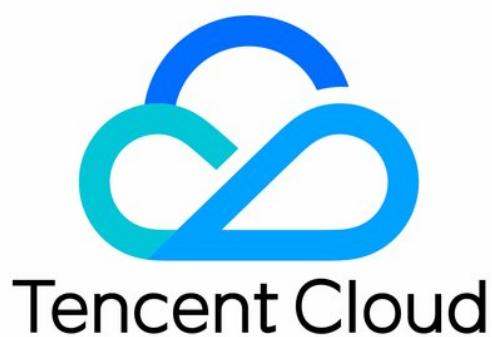


# Aegis Anti-DDoS

## SDK Documentation

## Product Documentation



## Copyright Notice

©2013-2019 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's prior written consent.

## Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

## Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

# Contents

[SDK Documentation](#)

[Watermark SDK](#)

# SDK Documentation

## Watermark SDK

Last updated : 2020-01-14 11:14:21

## SDK Preparations

Download the relevant [Demo](#) and [SDK](#). This article mainly includes access guides for Android, iOS and Windows.

## Android Access

### Preparations

- You need to complete the following steps:
  - i. Select the appropriate so file according to the operating platform, copy the so and jar files to the project directory and add dependencies.
  - ii. Call the SDK API function to generate watermark information.
  - iii. When sending a message, place the 20-byte watermark information in front of the message body.
- The SDK file contains so and jar files in the following directory structure:



- SDK API description:
  - Package: com.gamesec

- Class: Mark
- API description:

API name	Description
CreateSDKBuffFromStr	Generate the watermark

## Access Steps (Android Studio)

1. Copy the content of the sdk/android folder to the libs folder of the project directory:



2. Modify the project's build.gradle file, set the jni file directory and add jar dependencies:

```
android {  
    sourceSets {  
        main {  
            jniLibs.srcDirs =['libs/jni'] // Set the jni directory  
        }  
    }  
    dependencies {  
        implementation files('libs/gamesec.jar') // Add dependencies  
    }  
}
```

3. The access method for Eclipse is similar, but you don't need to configure the build.gradle file.

## API Call

1. Import the package.

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

2. Instantiate the Mark object.

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

3. Call CreateSDKBuffFromStr to generate the watermark.

```
byte [ ] CreateSDKBuffFromStr (String pSDKinfo, String buffer, String uDesIp, int uDesPort)
```

- Parameter description:

Parameter	Type	Meaning
pSDKinfo	String	Watermark protection key
buffer	String	Placeholder parameter; you can pass in an empty string
uDesIp	String	Server IP such as "1.2.3.4"
uDesPort	int	Server port

- Return value:

Type	Meaning
byte[]	The calculated watermark information; 20 bytes are taken

- Call example:

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

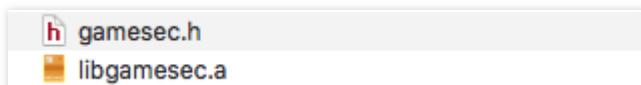
! Add watermark information to the message body. Below is an code example:

```
Socket s = new Socket(uDesIp, uDesPort);
OutputStream out = s.getOutputStream();
PrintWriter output = new PrintWriter(out, true);
// Pass in the watermark information first
output.print(bytes);
output.println("msg msg msg");
BufferedReader input = new BufferedReader(new InputStreamReader(s.getInputStream()));
String msg = input.readLine();
s.close();
```

## iOS Access

### Preparations

- You need to complete the following steps:
  - i. Copy the SDK file to the project directory. For a Swift project, you need to add the bridge file.
  - ii. Call the SDK API function to generate watermark information.
  - iii. When sending a message, place the 20-byte watermark information in front of the message body.
- The SDK file contains a and h files in the following directory structure:

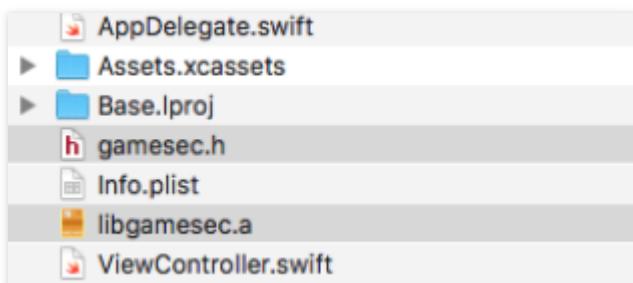


- API description:

