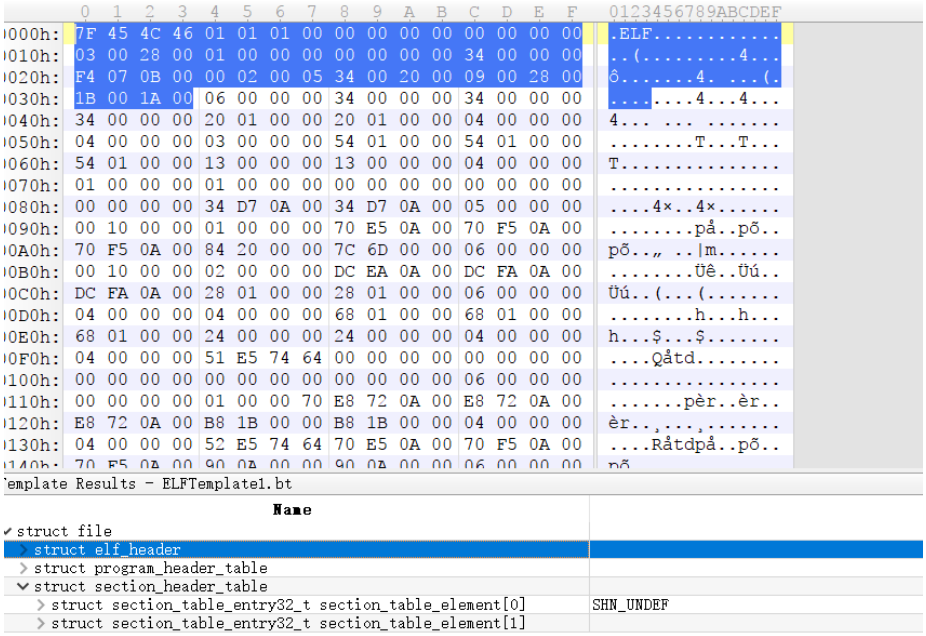
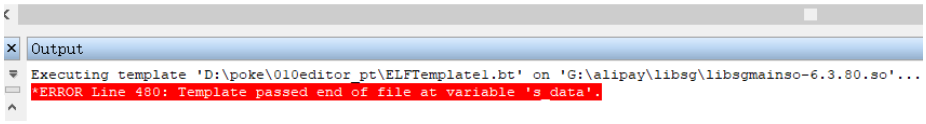
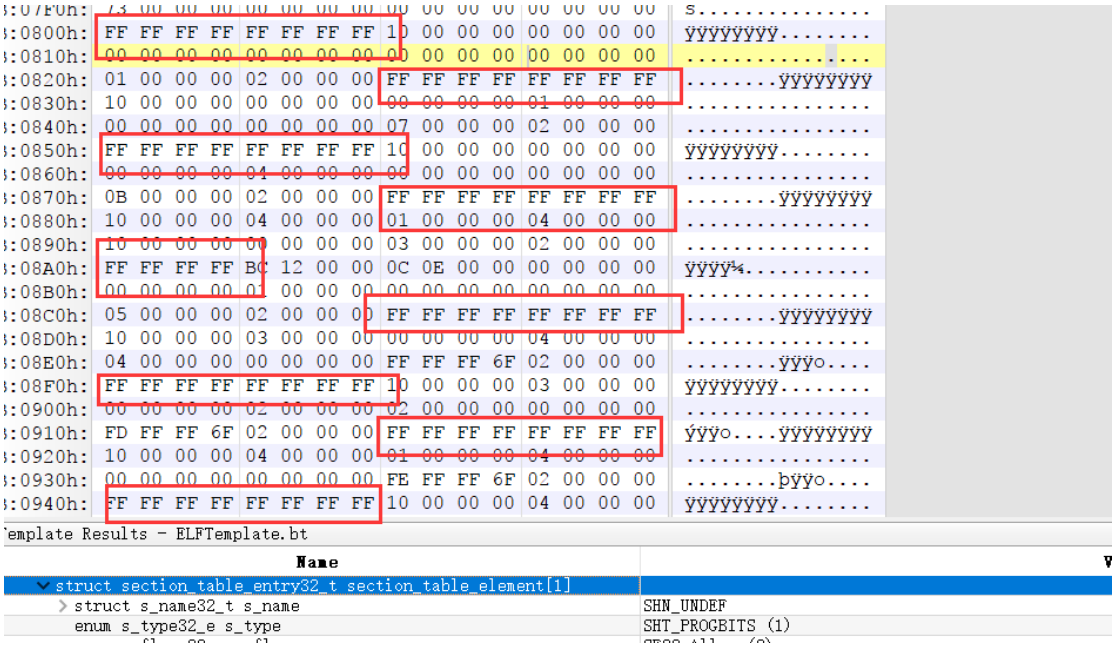
安全 sdk-so 保护

