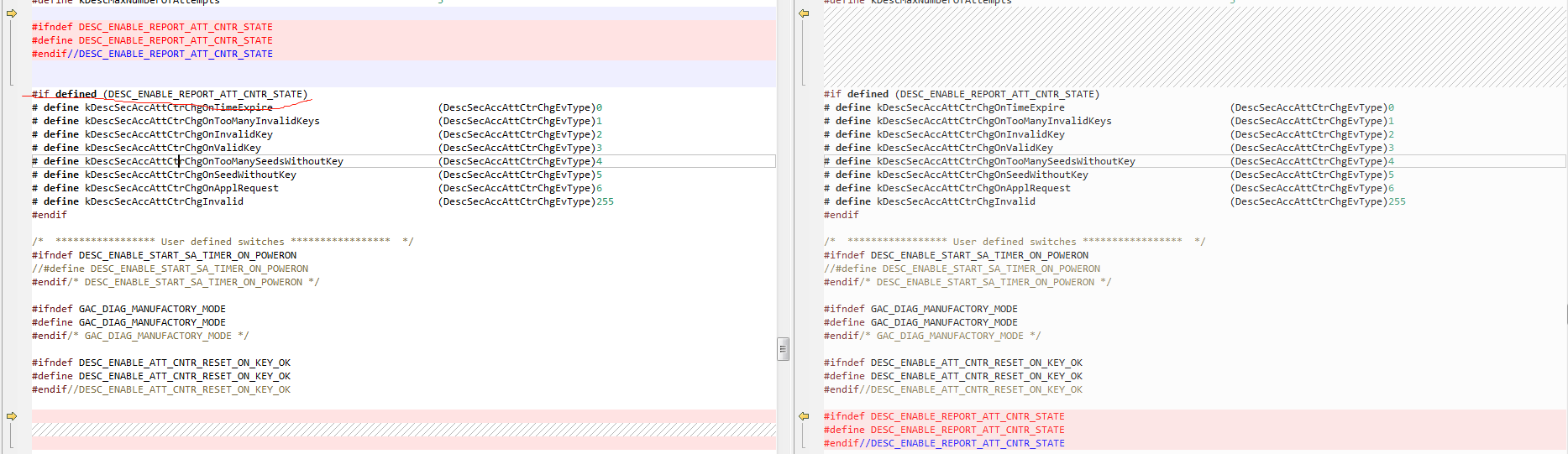
|  |  |  |  |
| --- | --- | --- | --- |
| 修改记录  注：如果修改点不熟悉，不确定修改点是否修改到位，请让周外阳来修改或者确认。 | | | |
| 修改人员 | 时间 | 修改点 | 备注 |
| 周外阳 |  | 第4条：功能寻址 0x7E 0x7F不做出响应 | Geny979补丁 |
| 周外阳 |  | 第16条：修改2E服务的NRC31优先级大于NRC13(long)。 | AM8功能要求 |
| 周外阳 |  | 第6条：限制28服务的通信控制参数范围 | AM8功能要求 |
| 周外阳 |  | 第17条：ccl\_par.c中屏蔽B-Can | AM8功能要求 |
| 周外阳 |  | 第19条：会话时EngStatus = run，回NRC22 | AM8功能要求 |
| 周外阳 |  | 第18条：desc.c中，工厂模式默认安全解锁 | AM8功能要求 |
| 周外阳 |  | 第20条： desc.c中，目的是27 01重复请求，回复相同seed。 | AM8功能要求 |
| 周外阳 |  | 14.desc.c文件中，组合ID超出回复NRC13 | AM8功能要求 |
| 周外阳 |  | 第21条：desc.c中，添加函数DescSecurityOnceInitPowerOn\_GAC(void); | AM8功能要求 |
| 周外阳 |  | 第22条： desc.c中，安全访问，delaytime和失败次数。 | AM8功能要求 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**1. 在desc.h中,**将一个宏的定义移到它本身的使用前，否则编译不过