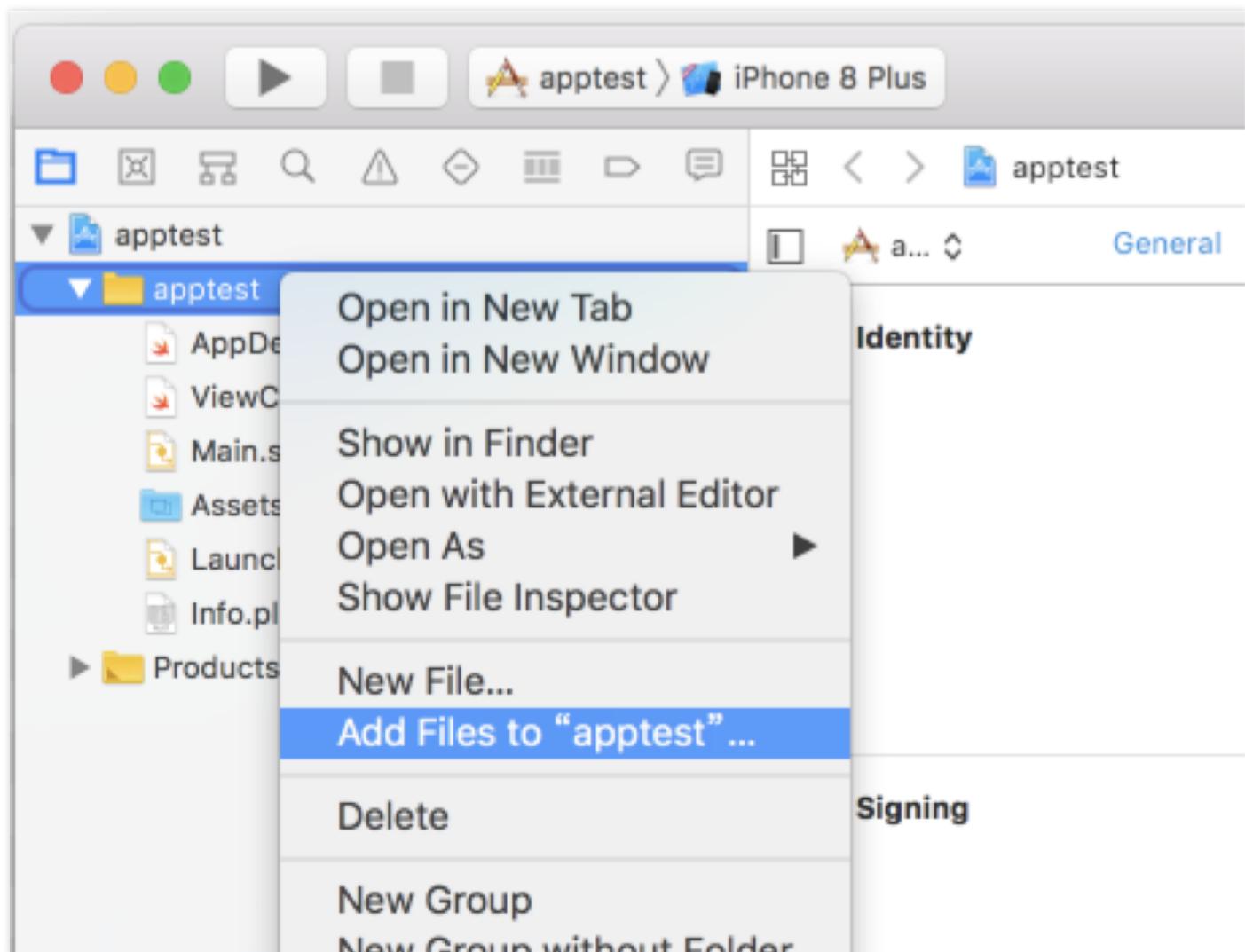
API name	Description
CreateSDKBuffFromStr	Generate the watermark