抹掉 Section 节表信息

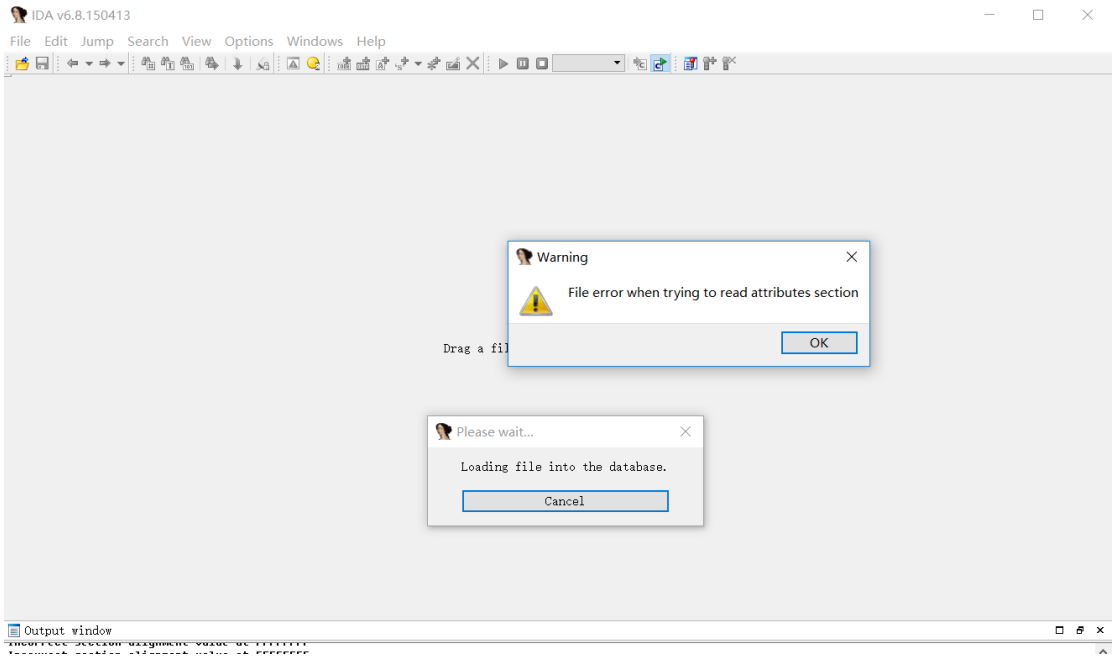




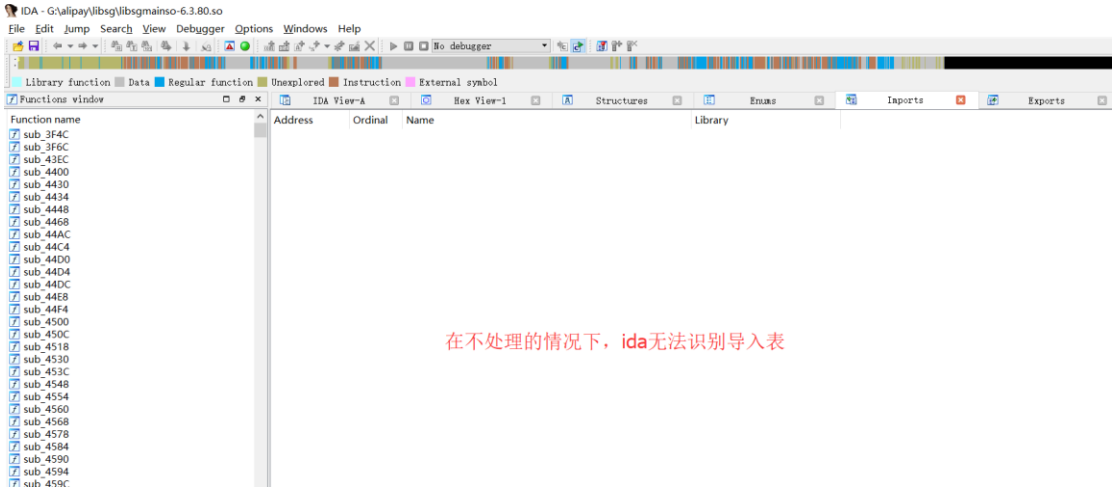
节头表信息:

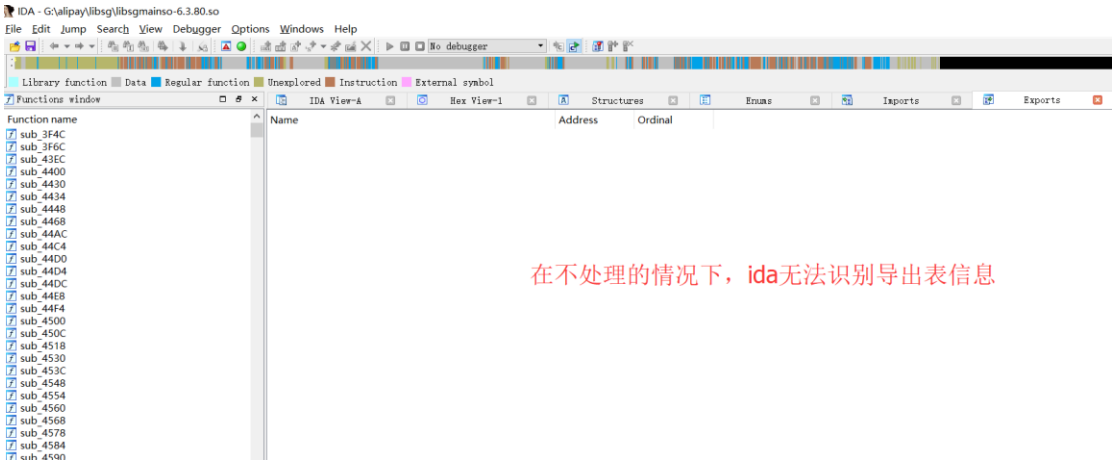


抹掉节头表导致 IDA（类型的基于节头表解析的逆向工具）加载失败



IDA 无法识别导入导出表





垃圾数据、指令

```
libsgmainso_6.3.80.so:E93DDA68
libsgmainso_6.3.80.so:E93DDA68 000 70 B5       PUSH            {R4-R6,LR}
libsgmainso_6.3.80.so:E93DDA6A 010 09 A4       ADR             R4, unk_E93DDA90
libsgmainso_6.3.80.so:E93DDA6C 010 02 25 20 35 MOVS            R5, #0x22
libsgmainso_6.3.80.so:E93DDA70 010 64 19       ADDS            R4, R4, R5
libsgmainso_6.3.80.so:E93DDA72 010 64 1C       ADDS            R4, R4, #1
libsgmainso_6.3.80.so:E93DDA74 010 01 D0       BEQ             loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76 010 00 D1       BNE             loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA78 010 7F BD       POP             {R0-R6,PC}
libsgmainso_6.3.80.so:E93DDA7A                 ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A                 loc_E93DDA7A                            ; CODE XREF: sub_E93DDA68+CTj
libsgmainso_6.3.80.so:E93DDA7A                                                         ; sub_E93DDA68+ETj
libsgmainso_6.3.80.so:E93DDA7A 010 03 94       STR             R4, [SP,#0x10+var_4]
libsgmainso_6.3.80.so:E93DDA7C 010 00 9C       LDR             R4, [SP,#0x10+var_10]
libsgmainso_6.3.80.so:E93DDA7E 010 01 9D       LDR             R5, [SP,#0x10+var_C]
libsgmainso_6.3.80.so:E93DDA80 010 6E 46       MOV             R6, SP
libsgmainso_6.3.80.so:E93DDA82 010 08 36       ADDS            R6, #8
libsgmainso_6.3.80.so:E93DDA84 010 B5 46       MOV             SP, R6
libsgmainso_6.3.80.so:E93DDA86 010 40 BD       POP             {R6,PC}
libsgmainso_6.3.80.so:E93DDA86                 ; End of function sub_E93DDA68
libsgmainso_6.3.80.so:E93DDA86
libsgmainso_6.3.80.so:E93DDA86                 ; ---------------------------------------------------------------------------
```