搜索宏：DESC\_ENABLE\_REPORT\_ATT\_CNTR\_STATE



**2. desc.h**

增加22服务多个DID访问的个数，5改为10

索引宏：kDescNumMaxPidList



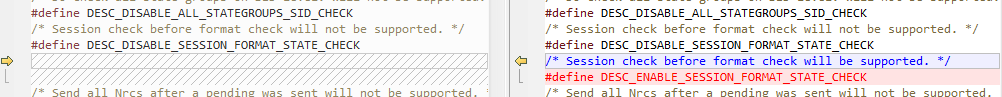
**3.desc.h中**，添加一个宏：398行DESC\_DISABLE\_SESSION\_FORMAT\_STATE\_CHECK，使得判断NRC13（total）之前判断NRC7E。

搜索宏：DESC\_DISABLE\_SESSION\_FORMAT\_STATE\_CHECK

复制黏贴：

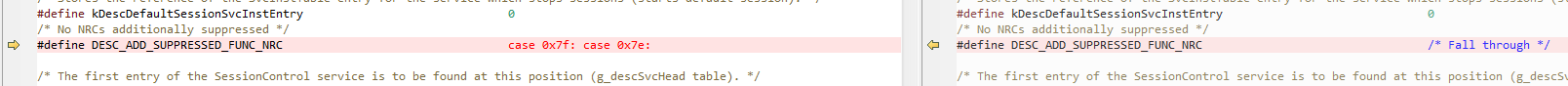
/\* Session check before format check will be supported. \*/

#define DESC\_ENABLE\_SESSION\_FORMAT\_STATE\_CHECK



4. 在desc.c中,功能寻址 0x7E 0x7F不做出响应

搜索宏：DESC\_ADD\_SUPPRESSED\_FUNC\_NRC

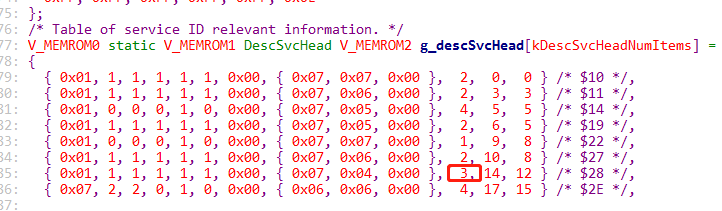


#define DESC\_ADD\_SUPPRESSED\_FUNC\_NRC case 0x7f: case 0x7e:

