

宙斯盾安全防护

SDK 文档

产品文档



腾讯云

【版权声明】

©2013-2020 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

SDK 文档
 水印 SDK

SDK 文档

水印 SDK

最近更新时间：2020-04-14 15:41:33

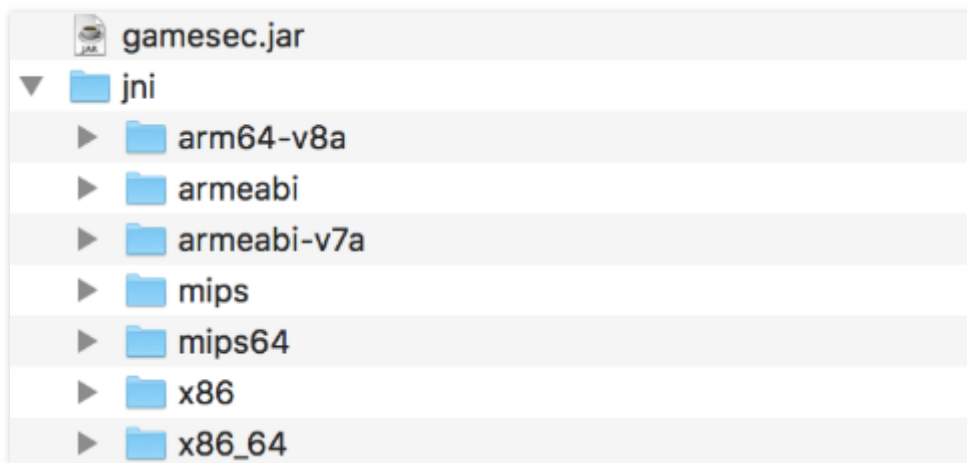
SDK 准备

下载相关 [Demo](#) 及 [SDK](#)。本文主要包括 Android、iOS 和 Windows 三个版本的接入指南。

Android 接入

预备工作

- 接入 SDK 需要完成以下步骤：
 - 根据运行平台选择相应的 so 文件，将 so 文件和 jar 文件拷贝到工程目录下并添加依赖。
 - 调用 SDK 接口函数，生成水印信息。
 - 发送报文时，将20字节水印信息放在消息体前面。
- SDK 文件包含 so 文件和 jar 文件，目录结构如下：

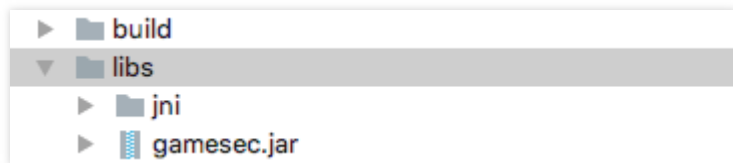


- SDK API 说明：
 - 程序包：com.gamesec
 - 类：Mark
- 接口说明：

接口名称	说明
CreateSDKBuffFromStr	生成水印

接入步骤 (Android Studio)

1. 将 sdk/android 文件夹下的内容拷贝到工程目录的 libs 文件夹下：



2. 修改项目的 build.gradle 文件，设置 jni 文件目录，添加 jar 依赖：

```
android {
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs/jni'] // 设置 jni 目录
        }
    }
}
dependencies {
    implementation files('libs/gamesec.jar') // 添加依赖
}
```

3. Eclipse 接入方法类似，不需要配置 build.gradle 文件。

接口调用

1. 导入程序包。

```
import com.gamesec.*;
```

2. 实例化 Mark 对象。

```
Mark mark = new Mark();
```

3. 调用 CreateSDKBuffFromStr 生成水印。

```
byte [] CreateSDKBuffFromStr (String pSDKInfo, String buffer, String uDeslp, int uDesPort)
```

o 参数说明：

参数	类型	含义
pSDKInfo	String	水印防护密钥

buffer	String	占位参数，传入空字符串即可
uDesIp	String	服务器 IP，如 "1.2.3.4"
uDesPort	int	服务器端口

◦ 返回值：

类型	含义
byte[]	计算的水印信息，取20字节

◦ 调用示例：

```
String pSDKInfo = "566c2dea9420eb37-b6c8-566c2dea9420eb3710525135e8485e80806a2f9c";
String uDesIp = "115.159.147.198";
int uDesPort = 8899;
byte[] bytes = mark.CreateSDKBuffFromStr(pSDKInfo, "", uDesIp, uDesPort);
```