迷惑代码，根本不会被执行

不管是否相等，都跳转到同一个地方

```
libsgmainso_6.3.80.so:E93DDA68
libsgmainso_6.3.80.so:E93DDA68 000 70 B5       PUSH            {R4-R6,LR}
libsgmainso_6.3.80.so:E93DDA6A 010 09 A4       ADR             R4, unk_E93DDA90
libsgmainso_6.3.80.so:E93DDA6C 010 02 25 20 35 MOVS            R5, #0x22
libsgmainso_6.3.80.so:E93DDA70 010 64 19       ADDS            R4, R4, R5
libsgmainso_6.3.80.so:E93DDA72 010 64 1C       ADDS            R4, R4, #1
libsgmainso_6.3.80.so:E93DDA74 010 01 D0       BEQ             loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76 010 00 D1       BNE             loc_E93DDA7A
libsgmainso_6.3.80.so:E93DDA76                 ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E93DDA78 010 7F BD       DCW 0xBD7F
libsgmainso_6.3.80.so:E93DDA7A                 ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E93DDA7A
libsgmainso_6.3.80.so:E93DDA7A                 loc_E93DDA7A                            ; CODE XREF: sub_E93DDA68+CTj
libsgmainso_6.3.80.so:E93DDA7A                                                         ; sub_E93DDA68+ETj
libsgmainso_6.3.80.so:E93DDA7A 010 03 94       STR             R4, [SP,#0x10+var_4]
libsgmainso_6.3.80.so:E93DDA7C 010 00 9C       LDR             R4, [SP,#0x10+var_10]
libsgmainso_6.3.80.so:E93DDA7E 010 01 9D       LDR             R5, [SP,#0x10+var_C]
libsgmainso_6.3.80.so:E93DDA80 010 6E 46       MOV             R6, SP
libsgmainso_6.3.80.so:E93DDA82 010 08 36       ADDS            R6, #8
libsgmainso_6.3.80.so:E93DDA84 010 B5 46       MOV             SP, R6
libsgmainso_6.3.80.so:E93DDA86 010 40 BD       POP             {R6,PC}
libsgmainso_6.3.80.so:E93DDA86                 ; End of function sub_E93DDA68
libsgmainso_6.3.80.so:E93DDA86
```

应该这样，条件为真

类似的迷惑代码

```
sgmainso_6.3.80.so:EF416A88 010 97 46 04 B5 DCD 0xB5044697
sgmainso_6.3.80.so:EF416A8C 010 07 BD C0 46 DCD 0x46C0BD07
sgmainso_6.3.80.so:EF416A90                 ; ---------------------------------------------------------------------------
sgmainso_6.3.80.so:EF416A90
sgmainso_6.3.80.so:EF416A90                 loc_EF416A90                            ; DATA XREF: JNI_OnLoad+2To
sgmainso_6.3.80.so:EF416A90 010 11 60       STR             R1, [R2]
sgmainso_6.3.80.so:EF416A92 010 70 BD       POP             {R4-R6,PC}
sgmainso_6.3.80.so:EF416A92                 ; End of function JNI_OnLoad
sgmainso_6.3.80.so:EF416A92
sgmainso_6.3.80.so:EF416A94                 ; ---------------------------------------------------------------------------
sgmainso_6.3.80.so:EF416A94 11 60           STR             R1, [R2]
sgmainso_6.3.80.so:EF416A96 08 BD           POP             {R3,PC}
sgmainso_6.3.80.so:EF416A98                 ; ---------------------------------------------------------------------------
sgmainso_6.3.80.so:EF416A98 B7 46           MOV             PC, R6
sgmainso_6.3.80.so:EF416A9A                 ; ---------------------------------------------------------------------------
sgmainso_6.3.80.so:EF416A9A 0C B5           PUSH            {R2,R3,LR}
sgmainso_6.3.80.so:EF416A9C 0B 00           MOVS            R3, R1
sgmainso_6.3.80.so:EF416A9E 0C B5           PUSH            {R2,R3,LR}
sgmainso_6.3.80.so:EF416AA0 F0 47           BLX             LR
sgmainso_6.3.80.so:EF416AA0                 ; ---------------------------------------------------------------------------
sgmainso_6.3.80.so:EF416AA2                 CODE32
sgmainso_6.3.80.so:EF416AA2 38              DCB 0x38 ; 8
sgmainso_6.3.80.so:EF416AA3 BD              DCB 0xBD ;
sgmainso_6.3.80.so:EF416AA4 11              DCB 0x11
```

