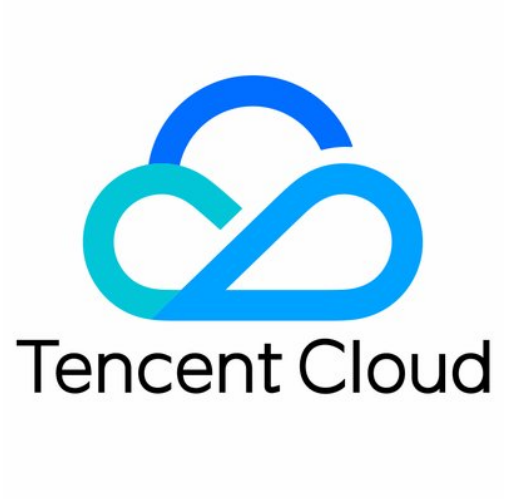
Game Multimedia Engine

Basic Feature Development Guide

Product Documentation



Contents

Basic Feature Development Guide
Authentication Key

Last updated：2024-01-18 11:56:22

This document describes the authentication key for all platforms to help you integrate and debug GME.

Backend Deployment of Voice Key

GME provides authentication keys for voice chat and offline voice. This document describes the backend deployment scheme.
The generation process of the signature used for authentication involves **plaintext**, **secret key** and **algorithm**.

Plaintext

The plaintext is the concatenation of the following fields in endian byte order:

Field	Type/Length	Description
cVer	unsigned char(1)	Version number. Enter `1` .
wOpenIDLen	unsigned short(2)	User account length
strOpenID	string	User account's characters
dwSdkAppid	unsigned short(4)	`SDKappid` of the developer
dwReserved1	unsigned int(4)	Enter `0` .
dwExpTime	unsigned int(4)	Expiration time (current time + validity period) in seconds. `300` is recommended.
dwReserved2	unsigned int(4)	Enter `-1` or `0xFFFFFFFF` .
dwReserved3	unsigned int(4)	Enter `0` .
wRoomIDLen	unsigned short(2)	Length of the ID of the room to enter. Enter `0` for the voice messaging service.
strRoomID	string	Characters of the ID of the room to enter

Key

Get the relevant permission key in the GME console.

Algorithm

The Tiny Encryption Algorithm (TEA) symmetric algorithm is used.
Generally, we recommend that you use the client deployment scheme in the initial stage, which later can be optimized for deployment on the game application's backend.

Scheme	Pros	Cons
Backend deployment	High security	Backend development and joint testing required
Client deployment	Quick integration	Low security

Backend Deployment

The encrypted string generated on the backend is sent to the client and used for the following scenario: When the EnterRoom API is called for entering a room, the encrypted string will be transferred to the `authBuffer` field in the parameters for room entering.

Algorithm Encryption Details

Key: Authentication key of `APPID` .
Encryption algorithm: TEA.
Note:
Change of the key in the console takes effect within 15 minutes to 1 hour. We recommend that you not change it frequently.