**5.desc.c**

把诊断28服务的最小长度2改为3

搜索数组：g\_descSvcHead

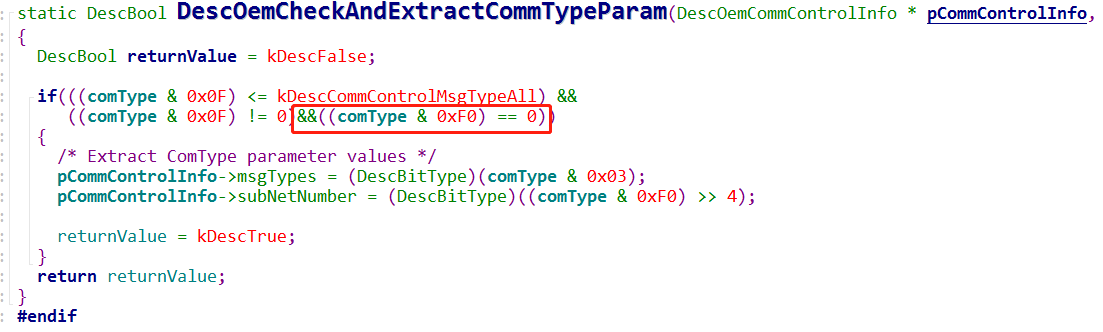


**6．Desc.c**

限制28服务的通信控制参数范围

索引函数：DescOemCheckAndExtractCommTypeParam

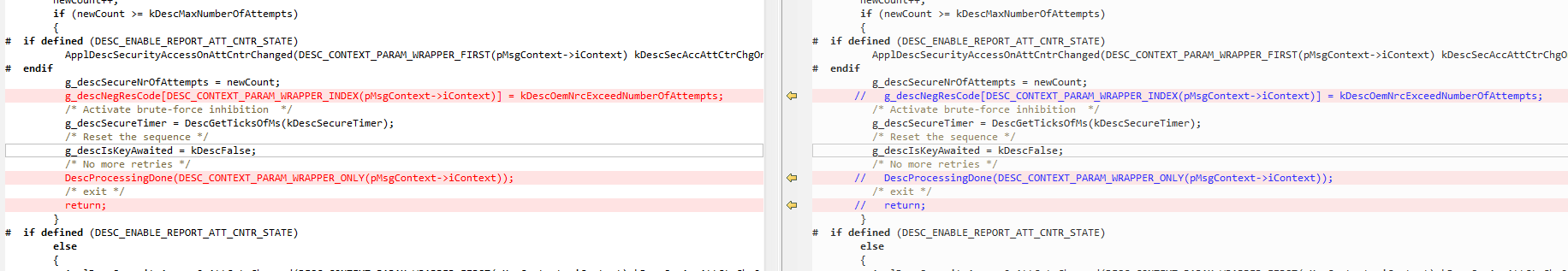
复制黏贴：((comType & 0x0F) != 0)&&((comType & 0xF0) == 0))



7. **Desc.c中，**修改0x27服务的安全访问尝试计数器逻辑

搜索宏： DescOemSecuritySeed

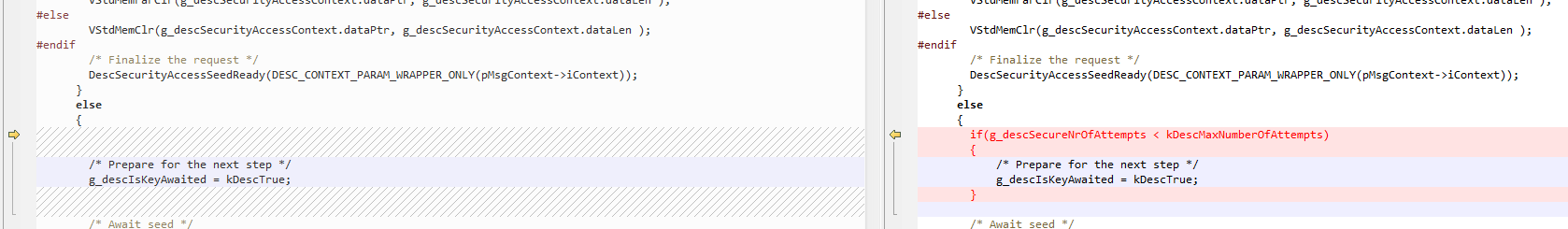
对比旧软件，直接替换该函数



if(g\_descSecureNrOfAttempts < kDescMaxNumberOfAttempts)