垃圾数据，迷惑反汇编器

动态计算目标地址

```
libsgmainso_6.3.80.so:EED8D848                                          ; libsgmainso_6.3.80.so:off_EEE2CAE4↓o
libsgmainso_6.3.80.so:EED8D848 01 10 CE E3    BIC             R1, LR, #1
libsgmainso_6.3.80.so:EED8D84C 00 01 91 E7    LDR             R0, [R1,R0,LSL#2]
libsgmainso_6.3.80.so:EED8D850 0E E0 80 E0    ADD             LR, R0, LR
libsgmainso_6.3.80.so:EED8D854 08 10 9D E5    LDR             R1, [SP,#8]
libsgmainso_6.3.80.so:EED8D858 08 E0 8D E5    STR             LR, [SP,#8]
libsgmainso_6.3.80.so:EED8D85C 01 E0 A0 E1    MOV             LR, R1
libsgmainso_6.3.80.so:EED8D860 03 80 BD E8    LDMFD           SP!, {R0,R1,PC}
libsgmainso_6.3.80.so:EED8D864                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:EED8D864 0E 10 A0 E1    MOV             R1, LR
libsgmainso_6.3.80.so:EED8D868 A1 10 A0 E1    MOV             R1, R1,LSR#1
libsgmainso_6.3.80.so:EED8D86C 81 10 A0 E1    MOV             R1, R1,LSL#1
libsgmainso_6.3.80.so:EED8D870 01 00 A0 E1    MOV             R0, R1
libsgmainso_6.3.80.so:EED8D874 00 10 91 E5    LDR             R1, [R1]
libsgmainso_6.3.80.so:EED8D878 00 10 81 E0    ADD             R1, R1, R0
libsgmainso_6.3.80.so:EED8D87C 00 00 91 E5    LDR             R0, [R1]
libsgmainso_6.3.80.so:EED8D880 10 00 8D E5    STR             R0, [SP,#0x10]
libsgmainso_6.3.80.so:EED8D884 04 E0 8E E2    ADD             LR, LR, #4
libsgmainso_6.3.80.so:EED8D888 0C E0 8D E5    STR             LR, [SP,#0xC]
libsgmainso_6.3.80.so:EED8D88C 03 40 BD E8    LDMFD           SP!, {R0,R1,LR}
libsgmainso_6.3.80.so:EED8D890 04 F0 9D E4    LDR             PC, [SP],#4
libsgmainso_6.3.80.so:EED8D894                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:EED8D894 03 00 2D E9    STMFD           SP!, {R0,R1}
libsgmainso_6.3.80.so:EED8D898 0E 10 A0 E1    MOV             R1, LR
libsgmainso_6.3.80.so:EED8D89C A1 10 A0 E1    MOV             R1, R1,LSR#1
libsgmainso_6.3.80.so:EED8D8A0 00 01 A0 E1    MOV             R0, R0,LSL#2
libsgmainso_6.3.80.so:EED8D8A4 81 10 A0 E1    MOV             R1, R1,LSL#1
libsgmainso_6.3.80.so:EED8D8A8 00 10 91 E7    LDR             R1, [R1,R0]
libsgmainso_6.3.80.so:EED8D8AC 81 10 A0 E1    MOV             R1, R1,LSL#1
```

很多，这里只是截三幅图

这样会导致调用不连续，只有计算出 pc 值才能分析清楚程序走向。

Ida 无法解析出正确的流程图和参数（垃圾代码）

下面的 JNI_OnLoad 是我动态调试找到的，但是 ida 明显解析错了参数，和函数流程。

```
void __fastcall JNI_OnLoad(int a1, int a2, int a3, int a4, int a5, int a6, int a7, int a8)
{
  int v8; // [sp+4h] [bp-Ch]@0

  if ( &loc_EF096AB2 )
  {
    if ( !&loc_EF096AB2 )
      JUMPOUT(__CS__, a8);
  }
  JUMPOUT(__CS__, v8);
}
```