4. 添加水印信息到消息体。代码示例如下：

```
Socket s = new Socket(uDesIp, uDesPort);
OutputStream out = s.getOutputStream();
PrintWriter output = new PrintWriter(out, true);
// 先传入水印信息
output.print(bytes);
output.println("msg msg msg");
BufferedReader input = new BufferedReader(new InputStreamReader(s.getInputStream()));
String msg = input.readLine();
s.close();
```

iOS 接入

预备工作

- 接入 SDK 需要完成以下步骤：
 - 将 SDK 文件拷贝到工程目录，Swift 工程需要添加桥文件。
 - 调用 SDK 接口函数，生成水印信息。
 - 发送报文时，将 20 字节水印信息放在消息体前面。

- SDK 文件包含 a 文件和 h 文件，目录结构如下：

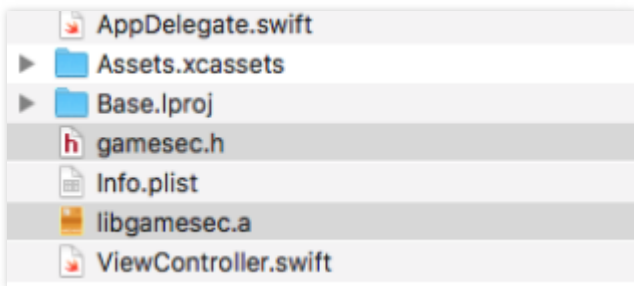


- 接口说明：<https://main.qcloudimg.com/raw/4ecce64e8d49226ad683a594f4aa3cf2.png>

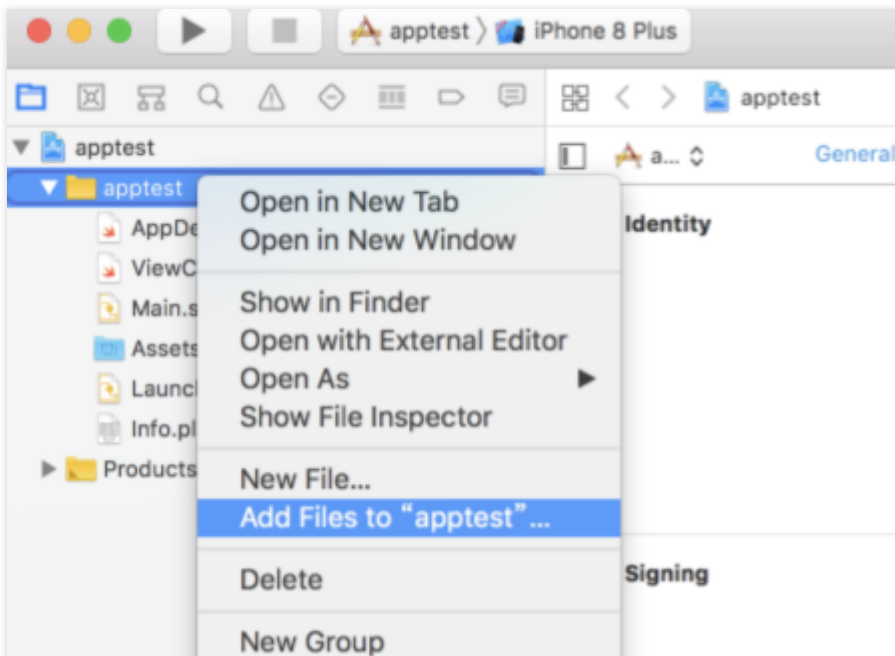
接口名称	说明
CreateSDKBuffFromStr	生成水印

接入步骤 (Xcode)

