

# 手游智能反外挂Android端接入文档（java）

## 一、接入说明

接入反外挂SDK，开发者需要完成以下步骤

1. 根据游戏开发平台将SDK拷贝到指定工程目录；
2. 修改项目配置；
3. 初始化SDK；
4. 根据用户登录信息调用SDK接口函数；
5. 验证SDK接入是否正确；
6. 可选择性的接入相关功能性接口；
7. 单独付费功能接入前需与易盾联系。
8. 本sdk最低支持的系统为Android 4.4

## 二、接入步骤

### 1、导入sdk

以Android Studio为例，将获取到的sdk的aar文件放到工程中的libs文件夹下，然后在app的build.gradle文件中增加如下代码

```
repositories {  
    flatDir {  
        dirs 'libs'  
    }  
}
```

在build.gradle配置文件的 dependencies依赖中增加对aar包的引用（x.x.x表示版本号，请联系技术支持确认最新的版本号）：

```
dependencies {  
    implementation (name:'HTProtectLib-x.x.x', ext:'aar')  
}
```

### 2、过滤需要的ABI

SDK提供了armeabi、armeabi-v7a、x86、arm64-v8a四种ABI的支持，默认会导出这四种ABI。

注意：

- 如果产品本身不支持这么多ABI，就需要对最终导出的ABI进行过滤，不然会CRASH
- 如果app只需要支持特定的ABI，比如armeabi，armeabi-v7a，x86三种，可以在build.gradle添加如下配置：

```
defaultConfig {
    applicationId "com.XX.XXX"
    minSdkVersion XX
    targetSdkVersion XX
    versionCode XX
    versionName "X.X.X"
    ndk {
        abiFilters "armeabi", "armeabi-v7a", "x86"
    }
}
```

### 3、添加权限信息

SDK不会申请任何权限，但是为了提升反外挂的效果，建议在AndroidManifest.xml文件中添加下列权限配置：

```
<!--网络通信-->
<uses-permission android:name="android.permission.INTERNET"/>
<!--获取设备信息-->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!--获取MAC地址-->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<!--获取网络状态-->
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

### 4、添加ProGuard配置

若使用ProGuard进行混淆，需要将SDK使用的类排除掉。若使用Android studio开发，则在proguard-rules.pro文件里添加如下信息：

```
-keep class com.netease.htprotect.**{*;}
-keep class com.netease.mobsec.**{*};
```

## 三、SDK接入调用说明

### 1、SDK初始化

**接口用途：**

用于初始化反外挂SDK。

**接入须知：**

使用其他接口之前，必须先调用初始化接口，建议在游戏启动后第一时间调用。

**注意：**

该接口为**必须调用接口**。

**函数原型：**

```
public static void init(Context context,String appId,String gameKey,int
serverType);
```

#### 参数说明：

参数	说明
context	当前环境的上下文
appId	易盾HTP分配appId
gameKey	易盾HTP分配gameKey，与手游加固的appKey对应，gameKey请联系易盾获取
serverType	默认值为1；中国大陆地区请填写1；中国台湾地区请填写2；其他地区请填写3；欧盟地区请填写4

如需要示例demo请联系易盾获取。

#### 示例代码：

```
....
import com.netease.htprotect.HTProtect;           // 调入sdk的接口类
....

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ....
        Context mContext = getApplicationContext();
        HTProtect.init(mContext, "Axxxxxxx", "xxxxxxxxxxxxxxxx", 1); //调用sdk初始化
接口init函数
        ....
    }

}
```

## 2、登录接口/设置角色信息

#### 接口用途：

在进行数据采集的过程中，会将角色ID、角色名称、角色账号等设置在反外挂采集的数据中一同上传，标识了用户的信息后，对应有恶意行为的用户可以进行相应的惩罚。

#### 接入须知：

须在init接口被调用后，且用户同意隐私政策后才能调用该接口，涉及到玩家登陆或者切换角色的地方，均需要调用该接口设置或者更新角色信息。

#### 注意：

该接口为反外挂业务**必须调用接口**，且用户同意隐私政策后才能调用该接口，[网易易盾隐私说明](#)。

函数原型：

