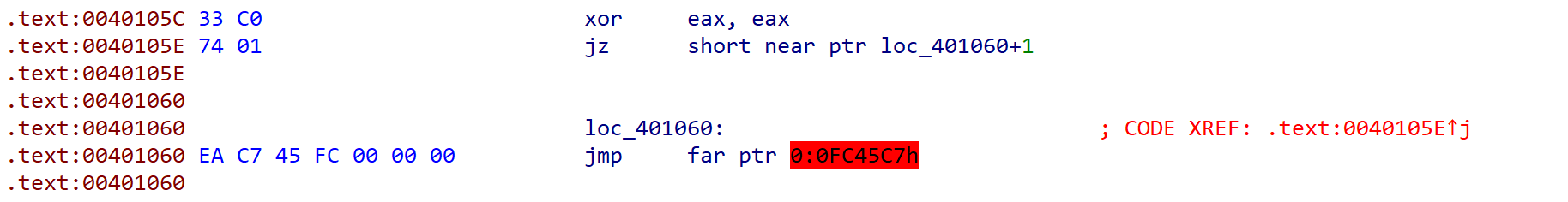
**问题-1**：程序中一共有2处花指令干扰，分别是哪一种类型的花指令？

第一处花指令：为jmp类指令形成的混淆

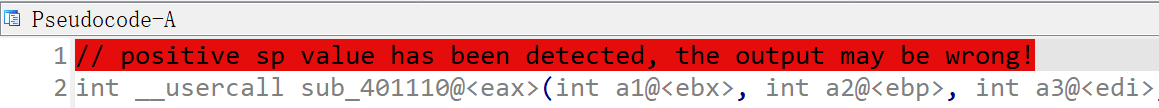


此处eax在自身异或后必为0，jz必跳转，为永恒跳转。

第二处花指令：此处花指令的类型为更改栈指针，破坏堆栈平衡

图形用户界面, 文本, 应用程序, 表格

描述已自动生成



此处花指令同上花指令，eax在自身异或后必为0，jz必跳转，为永恒跳转。

**问题-2：程序中一共有两处异常处理，分别是哪一种？**

第一处：

文本

描述已自动生成

此处错误为除零错误，触发条件为ecx为12，既输入的长度为12

第二处：

图片包含 图示

描述已自动生成

准确来说是在第004011CB处触发异常，此时函数为call sub\_401230，而sub\_401230函数经分析可知为SMC加密后的函数，当解密的密钥不正确时，SMC过程将解密错误，从而使call函数发生异常：



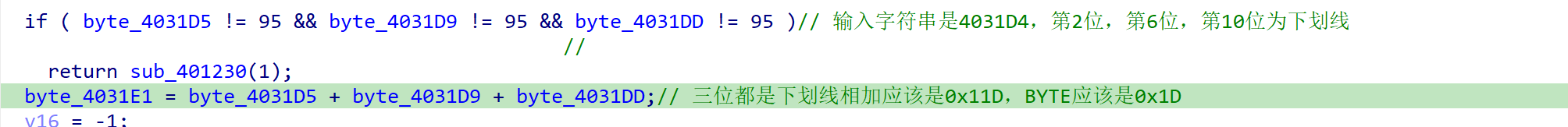
错误时很大概率会引起地址和指令被解密错误从而引起访问地址越界/错误的情况。

SMC解密函数为如图所示部分，其密钥为004031E1的值。

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

经过分析可得，正确密钥的第二位，第六位，第十位应该位下划线，此时004031E1的值正确应为0x1D，但其实三位中只要有一位时下划线都可以进入SMC解密过程，此时便会引发错误。



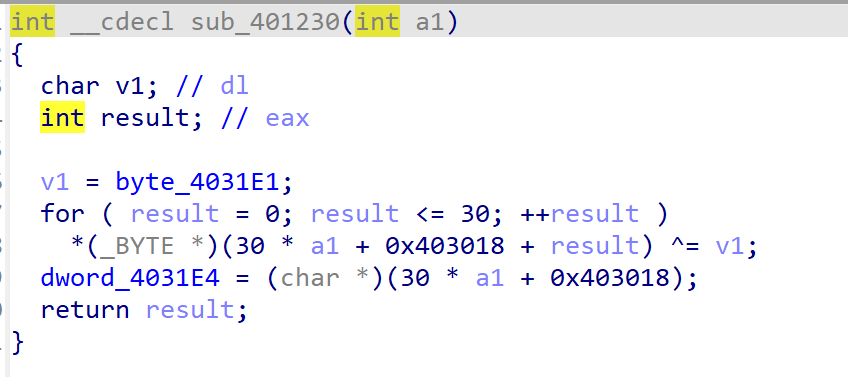
错误引发如图所示：

文本

描述已自动生成

**问题-3：程序使用了何种加密算法？在程序中总共使用了几次？各自用于实现什么目的？**

**程序主要使用的加密程序如图所示：**



其根据传入的变量a1和全局变量0x004031E1的值对内存中的两个不同字符串（取决于a1时0还是1）进行异或加密，其结果放入0x004031E4中为程序正常运行输出的值：

图形用户界面, 文本, 应用程序

描述已自动生成

因此该函数对最终结果的影响很大。

其在程序中总共使用了两次：

此次主要为如果目标的三位都不为下划线返回一个答案无关的值。

图片包含 文本

描述已自动生成

第二次运行如图所示，此处为答案为正确情况时，能过正确将0x403018处的字符串变为目标输出。

图片包含 文本

描述已自动生成

另有两处加密分别为上述的SMC解密，和SMC解密后的函数内部：

利用输入部分位进行异或加密和密钥进行比较

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**问题-4：画出程序的算法流程图，给出输入正确f FLAG 时的 CMD 运行窗口截图**

结合上述所有分析，我们先分析程序的正确流程：

首先在main函数内输入长度时12位，引发除零错误，转到错误处理，call sub\_401110函数。

图形用户界面, 表格

中度可信度描述已自动生成

文本

低可信度描述已自动生成

进入sub\_401110函数：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

首先如上分析，我们得让输入的第2，6，10位为下划线，使得后面的解密密钥正确为0x1D

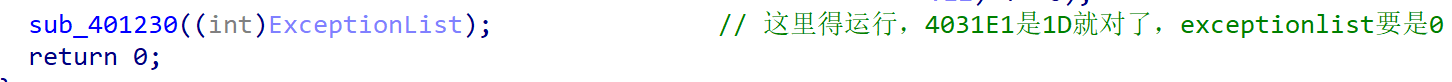
然后我们调用插件将0x004030E0处的函数解密，得到如下结果：

文本

描述已自动生成

结合代码和如图注释即可分析出密钥的剩余部分

如此我们才能使得上述提到的加密函数的第二次调用：



发挥作用。

然后当加密函数正确后，程序返回main部分，执行：

文本, 应用程序

描述已自动生成

将结果输出。

故而，通过上述过程，我们能分析出正确的密钥为：

1\_Kai\_Xin\_Wo

正确执行的结果如下：

文本

描述已自动生成

程序正确运行时流程图如下：

