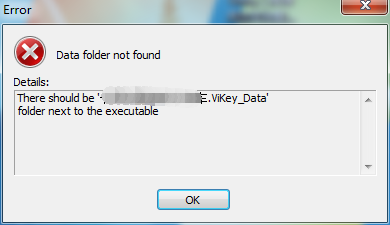
# Unity工程使用加密狗配置文档

1. **壳加密**

步骤如官方视频教程，在对xxx.exe加密后，打开xxx.ViKey.exe文件，会报错：



只需要把原配置文件夹更名为XXX.ViKey\_Data。

1. **DLL加密**

第一步：Unity3D里面创建你需要的项目，把ViKey.dll文件放到项目的Assets文件夹里面(网上有些人说建立一个专门放插件的文件夹来存放dll文件,当然，按照良好的项目路径习惯,也是合理的)。

文件分为64位和32位，请注意区分，目录：\3-[接口库文件]\Windows\动态链接库DLL\

第二步：在Unity工程新建Program.cs脚本来引入Vikey加密功能

Program.cs代码如下：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

//using System.Threading.Tasks;

using System.Runtime.InteropServices; // DllImport

namespace ViKeySample

{

class Program

{

// 错误代码

const long VIKEY\_SUCCESS = 0x00000000; //成功

const long VIKEY\_ERROR\_NO\_VIKEY = 0x80000001; //没有找到ViKey加密锁

const long VIKEY\_ERROR\_INVALID\_PASSWORD = 0x80000002; //密码错误

const long VIKEY\_ERROR\_NEED\_FIND = 0x80000003; //请先查找加密锁

const long VIKEY\_ERROR\_INVALID\_INDEX = 0x80000004; //无效的句柄

const long VIKEY\_ERROR\_INVALID\_VALUE = 0x80000005; //数值错误

const long VIKEY\_ERROR\_INVALID\_KEY = 0x80000006; //秘钥无效

const long VIKEY\_ERROR\_GET\_VALUE = 0x80000007; //读取信息错误

const long VIKEY\_ERROR\_SET\_VALUE = 0x80000008; //设置信息错误

const long VIKEY\_ERROR\_NO\_CHANCE = 0x80000009; //没有机会

const long VIKEY\_ERROR\_NO\_TAUTHORITY = 0x8000000A; //权限不足

const long VIKEY\_ERROR\_INVALID\_ADDR\_OR\_SIZE = 0x8000000B; //地址或长度错误

const long VIKEY\_ERROR\_RANDOM = 0x8000000C; //获取随机数错误

const long VIKEY\_ERROR\_SEED = 0x8000000D; //获取种子错误

const long VIKEY\_ERROR\_CONNECTION = 0x8000000E; //通信错误

const long VIKEY\_ERROR\_CALCULATE = 0x8000000F; //算法或计算错误

const long VIKEY\_ERROR\_MODULE = 0x80000010; //计数器错误

const long VIKEY\_ERROR\_GENERATE\_NEW\_PASSWORD = 0x80000011; //产生密码错误

const long VIKEY\_ERROR\_ENCRYPT\_FAILED = 0x80000012; //加密数据错误

const long VIKEY\_ERROR\_DECRYPT\_FAILED = 0x80000013; //解密数据错误

const long VIKEY\_ERROR\_ALREADY\_LOCKED = 0x80000014; //ViKey加密锁已经被锁定

const long VIKEY\_ERROR\_UNKNOWN\_COMMAND = 0x80000015; //无效的命令

const long VIKEY\_ERROR\_UNKNOWN\_ERROR = 0xFFFFFFFF; //未知错误

//ViKey加密狗类型 VikeyGetType返回值代表的类型

const uint ViKeyAPP = 0; //实用型加密狗ViKeyAPP

const uint ViKeySTD = 1; //标准型加密狗ViKeySTD

const uint ViKeyNET = 2; //网络型加密狗ViKeyNET

const uint ViKeyPRO = 3; //专业型加密狗ViKeyPRO

const uint ViKeyWEB = 4; //身份认证型加密狗ViKeyWEB

const uint ViKeyTIME = 5; //时间型加密狗ViKeyTIME

// 函数引用声明

[DllImport("ViKey")]

public static extern uint VikeyFind(ref uint pdwCount);

[DllImport("ViKey")]

public static extern uint VikeyGetHID(ushort Index, ref uint pdwHID);

[DllImport("ViKey")]

public static extern uint VikeyGetType(ushort Index, ref uint pType);