```
public static void setRoleInfo(String roleId, String roleName, String
roleAccount, String roleServer, int serverId,String gameJson);
```

参数说明：

参数	说明	赋值
roleId	玩家的角色ID	必填
roleName	玩家的角色名称	可填空
roleAccount	玩家的角色账号	可填空
roleServer	玩家的角色服务器	可填空
serverId	所属服务器的id	可填空
gameJson	游戏需要上传的信息，对应一个json字符串，包含key	可填空

gameJson字段说明：

KEY 名称	KEY 类型	KEY值	是否 必须	官网 标识	说明
游戏 版本号	字 符 串	GameVersion	可 选	游 戏 版 本	
资源 版本号	字 符 串	AssetVersion	可 选	资 源 文 件 版 本	
数据 转发 域名	字 符 串	TransHost	可 选		易盾网络不通时，反外挂数据转发域名地址，不需要设置“http://”或者“https://”前缀，请参照示例代码
数据 转发 IP	字 符 串	TransIP	可 选		填空
数据 转发 端口	int	TransPort	可 选		易盾网络不通时，反外挂数据转发端口

数据转发域名/端口:

设置之后, 当数据发送到易盾后端失败后, 会继续向游戏服务端发送数据, 由游戏方服务端转发到易盾后端. 域名可以选择设置一个或者都设置, 端口默认使用80. 此字段设置前需要和易盾技术人员确定. 尽量默认接入的时候带上, 之后有需要再搭建服务端转发, 数据转发只会在发送到易盾服务端失败的情况下会走转发流程。

**示例代码:**

```
JSONObject object = new JSONObject();
object.put("GameVersion", "1.0.1");
object.put("AssetVersion", "1.0.1");
object.put("TransHost", "test.163.com");
object.put("TransIP", "8.8.8.8");
object.put("TransPort", 80);
HTProtect.setRoleInfo("123456", "易小盾", "yd@163.com", "游戏测试服", 123, object.toString());
```

### 3、作弊监控（单独付费功能）

作弊监控用于注册、登录保护、营销反作弊等场景, 与反外挂是相互独立业务, 可单独接入。

**接口用途:**

用于可能被批量利用的业务场景里, 比如登录、注册、领取奖励、抽卡等情况, 业务方可通过该功能, 判断当前玩家实时状态. 业务客户端可通过该接口获取token, 并将该token发送到业务方服务端, 然后由业务方服务端向易盾服务端发起check请求, 获取玩家当前检测结果, 业务方服务端根据该结果做出相应操作。

**接入须知:**

通过该接口获取唯一凭证token, 业务方客户端需要将此token提交到业务后端, 业务后端再使用此token来请求check接口获取检测结果, [作弊监控（服务端）接入文档](#)。

**注意:**

该接口为同步接口, 会有一定的耗时, 默认超时时间为3s; 须在[SDK初始化和登录接口/设置角色信息](#)被调用后, 且用户同意隐私政策后才能调用该接口, [网易易盾隐私说明](#)。

若注册保护场景, 当前无角色信息, 可设置“roleId”为“0”, [登录接口/设置角色信息](#)调用的目的是告知sdk已同意隐私政策, 可获取mac等隐私信息。

**示例代码:**