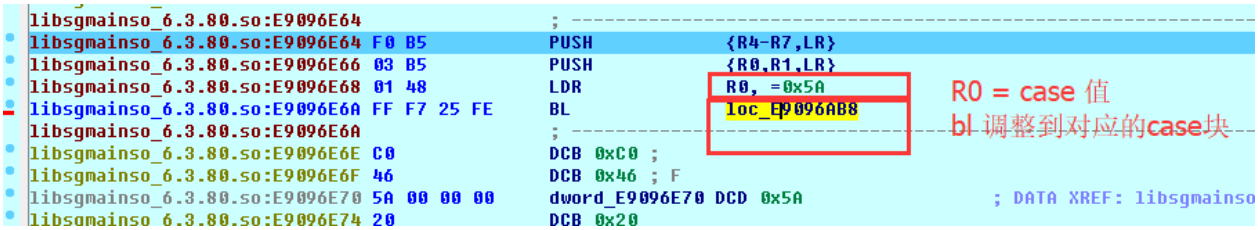
流程图：

这导致你必须单步调试，一边调试，一边帮助 ida 修正错误。

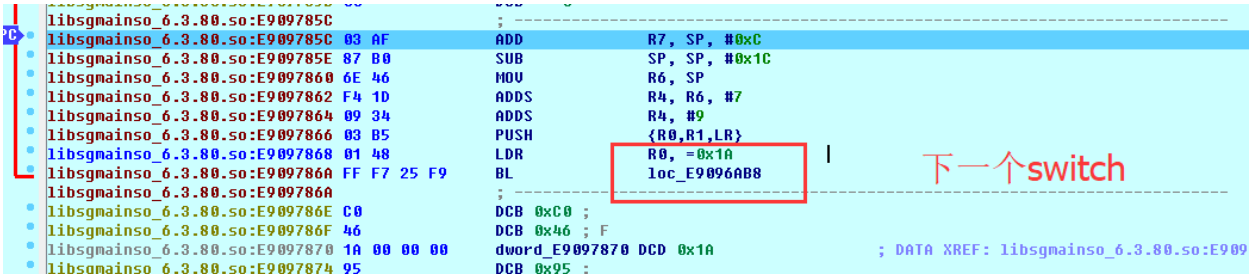
llvm 混淆

花指令一般不会被执行或者即使被执行它也不会改变堆栈平衡不会影响程序逻辑，而下面的特征符合 llvm 混淆（指令平坦化特征，要实现平坦化的方法分成多个基本块（就是 case 代码块）和一个入口块，为每个基本快编号，并让这些基本块都有共同的前驱模块和后继模块。前驱模块主要是进行基本块的分发，分发通过改变 switch 变量来实现。后继模块也可用于更新 switch 变量的值，并跳转到 switch 开始处），或者说是花指令和指令平坦化的结合。
参考阿里聚安全的一篇文章：
https://jaq.alibaba.com/community/art/show?articleid=835
它执行几个字节的指令，然后进入 case 块，执行完 case 块

