的存储指令集删除

❖**cmd指令为lockscreen**

## 远程修改锁屏密码接口

调用端：

手机端管控应用

请求：

GET /a/wp/user/get\_cmds HTTP/1.1

Content-Type: application/x-www-form-urlencoded

参数：

sid: SessionID

dev\_id:”目标设备ID”

返回值：

Content-Type: application/json

{

rt: error code, 详见：错误码列表

cmds: [

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”},

{id:”id”, cmd:”cmd”}

]

}

备注：

❖dev\_id为可选参数，默认为当前session所在设备

❖id为指令执行的序列号，唯一标识一次指令执行

❖客户端读取完所有指令以后就将数据库中的存储指令集删除

❖**cmd指令为chg\_screen\_pw:xxxxxxxx**

# 密码模块

为了确保手机端管控应用和云端通信的安全，本系统设计并添加了密码模块和安全服务平台。其中，密码模式分别放置在手机端和服务器端，用于管理证书和完成加解密操作，安全服务平台部署在信工所，用于签发证书。最终通过给传输的指令和数据信息进行加密，实现通信安全。

## 初始化接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | initAgent | | |
| 接口声明 | public int initAgent(String appID,String businessId,String userId, String serviceURL, boolean testFlag) | | |
| 接口功能 | 启动应用时，安全桌面调用初始化接口，对应用进行自验证，验证应用是否合法。 | | |
| 参数表：  参数名称/参数类型/参数说明 | appid | String | 应用标识(201403211138130121687) |
| businessId | String | 业务Id(8位数字) |
| userId | String | 用户Id (TelephonyMgr.getDeviceID)  IMEI(GSM)或MEID/ESN(CDMA) |
| serviceURL | String | 安全服务平台的地址（124.207.215.118:7080） |
| testFlag | Boolean | 测试标识，true为测试状态，false为正式状态(false) |
| 返回值 | 0：初始化成功 | | |

## 指令加密接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | instructCrpRequest | | |
| 接口声明 | public String instructCrpRequest (String src,String serviceId) | | |
| 接口功能 | 设备管控客户端将服务器Id，明文指令等内容传给本地密码模块，由本地密码模块将明文指令加密并签名后，将密文回传给设备管控客户端 | | |
| 参数表：  参数名称/参数类型/参数说明 | Src | String | 待加密的指令 |
| serviceId | String | 服务器ID(123456) |
| 返回值 | 加密后的安全短信/指令密文（密文格式：密文|签名值） | | |

## 指令解密接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | instructDecrpRequest | | |
| 接口声明 | public String instructDecrpRequest (String encryptData,String serviceId) | | |
| 接口功能 | 设备管控客户端将密文指令内容以及服务器证书传给本地密码模块，由本地密码模块将密文指令解密并检验签名后，将明文回传给设备管控客户端 | | |
| 参数表：  参数名称/参数类型/参数说明 | encryptData | String | 需要解密的密文 |
| serviceId | String | 服务器ID |
| 返回值 | 解密后的安全指令明文 | | |

# 附录1 管控指令列表：

|  |  |  |
| --- | --- | --- |
| 序号 | 指令代码 | 注释 |
| 1 | bjy | 蓝牙禁用 |
| 2 | bfjy | 蓝牙非禁用 |
| 3 | wjy | WIFI禁用 |
| 4 | wfjy | WIFI非禁用 |
| 5 | tjy | 录音禁用 |
| 6 | tfjy | 录音非禁用 |
| 7 | cjy | 相机禁用 |
| 8 | cfjy | 相机非禁用 |
| 9 | gjy | gps禁用 |
| 10 | gfjy | gps非禁用 |
| 11 | mjy | 移动数据流量禁用 |
| 12 | mfjy | 移动数据流量非禁用 |
| 13 | ucjy | USB连接禁用 |
| 14 | ucfjy | USB连接非禁用 |
| 15 | udjy | USB调试禁用 |
| 16 | udfjy | USB调试非禁用 |
| 17 | bxfjy | 强制蓝牙非禁用 |
| 18 | wxfjy | 强制wifi非禁用 |
| 19 | txfjy | 强制录音非禁用 |
| 20 | cxfjy | 强制相机非禁用 |
| 21 | gxfjy | 强制gps非禁用 |
| 22 | mxfjy | 强制移动数据流量非禁用 |
| 23 | ucxfjy | 强制USB连接非禁用 |
| 24 | udxfjy | 强制USB调试非禁用 |
| 25 | reset | 远程遥毁指令 |
| 26 | lockscreen | 远程锁屏指令 |
| 27 | unlocksreen | 远程解锁指令 |
| 28 | chg\_screen\_pw | 修改锁屏密码指令 |
| 29 | url<URL> | 推送通知 |
| 30 | install\_app<APP\_ID> | 向手机发送一个安装指定应用的命令 |
| 31 | qccl | 一键清除策略指令 |