```
String acToken = HTProtect.getToken();
```

**返回值:**

正常情况下, getToken接口是一定会有返回值, 只有一种情况会返回空, init接口调用失败; init接口失败目前只有一种情况: 未开通作弊监控服务。

### 4、模拟点击行为检测（单独付费功能）

**接口用途：**

批量的模拟点击极大的影响到游戏的正常运营，特别是工作室的参与，不仅影响到游戏的平台，还影响到游戏方的收入。易盾智能反外挂SDK提供相关的行为检测方案，使用模拟点击行为检测接口时，需要在游戏登录后，并且调用[登录接口/设置角色信息](#)，设置role\_id等信息后调用。

**接入须知：**

非强烈需要挖掘模拟点击类外挂的游戏，不需要接入。

请在作弊情况较严重的玩法和场景下调用registerTouchEvent和unregisterTouchEvent，开启和关闭点击数据的采集，初次接入建议设置一个玩法和场景id，并同步给易盾，如需接入多个玩法和场景需提前与易盾沟通。

使用该功能的时候，若接入的sdk不是aar包，需要将sdk内提供的assets目录下文件全部拷贝到apk的assets目录下。

**(1) 开启采集数据**

**函数原型：**

```
public static void registerTouchEvent(int gameId,int sceneId)
```

**函数说明：**

调用该接口后，开始采集点击事件的相关数据。

**参数说明：**

参数	说明
gameplayId	玩法id
sceneId	场景id

**示例代码：**

```
HTProtect.registerTouchEvent(123,456);
```

**(2) 关闭数据采集**

**函数原型：**

```
public static void unregisterTouchEvent()
```

**函数说明：**

取消点击事件数据采集。

**示例代码：**

```
HTProtect.unregisterTouchEvent();
```

## 5、通用查询

### 接口用途：

反外挂SDK提供root、模拟器、反外挂版本等信息查询服务，以及其他定制功能。

### 接入须知：

必须在setRoleInfo之后才可调用该接口

### 函数原型：

```
public static String ioctl(int request, String data);
```

### 参数说明：

参数	说明
request	对应向反作弊模块请求的命令
data	对应设置信息的附加数据，可为空

request值：

```
public enum RequestCmdID
{
    Cmd_GetEmulatorName = 1,
    Cmd_IsRootDevice = 2,
    Cmd_DeviceID = 3,
    Cmd_GetHTPVersion = 7,
    Cmd_GetCollectData = 8,
    Cmd_GetEnchTPVersion = 9
};
```

### 返回值：

错误返回 unsupported request （不支持的命令），正确返回对应命令的结果

### (1) 模拟器检测

向反外挂模块查询模拟器信息。无附加数据，返回值模拟器名称

### 示例代码：

```
String name = HTProtect.ioctl(RequestCmdID.Cmd_GetEmulatorName, "");
```

### 返回值：

对应查询的模拟器信息。若为模拟器则返回模拟器名称，示例 emulator:Nox，若为未知模拟器，则返回 emulator:Unknown。若为真机，则返回 emulator:None

## (2) root检测

向反外挂模块查询设备是否root。

示例代码：

```
String name = HTProtect.ioctl(RequestCmdID.Cmd_IsRootDevice, "");
```

返回值：

若为root设备，返回值为 `root:1`，若不是root设备，返回值为 `root:0`。

## (3) 反外挂版本号获取

向反外挂模块查询当前使用的智能反外挂版本号，掌握每个玩家的反外挂版本号。

示例代码：

```
String version = HTProtect.ioctl(RequestCmdID.Cmd_GetHTPVersion, "");
```

返回值：

版本号一定不为空，返回值举例："4.15.114"。

若要获取加密后的版本号，可以传入参数“Cmd\_GetEncHTPVersion”

示例代码：

```
String version = HTProtect.ioctl(RequestCmdID.Cmd_GetEncHTPVersion, "");
```

返回值：

版本号一定不为空，返回值举例："iF13uXaKVteXVA==|58807f98da0d945b5d60ba4e8f22a428"。

返回值需要使用易盾提供的服务端解密sdk进行解密。

## 6、心跳系统

接口用途：

保障反外挂服务的安全，防止易盾反外挂服务被中止或者被剥离的风险，定时向游戏方反馈心跳信息，告知游戏方当前反外挂运行状态，既方便游戏方实时掌握反外挂运行状态，也能保障易盾反外挂服务的正常运行。

同时也可通过心跳实时返回反外挂数据，如需要详细说明文档，请联系易盾获取。

接入须知：

- `registInfoReceiver`接口需要在init之后方可调用
- SDK会启动线程来调用onReceive函数，请注意多线程问题

函数原型：