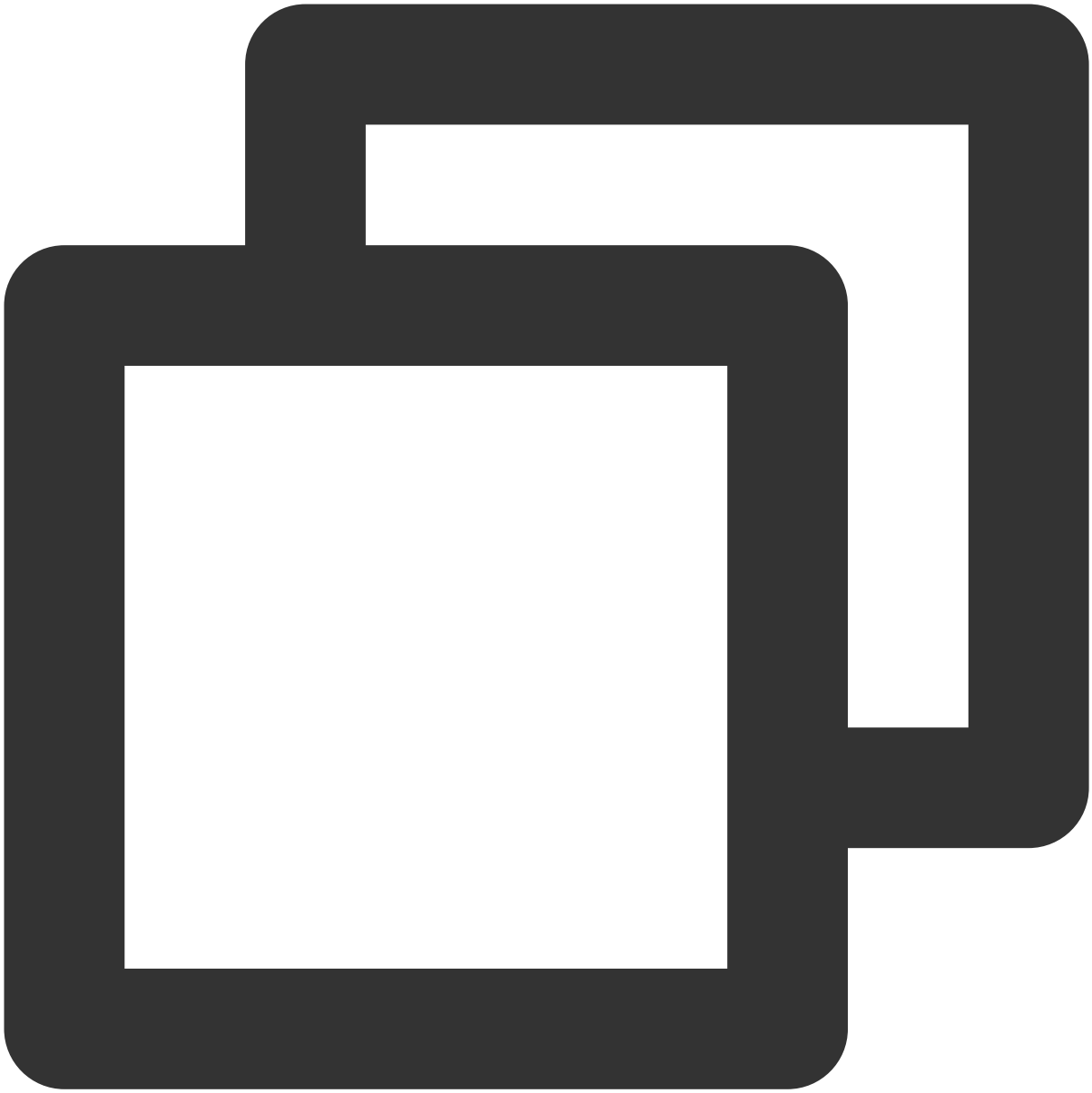
Encryption method

1. Reorganize the numbers in the plaintext in endian order.
2. Concatenate the plaintext fields into a string in the sequence how they are declared.
3. Encrypt the concatenated string with TEA. The string output by the `symmetry_encrypt` function is the permission encryption string.

Note:
Do not convert a binary string into a hexadecimal one.

Sample code

Taking C++ as an example, below is the sample code of the authentication key:



```
unsigned char pInBuf[512]={0};
    xel::byte_writer bw(pInBuf, sizeof(pInBuf));

    char cVer = 1;
    unsigned short wOpenIDLen = (unsigned short)strlen((const char *)strOpenID);
        if (wOpenIDLen > 127) wOpenIDLen = 127;
    unsigned short wRoomIDLen = (unsigned short)strlen((const char *)strRoomID);
        if (wRoomIDLen > 127) wRoomIDLen = 127;

    bw.write_byte(cVer);
    bw.write_int16(wOpenIDLen);
```

```
    bw.write_bytes(strOpenID, wOpenIDLen);
    bw.write_int32(dwSdkAppId);
    bw.write_int32(0 /*dwRoomID*/);
    bw.write_int32(expTime);
    bw.write_int32(nAuthBits);
    bw.write_int32(0 /*dwAccountType*/);
    bw.write_int16(wRoomIDLen);
    bw.write_bytes(strRoomID, wRoomIDLen);

    int pInLen = bw.bytes_write();

        unsigned char pEncryptOutBuf[512] = { 0 };
    int iEncrptyLen = 0;

        symmetry_encrypt((const unsigned char*)pInBuf, pInLen, (const unsigned char
```

You can also download the sample code in Java and Go here.

Voice Chat Role Configuration

Last updated：2023-05-19 15:13:57

Note: Currently, GME 3.x does not support voice chat role configuration.
This document describes how to access and debug GME APIs for the commanding voice mode.