[DllImport("ViKey")]

public static extern uint VikeyUserLogin(ushort Index, Byte[] pUserPassword);

[DllImport("ViKey")]

public static extern uint VikeyAdminLogin(ushort Index, Byte[] pAdminPassword);

[DllImport("ViKey")]

public static extern uint VikeyResetPassword(ushort Index, Byte[] pNewUserPassword, Byte[] pNewAdminPassword);

[DllImport("ViKey")]

public static extern uint VikeyLogoff(ushort Index);

[DllImport("ViKey")]

public static extern uint VikeyReadData(ushort Index, ushort pStartAddress, ushort pBufferLength, Byte[] pBuffer);

[DllImport("ViKey")]

public static extern uint VikeyWriteData(ushort Index, ushort pStartAddress, ushort pBufferLength, Byte[] pBuffer);

[DllImport("ViKey")]

public static extern uint ViKeyRandom(ushort Index, ref ushort pReturn1, ref ushort pReturn2, ref ushort pReturn3, ref ushort pReturn4);

[DllImport("ViKey")]

public static extern uint VikeySeed(ushort Index, ref uint pSeed, ref ushort pReturn1, ref ushort pReturn2, ref ushort pReturn3, ref ushort pReturn4);

[DllImport("ViKey")]

public static extern uint VikeySetSoftIDString(ushort Index, Byte[] SoftIDString);

[DllImport("ViKey")]

public static extern uint VikeyGetSoftIDString(ushort Index, Byte[] SoftIDString);

[DllImport("ViKey")]

public static extern uint ViKeySetModule(ushort Index, ushort ModelueIndex, ushort pValue, ushort pDecrease);

[DllImport("ViKey")]

public static extern uint ViKeyCheckModule(ushort Index, ushort ModelueIndex, ref ushort pIsZero, ref ushort pCanDecrase);

[DllImport("ViKey")]

public static extern uint ViKeyDecraseModule(ushort Index, ushort ModelueIndex);

[DllImport("ViKey")]

public static extern uint VikeySetPtroductName(ushort Index, Byte[] szName);

[DllImport("ViKey")]

public static extern uint VikeyGetPtroductName(ushort Index, Byte[] szName);

[DllImport("ViKey")]

public static extern uint VikeyGetTime(ushort Index, Byte[] pTime);

public static void ViKeyTest()

{

UnityEngine.Debug.Log("\*\*\*\*\*\*\*\*ViKey C# Sample\*\*\*\*\*\*\*\*");

uint HID, Count, ViKeyType = 0;

Byte[] buffer = new Byte[256];

uint retcode;

ushort j;

ushort Addr, Length;

ushort data1, data2, data3, data4;

string str1 = "1234567890123456";

string DefaultUserPassword = "11111111";

string DefaultAdminPassword = "00000000";

string strSoftIDString= "1234ABCD";

HID = Count=0;

data1 = data2 = data3 = data4 = 0;

//查找加密狗

retcode = VikeyFind(ref Count);

if (retcode != 0)

{

UnityEngine.Debug.Log("查找ViKey加密狗错误 error code: "+ retcode);

return;

}

for (j = 0; j < Count; j++)

{

//获取加密狗硬件序列号(HID)

retcode = VikeyGetHID(j, ref HID);

if (retcode != 0)

{

UnityEngine.Debug.Log("获取硬件序列号错误 error code: " + retcode);

return;

}

UnityEngine.Debug.Log("ViKey加密狗硬件序列号:" + HID);

retcode = VikeyGetType(j, ref ViKeyType);

if (retcode != 0)

{

UnityEngine.Debug.Log("获取ViKey加密狗类型 error code:"+ retcode);

return;

}

UnityEngine.Debug.Log("获取ViKey加密狗类型:" + ViKeyType);

// 用户登录ViKey加密狗

buffer = System.Text.Encoding.Default.GetBytes(DefaultUserPassword); // convert unicode to asccii

retcode = VikeyUserLogin(j, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("用户登陆ViKey加密狗失败 error code: "+ retcode);

return;

}

//管理员登陆加密狗

buffer = System.Text.Encoding.Default.GetBytes(DefaultAdminPassword); // convert unicode to asccii

retcode = VikeyAdminLogin(j, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("管理员登陆ViKey加密狗失败 error code: " + retcode);

return;

}

//ViKeyAPP不支持设置和读取设备名称

// string strPtroductName = "PtroductName";

// Byte[] szProductName = new Byte[32];

// szProductName = System.Text.Encoding.Unicode.GetBytes(strPtroductName);