```
public static void registInfoReceiver(InfoReceiver receiver);
```



参数说明：

参数	类型	说明
receiver	InfoReceiver	回调函数，接收心跳信息

InfoReceiver的接口定义：

```
public interface InfoReceiver{
    void onReceive(int Type, string Info);
}
```

onReceive参数说明：

参数	类型	说明
Type	int	接收数据的类型
Info	string	接收的具体数据

目前支持的type类型有：

值	定义	说明
1	NetHeartBeat.INFO_TYPE_HEARTBEAT	心跳数据，可直接在客户端解析
2	NetHeartBeat.INFO_TYPE_ENC_HEARTBEAT	加密后的心跳数据，必须发往服务端进行解密
3	NetHeartBeat.INFO_TYPE_CHEATINFO	检测信息，包括app基本信息以及检测到的异常信息

示例代码：

- 实现接口

```
....
import com.netease.htprotect.HTProtect;           // 调入sdk的接口类
....

private class MyInfoReceiver implements NetHeartBeat.InfoReceiver {
    public void onReceive(int type, String info) {
        if (type == NetHeartBeat.INFO_TYPE_HEARTBEAT) {
            Log.e(TAG,"info is:" + info);
            //（获取到info进行相关操作）
        }
        if (type == NetHeartBeat.INFO_TYPE_ENC_HEARTBEAT) {
            Log.e(TAG,"info is:" + info);
            //（获取到的info，为加密状态，需发往服务端进行解密）
        }
        if (type == NetHeartBeat.INFO_TYPE_CHEATINFO) {
            Log.e(TAG,"info is:" + info);
        }
    }
}
```

```

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ....
        Context mContext = getApplicationContext();
        HTTPProtect.init(mContext, "xxxxxxxxxxxxxxxx");
        ....
        MyInfoReceiver mInfoReceiver = new MyInfoReceiver();
        HTTPProtect.registInfoReceiver(mInfoReceiver); // 注册接口
        ....
    }
}

```

注：两种type类型均可接收，也可只接收其中一种，建议只获取加密后的心跳数据，然后发往服务端进行解析。

心跳系统是每10s内会返回一次数据("Info")，返回的数据为一个字符串，“Info”包含三种信息，分别是序列号、时间戳、网络状态，使用“key:value”格式并使用“||”分割，含义如下表所示：

名称	标示	说明	异常情况	是否每次必有
序列号	seq	从初始值“1”开始递增	序列号不存在，或者乱序	是
时间戳	t	当前时间戳，以秒为单位	时间戳非当前时间	是
网络标识	net	数据发送是否成功 (1 为成功，0 为失败)	网络异常，数据发送失败，则网络标识为0	是

Info的返回值示例：

- 易盾网络通信正常时：

```
seq:1||t:1589781840||net:1
```

- 易盾网络通信异常时：

```
seq:1||t:1589781840||net:0
```

处理建议：

- seq、t和net是否出现，该三个标识为心跳系统一定会返回的值，若无，说明反外挂异常
- seq的值是否符合“从1开始依次递增”的规则，若不符合，说明返回值被篡改
- net为0是否长期出现，若5分钟内net的取值都为0，说明网络异常
- 若出现反外挂异常或网络异常，请结合游戏业务数据综合判断并自行处理

## 7、数据校验

### 接口用途：

游戏协议如被破解，若为单机游戏，可直接影响到存档文件的安全性；若为帧同步游戏，则可通过破解协议修改战斗数据；其他类型游戏亦可被做成脱机协议，用于攻击或脱机外挂。为保障游戏协议安全，易盾提供“数据校验”方案，采用自研通讯协议校验算法。

### 接入须知：

客户端接入数据校验接口后，服务端对对应的数据进行校验，服务端数据校验相关文档请联系易盾获取。

### 函数原型：