{

}

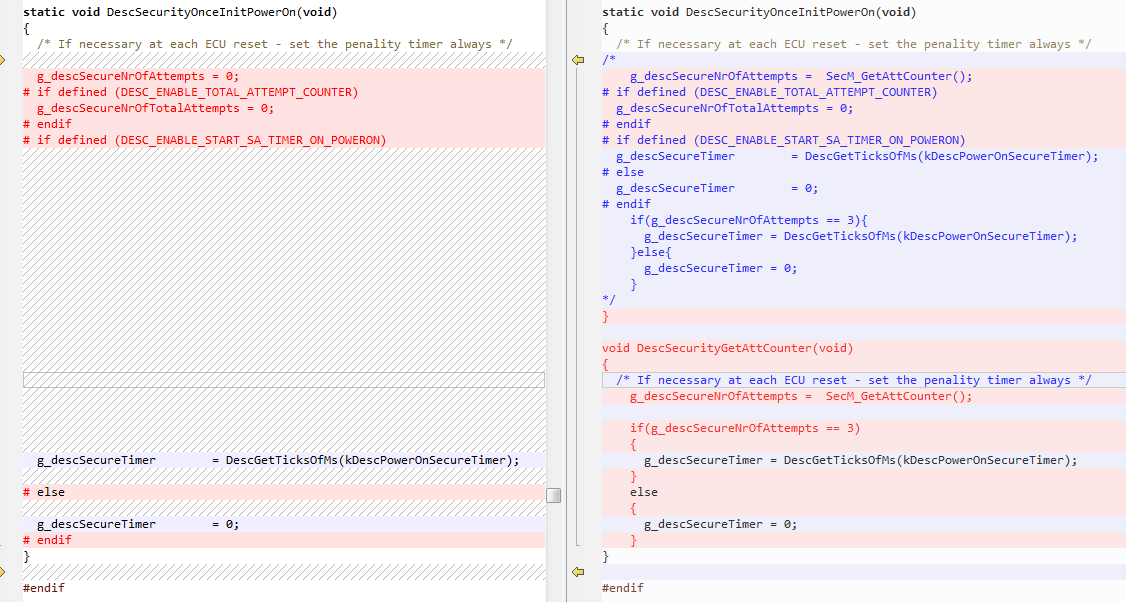


**8. 在desc.c中**,重构上电后安全计数器设置的函数

更改函数：DescSecurityOnceInitPowerOn

搜索宏：DescSecurityOnceInitPowerOn

对比旧软件，直接替换部分



static void DescSecurityOnceInitPowerOn(void)

{

/\*

g\_descSecureNrOfAttempts = SecM\_GetAttCounter();

# if defined (DESC\_ENABLE\_TOTAL\_ATTEMPT\_COUNTER)

g\_descSecureNrOfTotalAttempts = 0;

# endif

# if defined (DESC\_ENABLE\_START\_SA\_TIMER\_ON\_POWERON)

g\_descSecureTimer = DescGetTicksOfMs(kDescPowerOnSecureTimer);

# else

g\_descSecureTimer = 0;

# endif

if(g\_descSecureNrOfAttempts == 3){

g\_descSecureTimer = DescGetTicksOfMs(kDescPowerOnSecureTimer);

}else{

g\_descSecureTimer = 0;

}

\*/

}

void DescSecurityGetAttCounter(void)

{

/\* If necessary at each ECU reset - set the penality timer always \*/

g\_descSecureNrOfAttempts = SecM\_GetAttCounter();

if(g\_descSecureNrOfAttempts == 3)

{

g\_descSecureTimer = DescGetTicksOfMs(kDescPowerOnSecureTimer);

}

else

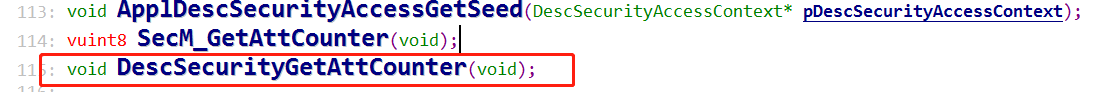
{

g\_descSecureTimer = 0;

}

}

然后再appdesc.h中添加一个函数声明：void DescSecurityGetAttCounter(void);

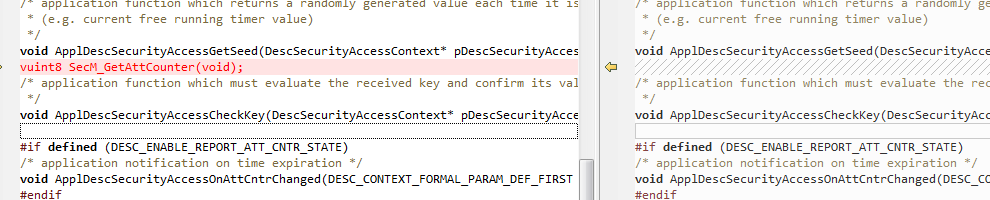


索引：void ApplDescSecurityAccessGetSeed(DescSecurityAccessContext\* pDescSecurityAccessContext);