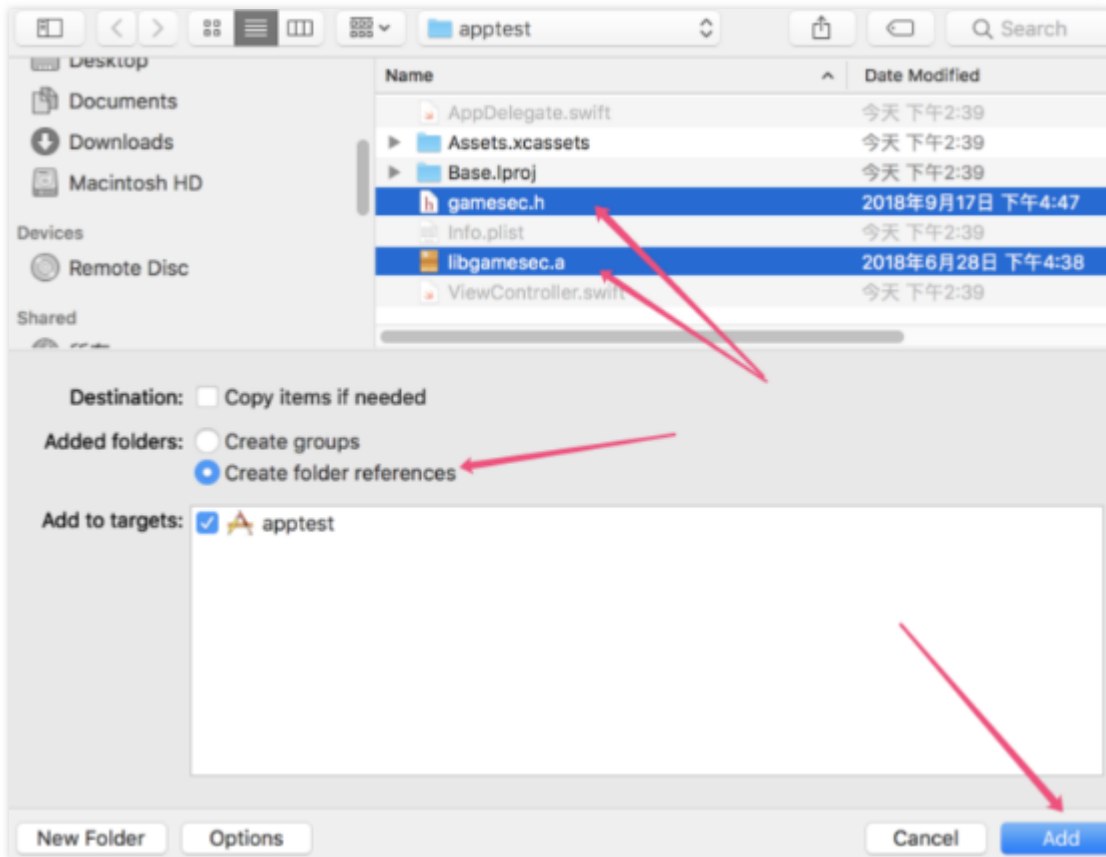
- 将 sdk/ios 文件夹下的内容拷贝到工程目录：



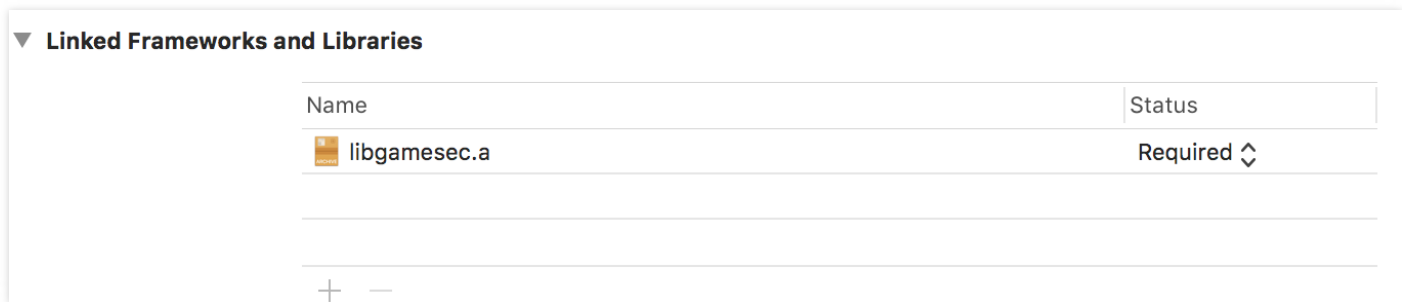
- 将 SDK 文件添加到 Xcode。右键工程名，单击“Add Files to”。



3. 在对话框中勾选“Create folder references”，选中 SDK 的两个文件，单击 Add。



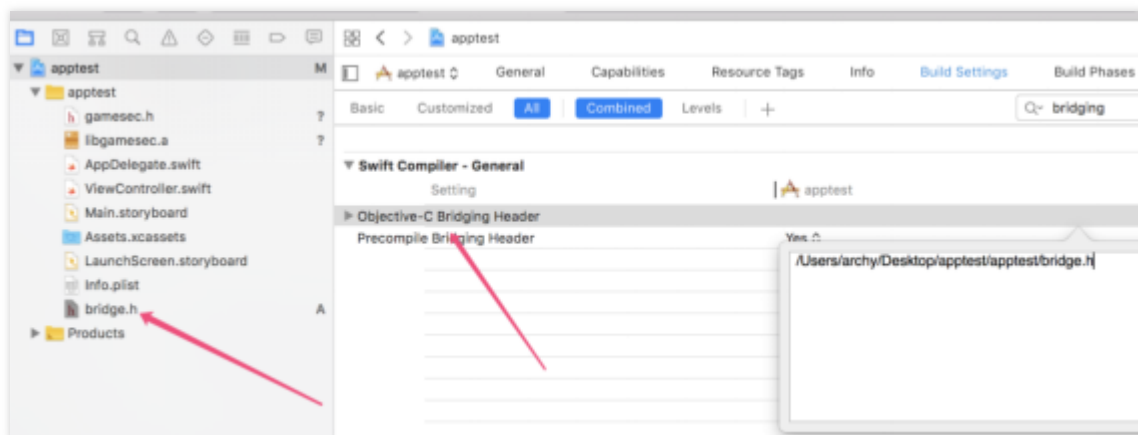
4. 左键工程名，选择 General，将 a 文件添加到“Linked Frameworks and Libraries”：



5. 如果是 Swift 项目，需要创建桥文件，Object-C 项目可以跳过此步骤。创建一个 Header File，命名为 bridge.h。并在文件中添加以下代码：

```
# import "gamesec.h";
```


6. 左键工程名，选择 Build Settings，将 bridge.h 添加到 Objective-C Bridging Header 中：



接口调用

1. Swift 项目可以直接调用生成水印函数，Objective-C 项目需要在使用的文件里面添加头文件：

```
# import "gamesec.h";
```

2. 调用 CreateSDKBuffFromStr 生成水印。

```
uint32_t CreateSDKBuffFromStr(char *pSDKinfo, uint8_t *buffer, char* uDstIp, uint16_t uDstPort);
```

参数说明：

参数	类型	含义
pSDKinfo	char *	水印防护密钥
buffer	uint8_t *	水印指针，输出水印结果
uDesIp	char *	服务器 IP，如 "1.2.3.4"
uDesPort	uint16_t	服务器端口