```
public static String getDataSign(String inputData,int algIndex);
```

### 参数说明：

参数	类型	说明
inputData	String	待校验的数据
algIndex	int	表示算法标识，需要联系易盾人员

### 返回值：

返回的数据即为校验结果，开发者需要将原始数据与校验结果发送到游戏服务端，由游戏服务端校验。

## 8、安全通信协议

对通信数据进行白盒加密和白盒HMAC签名，加密和签名密钥均被隐藏，攻击者无法获取，保障通信安全。

### 接入须知：

由于采用白盒算法，会存在一个白盒查找表(一个文件)来代替密钥，攻击者无法从白盒查找表中恢复出原始密钥，保障客户端加解密安全。此白盒查找表由易盾生成。 如果需要用到此接口，请联系易盾获取白盒查找表(需放在assets目录下)和服务端相关文档。

### 函数原型：

```
String safeComm(String input, int algType, boolean dec);
```

### 参数说明：

参数	类型	说明
input	String	待加密的数据
algType	int	表示算法标识，目前传 0
dec	boolean	true表示解密服务端的数据, false表示加密数据到服务端

### 返回值：

当dec为false时，返回的数据即为加密和签名结果，开发者需要将此数据与校验结果发送到游戏服务端，由游戏服务端解密和校验，加密失败返回空字符串""

当dec为true时，返回的数据为服务端数据解密后的结果，解密失败返回空字符串""

## 9、存档加解密

### 接口用途：

对需要本地存储的数据提供加解密功能

### 接入须知：

加密后的数据，需要接入方自行存储，sdk只提供加密和解密功能

### 函数原型：

加密对象为String：

```
String localSaveEncode(String inputData,int algIndex);
String localSaveDecode(String inputData,int algIndex);
```

加密对象为byte[]：

```
String localSaveBytesEncode (byte[]inputData,int algIndex);
byte[] localSaveBytesDecode (String inputData,int algIndex);
```

### 参数说明：

参数	说明
inputData	需要加密或者解密的数据
algIndex	参数为0：跟app安装绑定；参数为1：跟app签名绑定

### 返回值：

若是加密接口，返回的数据即为原始数据的密文。

若是解密接口，返回的数据即为密文对应的原始数据。

## 10、透传服务

随着外挂和恶意环境对抗的不断升级，部分外挂玩家使用工具屏蔽网易易盾域名和ip，阻断反外挂策略更新，影响反外挂效果，特别是云真机玩家，部分云真机自动屏蔽网易易盾域名。若游戏方需要检测云真机环境或者外挂对抗较为激烈建议接入透传服务。透传服务有两种接入方式，分别是**反外挂SDK自动化透传**和**游戏方自助式透传**，可根据业务需求选择接入方式。

强烈建议接入**游戏方自助式透传**，更详实的服务端接入文档请联系易盾。

### (1) 反外挂SDK自动化透传（透传1.0）

**优点：**接入相对简单；只有网络不通时候才触发，数据量小。

**缺点：**只支持http转发，安全强度不够，易屏蔽。

#### 接入须知：

客户端接入方式：在[登录接口/设置角色信息](#)中设置相应的透传参数，其中域名是必填项，在易盾网络被屏蔽和网络不通的情况下，将走游戏设置的透传域名进行网络数据传输

服务端接入方式：[易盾反外挂SDK自动化透传](#)。

### (2) 游戏方自助式透传（透传2.0）

**优点：**安全强度高，隐蔽性高。

**缺点：**只支持嫌疑数据上报，不支持反外挂策略更新等；每次调用该接口都会获取数据，数据量大。

#### 接入须知：

客户端接入方式：在玩家登录游戏后，在[通用查询](#)中，设置RequestCmdID为Cmd\_GetCollectData，获取反外挂嫌疑数据，然后（可与游戏数据一起）上报到游戏服务器，游戏服务端再将反外挂嫌疑数据发送到易盾服务器。

服务端接入方式：[游戏方自助式透传](#)。