void DescSecurityGetAttCounter(void);

vuint8 SecM\_GetAttCounter(void);（第9点）

9. 在appdesc.h中,新增一个函数vuint8 SecM\_GetAttCounter(void);用于上电后读取EEP存储的安全访问失败计数

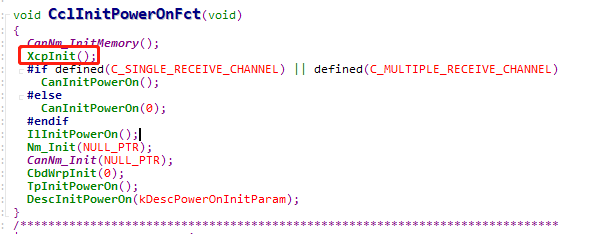


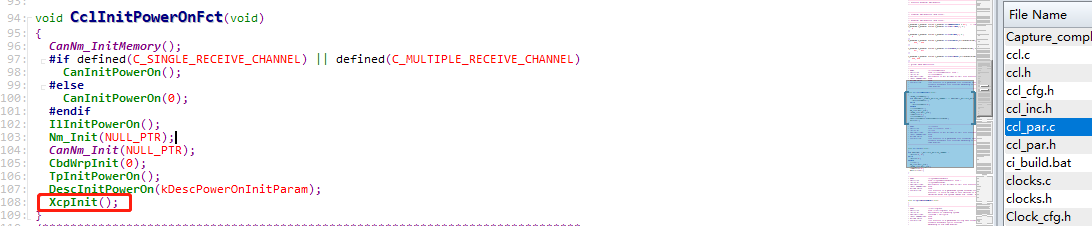
10、ccl\_par.c

Geny后XCP的初始化需要在CanInitPowerOn之后：

把下图红框框住的函数放在最下面调用。

索引：CclInitPowerOnFct(void)