// retcode = VikeySetPtroductName(j, szProductName);

// if (retcode != 0)

// {

// UnityEngine.Debug.Log("设置设备名称失败 error code: "+ retcode);

// }

//

// retcode = VikeyGetPtroductName(j, szProductName);

// if (retcode != 0)

// {

// UnityEngine.Debug.Log("获取设备名称失败 error code: "+ retcode);

// }

// else

// {

// UnityEngine.Debug.Log("获取设备名称成功: " + System.Text.Encoding.Unicode.GetString(szProductName));

// }

buffer = System.Text.Encoding.Default.GetBytes(str1); // convert unicode to asccii

// write data to Vikey

Addr = 0;

Length = 16;

retcode = VikeyWriteData(j, Addr, Length, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("写数据失败 error code: "+retcode);

return;

}

UnityEngine.Debug.Log("Write:" + str1);

// p1 = 4;

// p2 = 26;

buffer = new Byte[str1.Length];

// read dongle memory

retcode = VikeyReadData(j, Addr, Length, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("读数据失败 error code: "+ retcode);

return;

}

str1 = System.Text.Encoding.ASCII.GetString(buffer);

UnityEngine.Debug.Log("Read:" + str1);

// random generation function

retcode = ViKeyRandom(j, ref data1, ref data2, ref data3, ref data4);

if (retcode != 0)

{

UnityEngine.Debug.Log("获取随机数失败 error code: "+ retcode);

return;

}

UnityEngine.Debug.Log("Random: " + data1);

// ViKeyAPP 不支持此功能

// uint Seed;

// Seed = 0x12345678;

// retcode = VikeySeed(ref handle[j], ref Seed, ref data1, ref data2, ref data3, ref data4);

// if (retcode != 0)

// {

// UnityEngine.Debug.Log("获取种子失败 error code: "+ retcode);

// return;

// }

// UnityEngine.Debug.Log("Seed:" + data1 + " " + data2 + " " + data3 + " " + data4);

// write SoftID

buffer = System.Text.Encoding.Default.GetBytes(strSoftIDString); // convert unicode to asccii

retcode = VikeySetSoftIDString(j, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("设置软件ID error code: "+ retcode);

return;

}

UnityEngine.Debug.Log("设置软件ID: " + strSoftIDString);

// read SoftID

buffer = new Byte[strSoftIDString.Length];

retcode = VikeyGetSoftIDString(j, buffer);

if (retcode != 0)

{

UnityEngine.Debug.Log("读取软件ID error code: "+ retcode);

return;

}

strSoftIDString = System.Text.Encoding.ASCII.GetString(buffer);

UnityEngine.Debug.Log("读取软件ID: " + strSoftIDString);

//时钟狗支持获取时间

//retcode = VikeyGetTime(j, buffer);

//if (retcode != 0)

//{

// UnityEngine.Debug.Log("读取加密狗内部时间 error code: "+ retcode);

// return;

//}

//UnityEngine.Debug.Log("读取加密狗内部时间: " + buffer[0] + "\\" + buffer[1] + "\\" + buffer[2] + "\\ " +

//buffer[3] + ":" + buffer[4] + ":" + buffer[5]);

// Logoff ViKey

retcode = VikeyLogoff(j);

if (retcode != 0)

{

UnityEngine.Debug.Log("注销ViKey加密狗 error code: "+ retcode);

return;

}

}

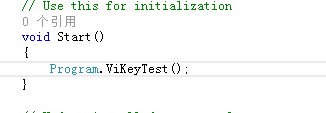
//getch();

}

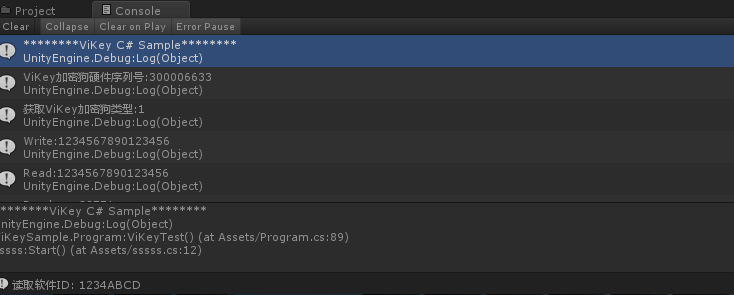
}

}

第三步：新建场景并在项目启动时调用Program.ViKeyTest();



①结果如下：加密狗插入↓



②结果如下：加密狗弹出↓

