## MTK平台上黑名单功能总结

黑名单作为一项非常重要的功能，一直以来Android原生都没有提供支持，因此各芯片厂商都有自己的实现方案。本文总结了MTK平台芯片在Android 4.4，Android 5.x和Android 6.0这三个版本上黑名单功能的实现方式。

MTK平台上以Content Provider的形式提供黑名单数据库的接口，但在三个版本上实现略有差异。

### Android 4.4

* Number：电话号码
* Type：枚举型，取值为BLOCK\_VOICE\_CALL，BLOCK\_VIDEO\_CALL和BLOCK\_VOICE\_AND\_VIDEO\_CALL三种，可见视频电话与普通电话黑名单是可以分开管理的

### Android 5.x

* Number：电话号码

Android 5.x引入了MPlugin架构，将黑名单数据库放到OPxxPlugin中实现，与Android 4.4相比黑名单数据库变化不大，只是移除Type字段，不再区分视频电话和普通电话。

### Android 6.0

* PHONE\_NUMBER：电话号码
* NAME：联系人姓名，可为空

Android 6.0将黑名单功能从OPxxPlugin中独立出来，放到单独的Blacklist模块中实现，并且数据库变化也非常大。

表1 MTK平台上三个版本黑名单对比

|  |  |  |  |
| --- | --- | --- | --- |
|  | Android 4.4 | Android 5.x | Android 6.0 |
| **CONTENT\_URI** | content://reject/list | content://reject/list | content://com.cmcc.ccs.black\_list |
| **表结构** | Number  Type | Number | PHONE\_NUMBER  NAME |
| **访问权限** | READ\_REJECT & WRITE\_REJECT | READ\_REJECT & WRITE\_REJECT | 无 |
| **实现类** | 未知 | OP01Plugin.apk  com.mediatek.phone.callrejection.CallRejectContentProvider | Blacklist.apk  com.mediatek.rcs.blacklist.BlacklistProvider |
| **代表芯片** | MT6582 | MT6753 | MT6755 |

### 查询黑名单

黑名单数据库中存储的是原始的电话号码，并未进行格式化，因此不能直接通过query方式查询某个号码是否黑名单，而必须读出黑名单中的所有电话号码，再通过PhoneNumberUtils.compare方法逐个进行模糊匹配。

***public******boolean******[autoReject](http://172.16.58.59:8080/source/s?refs=autoReject&project=s2m)****(*[*String*](http://172.16.58.59:8080/source/s?defs=String&project=s2m)***[number](http://172.16.58.59:8080/source/s?refs=number&project=s2m)****) {*

[*Cursor*](http://172.16.58.59:8080/source/s?defs=Cursor&project=s2m)[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m) *=* [*null*](http://172.16.58.59:8080/source/s?defs=null&project=s2m)*;*

[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m) *=* [***mContext***](http://172.16.58.59:8080/source/xref/s2m/alps/vendor/mediatek/proprietary/operator/OP01/packages/apps/Plugins/src/com/mediatek/phone/plugin/Op01IncomingCallExt.java#mContext)*.*[*getContentResolver*](http://172.16.58.59:8080/source/s?defs=getContentResolver&project=s2m)*().*[*query*](http://172.16.58.59:8080/source/s?defs=query&project=s2m)*(*[*Uri*](http://172.16.58.59:8080/source/s?defs=Uri&project=s2m)*.*[*parse*](http://172.16.58.59:8080/source/s?defs=parse&project=s2m)*(*[***RCS\_BLACK\_LIST\_URI***](http://172.16.58.59:8080/source/xref/s2m/alps/vendor/mediatek/proprietary/operator/OP01/packages/apps/Plugins/src/com/mediatek/phone/plugin/Op01IncomingCallExt.java#RCS_BLACK_LIST_URI)*),*

[***RCS\_BLACK\_LIST\_PROJECTION***](http://172.16.58.59:8080/source/xref/s2m/alps/vendor/mediatek/proprietary/operator/OP01/packages/apps/Plugins/src/com/mediatek/phone/plugin/Op01IncomingCallExt.java#RCS_BLACK_LIST_PROJECTION)*,* [*null*](http://172.16.58.59:8080/source/s?defs=null&project=s2m)*,* [*null*](http://172.16.58.59:8080/source/s?defs=null&project=s2m)*,* [*null*](http://172.16.58.59:8080/source/s?defs=null&project=s2m)*);*

***if*** *(*[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m) *==* [*null*](http://172.16.58.59:8080/source/s?defs=null&project=s2m)*) {*

***return******false****;*

*}*

[*String*](http://172.16.58.59:8080/source/s?defs=String&project=s2m)[*blockNumber*](http://172.16.58.59:8080/source/s?defs=blockNumber&project=s2m)*;*

***boolean***[*result*](http://172.16.58.59:8080/source/s?defs=result&project=s2m) *=* ***false****;*

[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m)*.*[*moveToFirst*](http://172.16.58.59:8080/source/s?defs=moveToFirst&project=s2m)*();*

***while*** *(![cursor](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m).*[*isAfterLast*](http://172.16.58.59:8080/source/s?defs=isAfterLast&project=s2m)*()) {*

[*blockNumber*](http://172.16.58.59:8080/source/s?defs=blockNumber&project=s2m) *=* [*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m)*.*[*getString*](http://172.16.58.59:8080/source/s?defs=getString&project=s2m)*(0);*

***if*** *([PhoneNumberUtils](http://172.16.58.59:8080/source/s?defs=PhoneNumberUtils&project=s2m).*[*compare*](http://172.16.58.59:8080/source/s?defs=compare&project=s2m)*(*[***number***](http://172.16.58.59:8080/source/xref/s2m/alps/vendor/mediatek/proprietary/operator/OP01/packages/apps/Plugins/src/com/mediatek/phone/plugin/Op01IncomingCallExt.java#number)*,* [*blockNumber*](http://172.16.58.59:8080/source/s?defs=blockNumber&project=s2m)*)) {*

[*result*](http://172.16.58.59:8080/source/s?defs=result&project=s2m) *=* ***true****;*

***break****;*

*}*

[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m)*.*[*moveToNext*](http://172.16.58.59:8080/source/s?defs=moveToNext&project=s2m)*();*

*}*

[*cursor*](http://172.16.58.59:8080/source/s?defs=cursor&project=s2m)*.*[*close*](http://172.16.58.59:8080/source/s?defs=close&project=s2m)*();*

***return***[*result*](http://172.16.58.59:8080/source/s?defs=result&project=s2m)*;*

*}*

### Android N

Android N原生提供黑名单数据库支持，实现类在android.provider.BlockedNumberContract，提供的接口主要有：

* canCurrentUserBlockNumbers：判断能否访问黑名单数据库，只有默认的电话/短信等少数应用才可以访问
* isBlocked：判断某个号码是否黑名单

BlockedNumberContract只保存黑名单电话号码，不会自动关联联系人，并且除了保存原始电话号码在COLUMN\_ORIGINAL\_NUMBER外，还自动生成并保存格式化的电话号码在COLUMN\_E164\_NUMBER。