**11.** **ccl\_par.c中，Bussleep 状态下，不响应诊断报文**

**在CclScheduleTask函数中增加：**

**Nm\_StateType gubNmState;**

**Nm\_ModeType gubNmMode;**

**Nm\_GetState (0 , &gubNmState, &gubNmMode );**

**if((gubNmState!=NM\_STATE\_UNINIT) && (gubNmState!=NM\_STATE\_BUS\_SLEEP ))**

**{**

**DescTask();**

**}**



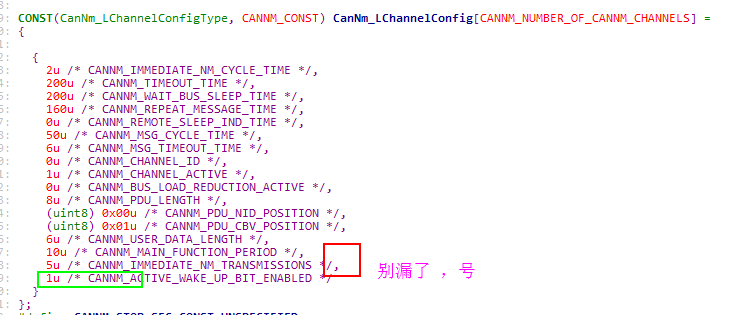
**12. 网络管理报文Control Bit**

12.1文件名：CanNm\_cfg.c

在 数组：CanNm\_LChannelConfig;增加

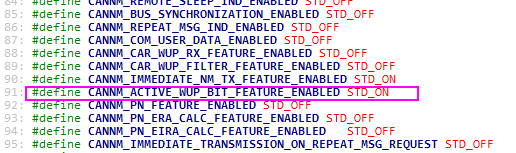
索引：CanNm\_LChannelConfig

,1u /\* CANNM\_ACTIVE\_WAKE\_UP\_BIT\_ENABLED \*/



12.2 文件：CanNm\_cfg.h

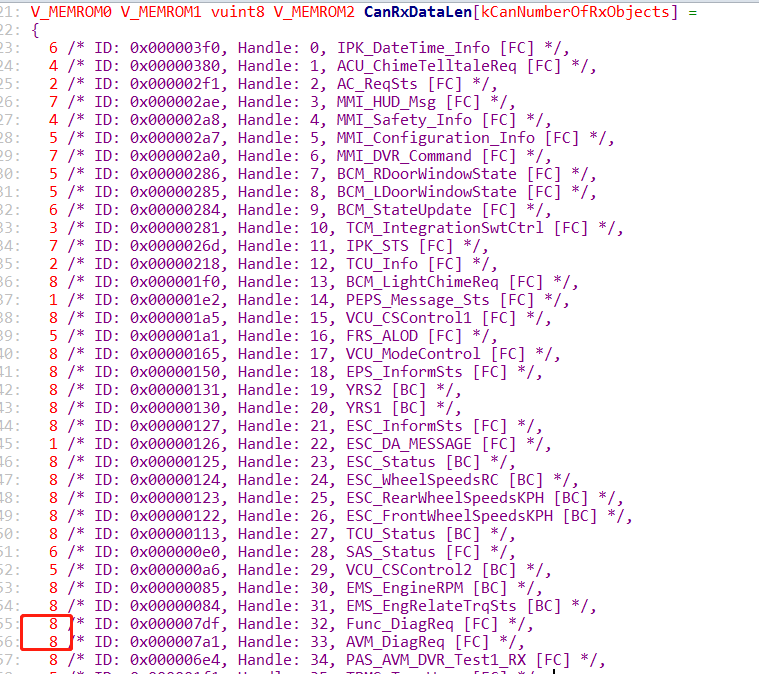
宏：CANNM\_ACTIVE\_WUP\_BIT\_FEATURE\_ENABLED 改成 STD ON



**13.Can\_par.c中**

将诊断请求报文的DLC长度更改为8.