## Access Steps (Xcode)

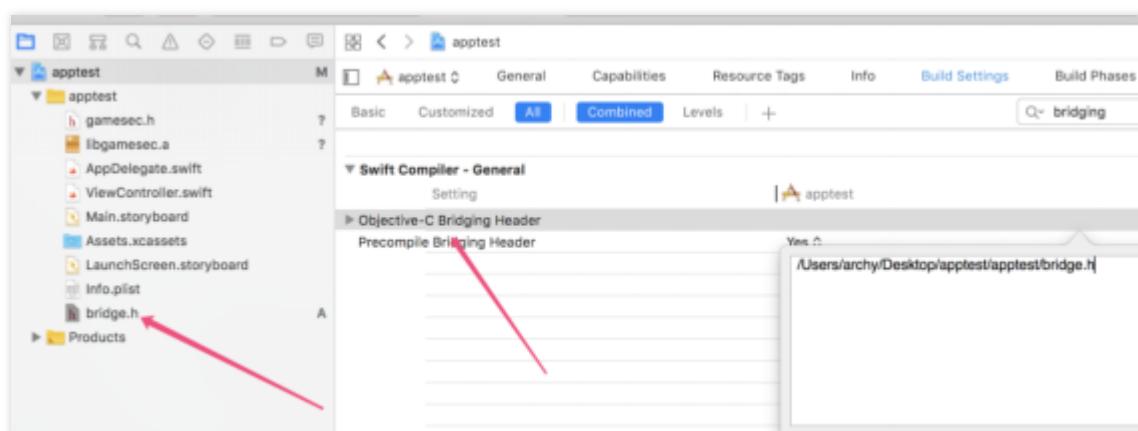
- I. Copy the content of the sdk/ios folder to the project directory:



2. Add the SDK file to Xcode. Right-click the project name and click "Add Files to".



3. Select "Create folder references" in the dialog box, select the two files of the SDK and click Add.



1. Click the project name, select General and add the a file to "Linked Frameworks and Libraries":

Name	Status
libgamesec.a	Required

2. If it is a Swift project, you need to create a bridge file; for an Object-C project, skip this step. Create a Header File and name it bridge.h. Add the following code to the file:

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

3. Click the project name, select Build Settings and add bridge.h to the Objective-C Bridging Header:

## API Call

1. For a Swift project, you can directly call the watermark generation function. For an Object-C project, you need to add the header file to the file used:

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

2. Call CreateSDKBuffFromStr to generate the watermark.

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

## Parameter description:

Parameter	Type	Meaning
-----------	------	---------

pSDKinfo	char *	Watermark protection key
buffer	uint8_t *	Watermark pointer used to output the watermark result
uDesIp	char *	Server IP such as "1.2.3.4"
uDesPort	uint16_t	Server port

**Note:**

The watermark result is stored in the parameter buffer, and 20 bytes are taken.

## 3. Call example:

## Swift call:

```

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;
// The watermark information is in the first 20 bytes. Note: The output here is uint8.
print("\(b)");
}
**Object-C call:**

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++)
{
// The watermark information is in the first 20 bytes
NSLog(@"%@", (int8_t)buffer[i]);
}

```

1. When sending a message, place the 20-byte watermark information in front of the message body.

## Windows Access

### Preparations

The SDK is the gamesec.dll file and has a watermark generation function:

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

#### Parameter description:

Parameter	Type	Meaning
pSDKinfo	char *	Watermark protection key
buffer	uint8_t *	Watermark pointer used to output the watermark result
uDesIp	char *	Server IP such as "1.2.3.4"
uDesPort	uint16_t	Server port

#### Note:

The watermark result is stored in the parameter buffer, and 20 bytes are taken.

### API Call

When using the watermark function, import the dll file first, then you can use the LoadLibrary function (you need to add Windows.h):

```
// Define the function pointer  
typedef int(*FUNC)(char *, uint8_t *, char*, uint16_t );  
// Set the dll path  
HINSTANCE Hint = ::LoadLibrary(L"E:\\sdk\\gamesec.dll");  
FUNC CreateSDKBuffFromStr = (FUNC)GetProcAddress(Hint, "CreateSDKBuffFromStr");
```

Complete call example:

```
// Save the watermark  
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);

// Call 10 times
for (int i = 0; i < 5; i++) {
CreateSDKBuffFromStr(pSDKinfo, buffer, (char *)UDP_SERVER_IP, UDP_TEST_PORT);

for (int i = 0; i <= 20; i++)
{
// The watermark is in the first 20 bytes
printf("%d ", (int8_t)buffer[i]);
}
printf("\n\n");
}
```