Use Cases

In commanding game scenarios, GME provides the host and listener roles. If a user sets the role to host before room entry, the user can enable the mic to speak and enable the speaker to hear others talking in the room after room entry. If the user enters the room as a listener, the user cannot speak in the room after room entry even the mic is enabled.

Prerequisites

You have created a GME application and obtained the `AppID` and `Key` of the SDK as instructed in Activating Services.
You have **activated the voice chat service of GME** as instructed in Activating Services.
You have **integrated the GME SDK** as instructed in Quick Integration of Native SDK.

Integration directions

Below is the process of connecting to the commanding voice mode:
1. Use GME
2. Set the role
3. Use the voice chat service
4. Turn on the mic
5. Change the role
6. Exit the room

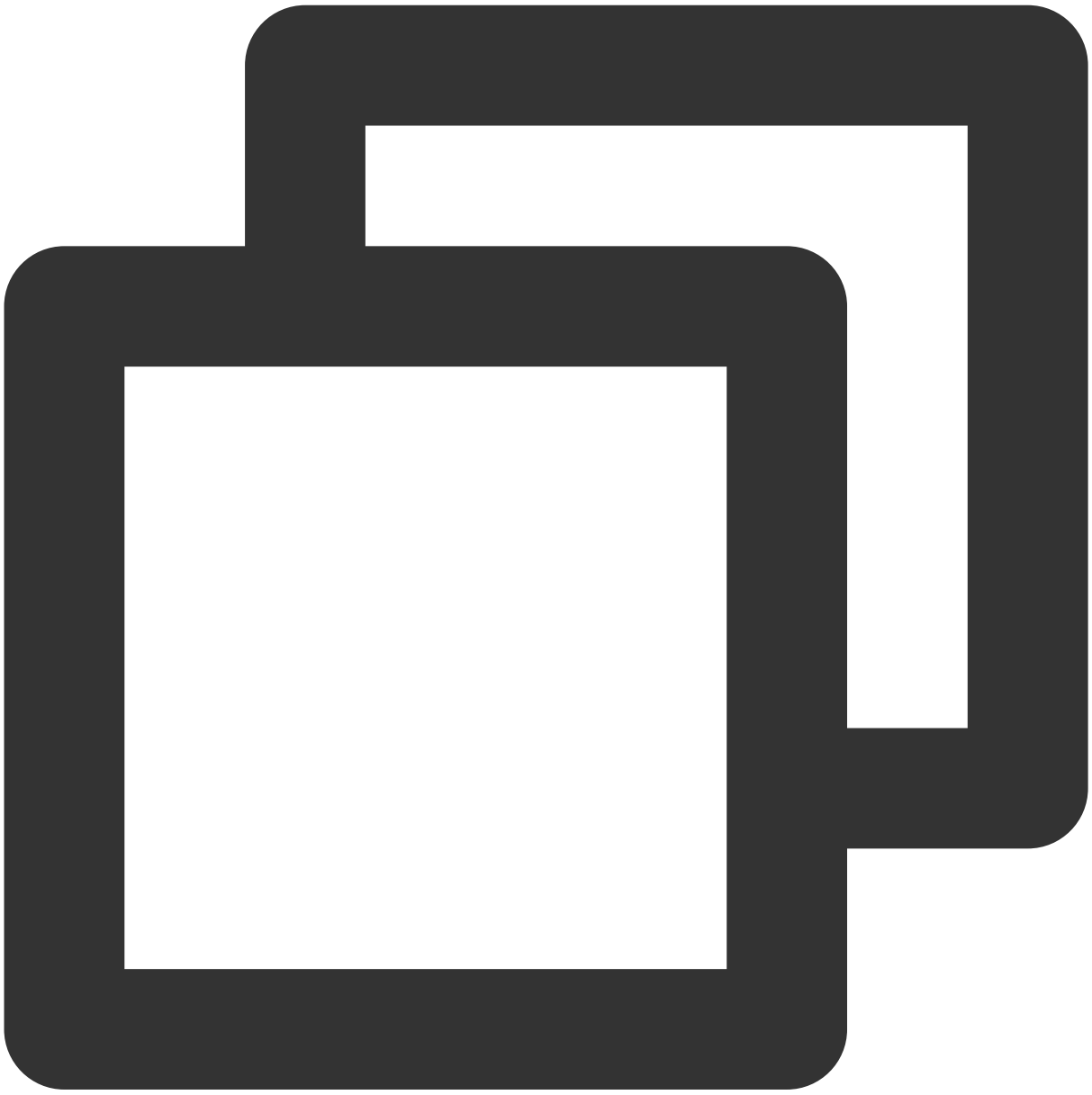
Step 1. Use GME

For more information on how to call and integrate the GME SDK, see Quick Integration of Native SDK, Quick Integration of SDK for Unity, and Quick Integration of SDK for Unreal Engine.