Switch 选择块



执行完跳转到下一个 switch 块



跳转到 case 块的地方

```
libsgmainso_6.3.80.so:E912CAE8                ; Attributes: thunk
libsgmainso_6.3.80.so:E912CAE8
libsgmainso_6.3.80.so:E912CAE8                sub_E912CAE8
libsgmainso_6.3.80.so:E912CAE8 000 78 47      BX              PC
libsgmainso_6.3.80.so:E912CAE8                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E912CAEA 000 C0 46      ALIGN 4
libsgmainso_6.3.80.so:E912CAEA                ; End of function sub_E912CAE8
libsgmainso_6.3.80.so:E912CAEA
libsgmainso_6.3.80.so:E912CAEC                CODE32
libsgmainso_6.3.80.so:E912CAEC
libsgmainso_6.3.80.so:E912CAEC                ; =============== S U B R O U T I N E =======================================
libsgmainso_6.3.80.so:E912CAEC
libsgmainso_6.3.80.so:E912CAEC                ; Attributes: thunk
libsgmainso_6.3.80.so:E912CAEC
libsgmainso_6.3.80.so:E912CAEC                sub_E912CAEC                            ; CODE XREF: sub_E912CAE8↑j
libsgmainso_6.3.80.so:E912CAEC 000 00 C0 9F E5 LDR             R12, =(loc_E908D8C8 - 0xE912CAF8)
libsgmainso_6.3.80.so:E912CAF0 000 0F F0 8C E0 ADD             PC, R12, PC ; loc_E908D8C8
libsgmainso_6.3.80.so:E912CAF0 000            ; End of function sub_E912CAEC
libsgmainso_6.3.80.so:E912CAF0
libsgmainso_6.3.80.so:E912CAF0                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E912CAF4 D0 0D F6 FF    off_E912CAF4 DCD loc_E908D8C8 - 0xE912CAF8
libsgmainso_6.3.80.so:E912CAF4                                                        ; DATA XREF: sub_E912CAEC↑r
libsgmainso_6.3.80.so:E912CAF8                CODE16
libsgmainso_6.3.80.so:E912CAF8
libsgmainso_6.3.80.so:E912CAF8                ; =============== S U B R O U T I N E =======================================
libsgmainso_6.3.80.so:E912CAF8
libsgmainso_6.3.80.so:E912CAF8                ; Attributes: thunk
libsgmainso_6.3.80.so:E912CAF8
libsgmainso_6.3.80.so:E912CAF8                sub_E912CAF8
libsgmainso_6.3.80.so:E912CAF8 000 78 47      BX              PC
libsgmainso_6.3.80.so:E912CAF8                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E912CAFA 000 C0 46      ALIGN 4
libsgmainso_6.3.80.so:E912CAFA                ; End of function sub_E912CAF8
libsgmainso_6.3.80.so:E912CAFA
libsgmainso_6.3.80.so:E912CAFC                CODE32
libsgmainso_6.3.80.so:E912CAFC
libsgmainso_6.3.80.so:E912CAFC                ; =============== S U B R O U T I N E =======================================
libsgmainso_6.3.80.so:E912CAFC
libsgmainso_6.3.80.so:E912CAFC                ; Attributes: thunk
libsgmainso_6.3.80.so:E912CAFC
libsgmainso_6.3.80.so:E912CAFC                sub_E912CAFC                            ; CODE XREF: sub_E912CAF8↑j
libsgmainso_6.3.80.so:E912CAFC 000 00 C0 9F E5 LDR             R12, =(loc_E908D864 - 0xE912CB08)
libsgmainso_6.3.80.so:E912CB00 000 0F F0 8C E0 ADD             PC, R12, PC ; loc_E908D864
libsgmainso_6.3.80.so:E912CB00 000            ; End of function sub_E912CAFC
libsgmainso_6.3.80.so:E912CB00
libsgmainso_6.3.80.so:E912CB00                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E912CB04 5C 0D F6 FF    off_E912CB04 DCD loc_E908D864 - 0xE912CB08
libsgmainso_6.3.80.so:E912CB04                                                        ; DATA XREF: sub_E912CAFC↑r
libsgmainso_6.3.80.so:E912CB08 78             DCB 0x78 ; x
libsgmainso_6.3.80.so:E912CB09 47             DCB 0x47 ; G
libsgmainso_6.3.80.so:E912CB0A C0             DCB 0xC0 ;
libsgmainso_6.3.80.so:E912CB0B 46             DCB 0x46 ; F
libsgmainso_6.3.80.so:E912CB0C 00             DCB    0
libsgmainso_6.3.80.so:E912CB0D C0             DCB 0xC0 ;
libsgmainso_6.3.80.so:E912CB0E 9F             DCB 0x9F ;
libsgmainso_6.3.80.so:E912CB0F E5             DCB 0xE5 ;
libsgmainso_6.3.80.so:E912CB10 0F             DCB  0xF
libsgmainso_6.3.80.so:E912CB11 F0             DCB 0xF0 ;
libsgmainso_6.3.80.so:E912CB12 8C             DCB 0x8C ;
libsgmainso_6.3.80.so:E912CB13 E0             DCB 0xE0 ;
libsgmainso_6.3.80.so:E912CB14 84             DCB 0x84 ;
libsgmainso_6.3.80.so:E912CB15 04             DCB    4
libsgmainso_6.3.80.so:E912CB16 F6             DCB 0xF6 ;
libsgmainso_6.3.80.so:E912CB17 FF             DCB 0xFF
libsgmainso_6.3.80.so:E912CB18 78             DCB 0x78 ; x
```