# 附录2 设备静态信息key列表

1. dev\_model : 设备型号

2. system\_version : 系统版本

3. core\_version : 内核版本

4. firmware\_version：固件版本

5. UI\_version：UI信息

6. imei : IMEI

7.imsi：IMSI

8. sim\_sn : SIM序列号

9. signal\_strength : 信号强度

10. network\_type : 网络类型

11. phone\_type : 手机类型（手机制式）

12. wifi\_bssid : WIFIBSSID

13. wifi\_ip\_addr : WIFIIP地址

14. wifi\_mac\_addr : WIFIMAC地址

15. wifi\_rssi : WIFIRSSI

16. wifi\_ssid : WIFISSID

17. wifi\_network\_id : WIFI网络id

18. bluetooth\_name : 蓝牙名称

19. bluetooth\_addr : 蓝牙地址

20. bluetooth\_status : 蓝牙状态

21. data\_activity\_status : 数据活动状态

22. data\_conn\_status : 数据连接状态

# 附录3 错误码列表

0, *'正确'*

1, *'未知错误'*

2, *'用户名或密码错误'*

3, *'数据库操作失败'*

4, *'Session数据库操作失败'*

5, *'请求参数错误'*

6, *'用户未登录'*

7, *'权限不足'*

8, *'Session已失效'*

9, *'邮箱配置错误'*

10, *'缺少邮箱配置'*

11, *'EMAIL地址错误'*

12, *'用户已存在'*

15, *'无法向目标发送SMS信息'*

16, *'错误的手机号码'*

17, *'用户不存在'*

18, *'密码不符合规则'*

19, *'旧密码错误'*

20, *'数据类型错误'*

21, *'数据长度错误'*

22, *'此用户设备数据过多'*

23, *'验证码错误'*

24, *'创建验证码失败'*

25, *'错误的短信模板'*

26, *'发送短信时出错'*

27, *'发送短信响应的结果有错误'*

28, *'需要验证码'*

29, *'连接或登录LDAP Server失败'*

30, *'当前用户验证方式不允许此操作'*

31, *'当前版本不允许此操作'*

32, *'修改LDAP中的数据时发生错误'*

33, *'开启自动同步LDAP失败'*

34, *'保存企业LOGO失败'*

36, *'已存在同名APP'*

37, *'不存在的APP'*

38, *'已存在同名组织'*

39, *'不存在的组织'*

40,*'读取LDAP数据失败'*

41,*'用户未在客户端激活'*

42,*'写入LDAP数据失败'*

43,*'删除LDAP失败'*

44,*'文件中section不存在'*

45,*'定位失败'*

46*,'初始化加密机失败'*

47*,'加密或解密失败'*

48*,'反初始化加密机失败'*

49*,'加载共享库失败'*

50*,'解密数据丢包'*

51*,'三次登陆失败'*

52*,'用户已在其它地点登陆'*

# 附录4 消息推送

Push server是基于nginx服务器的第三方模块nginx\_push\_stream\_module来实现的。nginx\_push\_stream\_module使用http技术来实现连接管道，在项目里主要用于实时消息的推送，如命令的下发、策略下发后实时的通知手机客户端等。

nginx\_push\_stream\_module主要采用pub/sub模式来管理长连接，用户（如手机客户端）可以申请连接通道，通道建立后订阅该通道，消息推送者（如服务器端）可以向连接通道发送消息，这样订阅该通道的所有用户可及时接收到该消息。

## 附录4.1 push sever 地址

**<server ip>:<port>**

server ip 为服务器地址，端口号port为8001。

## 附录4.2 订阅消息

接收端：

手机端管控应用

请求：

采用WebSocket协议，连接: ws://<severip>/ws/<channel id>

消息格式：

每当有消息时，就会收到以下json包：

{

id: 消息ID

channel: channel id

text: 消息内容

}

消息内容text说明：

❖cmd 当前设备有新的cmd需要执行

❖NewStrategy:strategy\_id, strategy\_id为策略ID,时间戳（毫秒级）当前设备收到一条新的策略

❖ModStrategy:strategy\_id,当前设备有一条策略被修改

❖DelStrategy:strategy\_id,当前设备有一条策略被删除

❖contacts当前设备有需要同步的联系人

❖apps当前设备有需要下载的应用

❖updata:当前设备有需要下载的应用IP地址:8083/download/updata/xxxx.apk(XXXX.img)