Step 2. Setting the role

Before calling the `EnterRoom` API, you need to call the `SetAudioRole` API to set the role of the current user in voice chat.

Function prototype

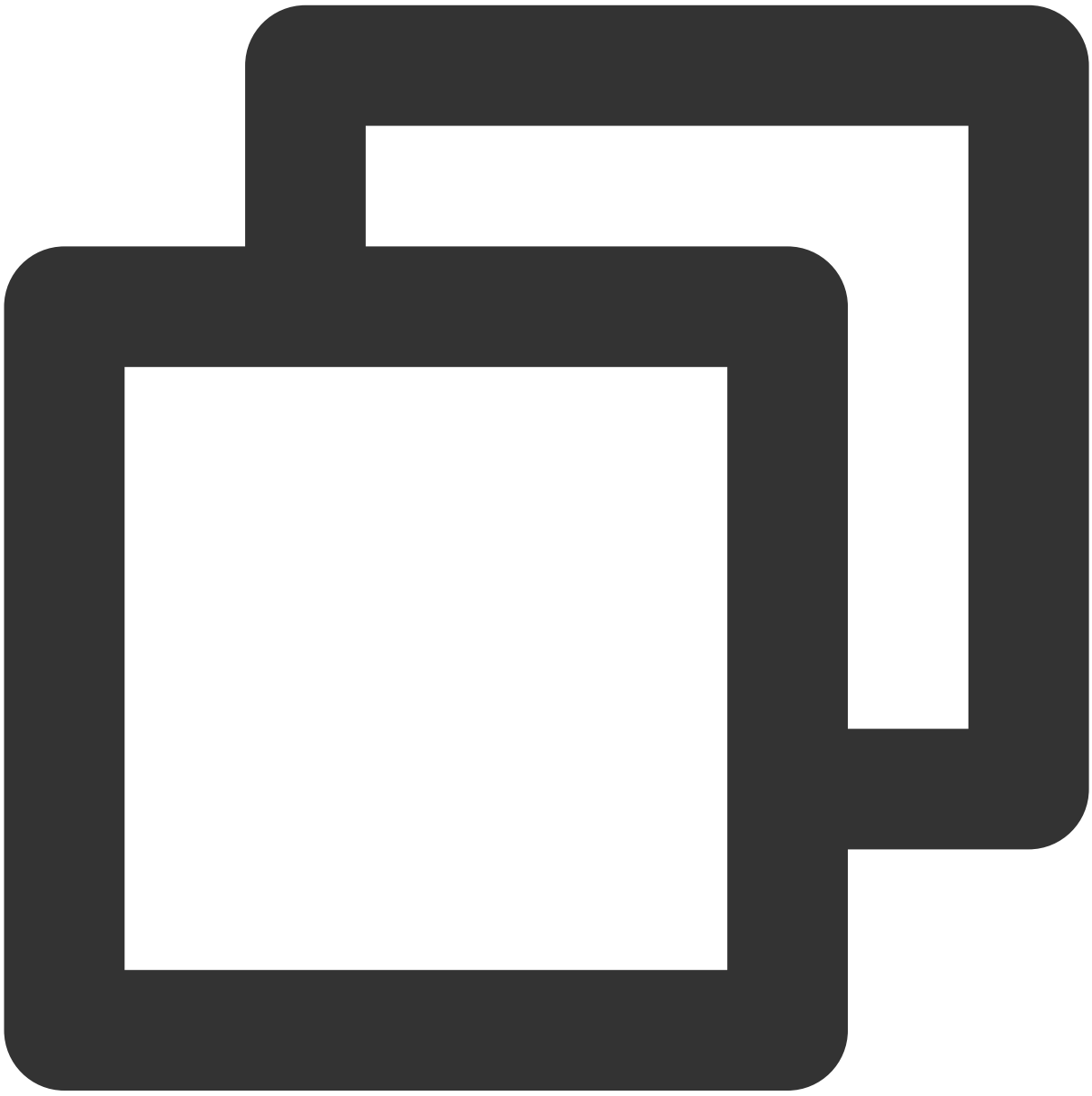


```
public abstract int SetAudioRole(ITMG_AUDIO_MEMBER_ROLE role);
```

Parameter	Type	Description
role	ITMG_AUDIO_MEMBER_ROLE	ITMG_AUDIO_MEMBER_ROLE_ANCHOR: Host, who can enable the mic and speaker in the room.