case块特征，机器码47 78开始0x10个字节，case块特别多

Case 块

```
libsgmainso_6.3.80.so:E908D848                                                        ; libsgmainso_6.3.80.so:off_E912CAE4↓o
libsgmainso_6.3.80.so:E908D848 01 10 CE E3    BIC             R1, LR, #1
libsgmainso_6.3.80.so:E908D84C 00 01 91 E7    LDR             R0, [R1,R0,LSL#2]
libsgmainso_6.3.80.so:E908D850 0E E0 80 E0    ADD             LR, R0, LR
libsgmainso_6.3.80.so:E908D854 08 10 9D E5    LDR             R1, [SP,#8]
libsgmainso_6.3.80.so:E908D858 08 E0 8D E5    STR             LR, [SP,#8]
libsgmainso_6.3.80.so:E908D85C 01 E0 A0 E1    MOV             LR, R1
libsgmainso_6.3.80.so:E908D860 03 80 BD E8    LDMFD           SP!, {R0,R1,PC}
libsgmainso_6.3.80.so:E908D864                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E908D864
libsgmainso_6.3.80.so:E908D864                loc_E908D864                            ; DATA XREF: sub_E912CAFC+4↓o
libsgmainso_6.3.80.so:E908D864                                                        ; libsgmainso_6.3.80.so:off_E912CB04↓o
libsgmainso_6.3.80.so:E908D864 0E 10 A0 E1    MOV             R1, LR
libsgmainso_6.3.80.so:E908D868 A1 10 A0 E1    MOV             R1, R1,LSR#1
libsgmainso_6.3.80.so:E908D86C 81 10 A0 E1    MOV             R1, R1,LSL#1
libsgmainso_6.3.80.so:E908D870 01 00 A0 E1    MOV             R0, R1
libsgmainso_6.3.80.so:E908D874 00 10 91 E5    LDR             R1, [R1]
libsgmainso_6.3.80.so:E908D878 00 10 81 E0    ADD             R1, R1, R0
libsgmainso_6.3.80.so:E908D87C 00 00 91 E5    LDR             R0, [R1]
libsgmainso_6.3.80.so:E908D880 10 00 8D E5    STR             R0, [SP,#0x10]
libsgmainso_6.3.80.so:E908D884 04 E0 8E E2    ADD             LR, LR, #4
libsgmainso_6.3.80.so:E908D888 0C E0 8D E5    STR             LR, [SP,#0xC]
libsgmainso_6.3.80.so:E908D88C 03 40 BD E8    LDMFD           SP!, {R0,R1,LR}
libsgmainso_6.3.80.so:E908D890 04 F0 9D E4    LDR             PC, [SP],#4
libsgmainso_6.3.80.so:E908D894                ; ---------------------------------------------------------------------------
libsgmainso_6.3.80.so:E908D894 03 00 2D E9    STMFD           SP!, {R0,R1}
libsgmainso_6.3.80.so:E908D898 0E 10 A0 E1    MOV             R1, LR
libsgmainso_6.3.80.so:E908D89C A1 10 A0 E1    MOV             R1, R1,LSR#1
libsgmainso_6.3.80.so:E908D8A0 00 01 A0 E1    MOV             R0, R0,LSL#2
libsgmainso_6.3.80.so:E908D8A4 81 10 A0 E1    MOV             R1, R1,LSL#1
libsgmainso_6.3.80.so:E908D8A8 00 10 91 E7    LDR             R1, [R1,R0]
libsgmainso_6.3.80.so:E908D8AC 81 10 A0 E1    MOV             R1, R1,LSL#1
libsgmainso_6.3.80.so:E908D8B0 01 E0 8E E0    ADD             LR, LR, R1
libsgmainso_6.3.80.so:E908D8B4 03 00 BD E8    LDMFD           SP!, {R0,R1}
libsgmainso_6.3.80.so:E908D8B8 08 00 9D E5    LDR             R0, [SP,#8]
```

case块，每段直接被ida用下划线分割

但很遗憾 pc 值是动态的，导致即使你能找到 switch case 块，也无法构造完整的程序流程图。

补充

由于写这些文档时已经停止了分析工作，so 保护应该还有我没有发现的部分，此文档只是提供我发现的保护措施供大家分享。

这里面保护强度最强的应该算 llvm 混淆，在无法去掉混淆时（基本接近无解），只能一步一步调试（消耗时间成本）。