注意：

水印结果保存在参数 buffer 中，取20字节。

3. 调用示例。

Swift 调用：

```
let pSDKInfo = UnsafeMutablePointer<Int8>(mutating: (
    "566c2dea9420eb37-b6c8-566c2dea9420eb3710525135e8485e80806a2f9c"
    as NSString).utf8String);
var buffer = UnsafeMutablePointer<UInt8>.allocate(capacity: 20);
let uDstIp = UnsafeMutablePointer<Int8>(mutating: (
    "115.159.147.198" as NSString).utf8String);
let uDstport = UInt16.init("8899")!;
```

```
CreateSDKBuffFromStr(pSDKInfo, buffer, uDstIp, uDstport);
```

```
for i in 0 ..< 20 {
    let b = (buffer+i).pointee;
    // 水印信息在前20字节，注意这里输出的是 uint8
    print(" \b");
}
**Object-C 调用：**
```

```
char *pSDKInfo = "566c2dea9420eb37-b6c8-566c2dea9420eb3710525135e8485e80806a2f9c";
uint8_t buffer[20];
char *uDstIp = "115.159.147.198";
uint16_t uDstPort = 8899;
```

```
CreateSDKBuffFromStr(pSDKInfo, buffer, uDstIp, uDstPort);
```

```
for(int i=0;i<20;i++)
{
    // 水印信息在前20字节
    NSLog(@"%d", (int8_t)buffer[i]);
}
```

4. 发送报文前，添加 20 字节水印信息到消息体前面。

Windows 接入

预备工作

SDK 为 gamesec.dll 文件，有一个生成水印的函数：

```
uint32_t CreateSDKBuffFromStr(char *pSDKInfo, uint8_t *buffer, char* uDstIp, uint16_t uDstPort);
```

参数说明：

参数	类型	含义
pSDKInfo	char *	水印防护密钥
buffer	uint8_t *	水印指针，输出水印结果
uDesIp	char *	服务器 IP，如 "1.2.3.4"
uDesPort	uint16_t	服务器端口

注意：

水印结果保存在参数 buffer 中，取20字节。

接口调用

在使用水印函数时，需先导入 dll 文件，可以使用 LoadLibrary 函数（需要添加 Windows.h）：

```
// 定义函数指针
typedef int(*FUNC)(char *, uint8_t *, char*, uint16_t);
// 设置 dll 路径
HINSTANCE Hint = ::LoadLibrary(L"E:\\sdk\\gamesec.dll");
FUNC CreateSDKBuffFromStr = (FUNC)GetProcAddress(Hint, "CreateSDKBuffFromStr");
```

完整调用示例：

```
// 保存水印
uint8_t buffer[BUFFER_SIZE];
memset(buffer, 0, BUFFER_SIZE);

int UDP_TEST_PORT = 8899;

const char * CONST_UDP_SERVER_IP = "115.159.147.198";
char * UDP_SERVER_IP = new char[strlen(CONST_UDP_SERVER_IP)];
strcpy(UDP_SERVER_IP, CONST_UDP_SERVER_IP);

const char * CONST_pSDKInfo =
"566c2dea9420eb37-b6c8-566c2dea9420eb3710525135e8485e80806a2f9c";
char * pSDKInfo = new char[strlen(CONST_pSDKInfo)];
strcpy(pSDKInfo, CONST_pSDKInfo);

// 调用10次
```

```
for (int i = 0; i < 5; i++) {  
    CreateSDKBuffFromStr(pSDKInfo, buffer, (char *)UDP_SERVER_IP, UDP_TEST_PORT);  
  
    for (int i = 0; i <= 20; i++)  
    {  
        // 水印在前20字节  
        printf("%d ", (int8_t)buffer[i]);  
    }  
    printf("\n\n");  
}
```