		ITMG_AUDIO_MEMBER_ROLE_AUDIENCE: Listener, who can enable only the speaker in the room to listen to others.

Sample code



```
ITMGContext.GetInstance().SetAudioRole(ITMG_AUDIO_MEMBER_ROLE.ITMG_AUDIO_MEMBER_ROL
```

Step 3. Use the real-time voice service

Call the `EnterRoom` API to enter a voice chat room.

Step 4. Enable the mic

If your role is host, you can normally call the `EnableMic` and `EnableSpeaker` APIs to enable the mic and speaker respectively.
If your role is listener, you can normally use the `EnableSpeaker` API to enable the speaker, but the `AV_ERR_INVALID_ARGUMENT` error code (1004) will be returned to remind that you are in listener mode and the mic cannot be enabled when you call the `EnableMic` API.

Step 5. Change the role

In the room, you can call `SetAudioRole` to change the role.
If the role is not set, the new role will be used.
If the role is set, the new role will be used.
If no role is set or the role is host and you are speaking with the mic enabled, but the mic is still enabled after the role is changed to listener, we recommend that you call the `EnableMic` API at the business layer to change the mic status and the mic UI status.

Step 6. Exit the room

After the `ExitRoom` API is called to exit the voice chat room, the role status becomes invalid, and the role needs to be set again.

Sound Quality

Last updated：2024-01-18 11:58:24

This document describes how to choose sound quality in the GME SDK.

Overview of Sound Quality Types

Sound Quality Type	Description	Parameter	Bitrate	Sample Rate	Volume Type	App Sce
ITMG_ROOM_TYPE_FLUENCY	Smooth	1	30 Kbps	16 kHz	Speaker: Call volume Headset: Media volume Bluetooth headset: Headset audio capturing over the HFP protocol	Smo sour qual ultra dela whic suita grou in ga like and gam
ITMG_ROOM_TYPE_STANDARD	Standard	2	64 Kbps	48 kHz	Speaker: Call volume Headset: Media volume Bluetooth headset: Headset audio capturing over the HFP protocol	Goo qual a m later whic suita voic in ca gam as Wer and gam
ITMG_ROOM_TYPE_HIGHQUALITY	HD	3	64 Kbps	48 kHz	Speaker: Media	HD qual

					volume Headset: Media volume Bluetooth headset: Mobile phone audio capturing over the A2DP protocol	a re high later whic suita mus dan gam requ sour qual as n play and kara

Note:

Currently, only the ITMG_ROOM_TYPE_FLUENCY audio quality type is provided by default. To use other audio quality types, submit a ticket.

Concepts

Media volume and call volume

Two volume modes are configured in a mobile phone: Media volume and call volume. Media volume is generally used to playback media files, and call volume is used in phone calls and communications.

For an Android mobile phone, the current volume mode is displayed on the screen when you press the volume key. As shown below, the call volume is on the left and the media volume is on the right.

Media volume and call volume

Q: What should I do if the mobile phone volume level becomes very low after room entry but becomes very high after mic-on? A: Troubleshoot as instructed in Sound and Audio.

Bluetooth headset protocols

There are two Bluetooth headset protocols with different audio performance:

Protocol	Playback Performance	Capturing Performance
HFP	The headset audio is mono only.	The headset audio can be captured.
A2DP	The headset audio is stereo and has a better audio quality.	The headset audio cannot be directly captured through this channel, and you need to use the phone or the PC mic to capture the audio.

Traffic consumption

The traffic is subject to the bitrate and number of users speaking in the room. The specific calculation formula is: bitrate * number of users / 8 = number of bytes.

Audio processing effect

The audio signal is collected by the capturing module on the mobile device, and is encoded by an audio encoder after audio preprocessing processes such as mixing cancellation, noise reduction, and automatic gain control. The methods used in the preprocessing processes, acoustic echo canceling (AEC), automatic gain control (AGC), and automatic noise suppression (ANS, also known as noise cancellation and noise suppression), are commonly known as 3A.

The difference between smooth sound quality and the other two sound qualities is ANS (noise reduction). ANS is enabled for the smooth sound quality, and is disabled for the standard and HD sound quality.