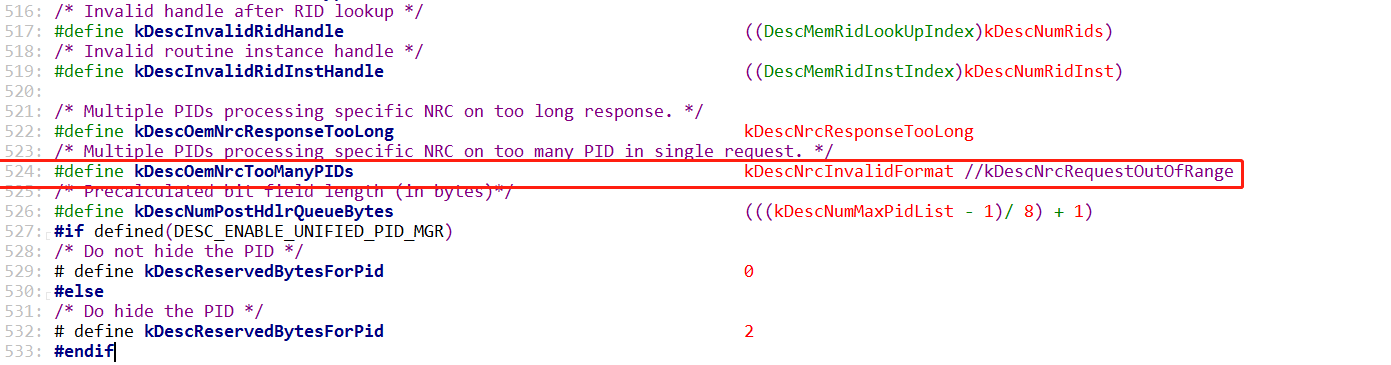
索引数组：CanRxDataLen



**14.desc.c文**件中，大于DID请求个数的话返回NC13，因此将kDescNrcRequestOutOfRange修改为kDescNrcInvalidFormat。

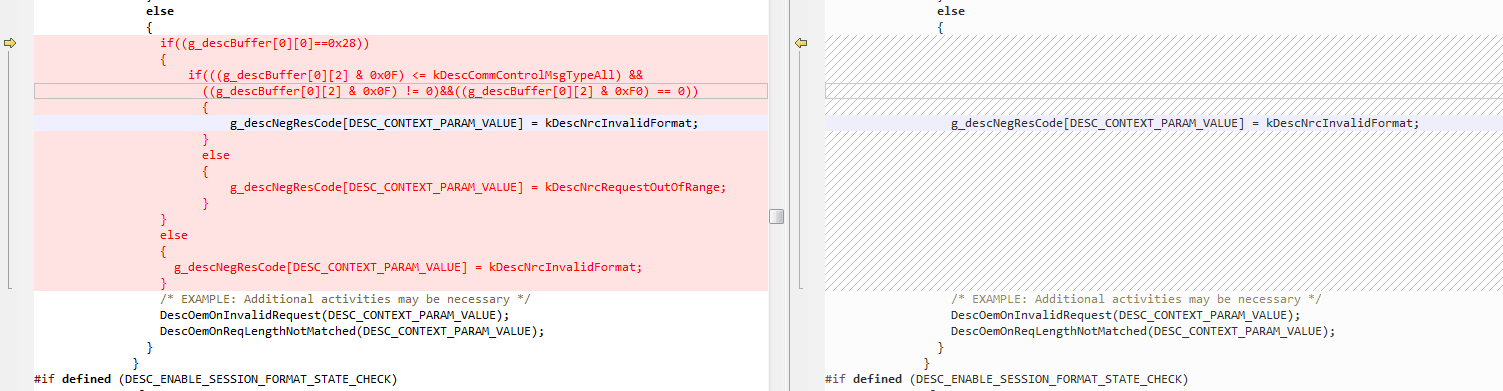
索引位置：kDescOemNrcTooManyPIDs

复制黏贴： kDescNrcInvalidFormat //kDescNrcRequestOutOfRange



第15条：desc.c文件中，修改28服务的NRC31优先级大于NRC13。

索引位置：DescDispatcher函数的4973行附件。对比修改如下：



复制黏贴：

if((g\_descBuffer[0][0]==0x28))

{

if(((g\_descBuffer[0][2] & 0x0F) <= kDescCommControlMsgTypeAll) &&

((g\_descBuffer[0][2] & 0x0F) != 0)&&((g\_descBuffer[0][2] & 0xF0) == 0))

{

g\_descNegResCode[DESC\_CONTEXT\_PARAM\_VALUE] = kDescNrcInvalidFormat;

}

else

{

g\_descNegResCode[DESC\_CONTEXT\_PARAM\_VALUE] = kDescNrcRequestOutOfRange;

}

}

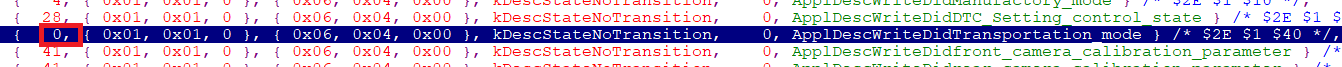
else

{

g\_descNegResCode[DESC\_CONTEXT\_PARAM\_VALUE] = kDescNrcInvalidFormat;

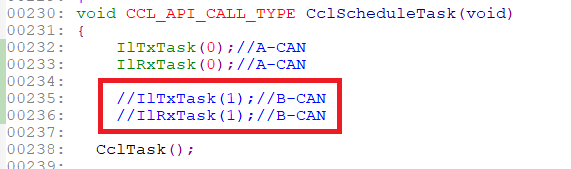
}

第16条：desc.c文件中，修改2E服务的NRC31优先级大于NRC13(long)。（13(short)>31>13(long)>33>22>72(ISO)）



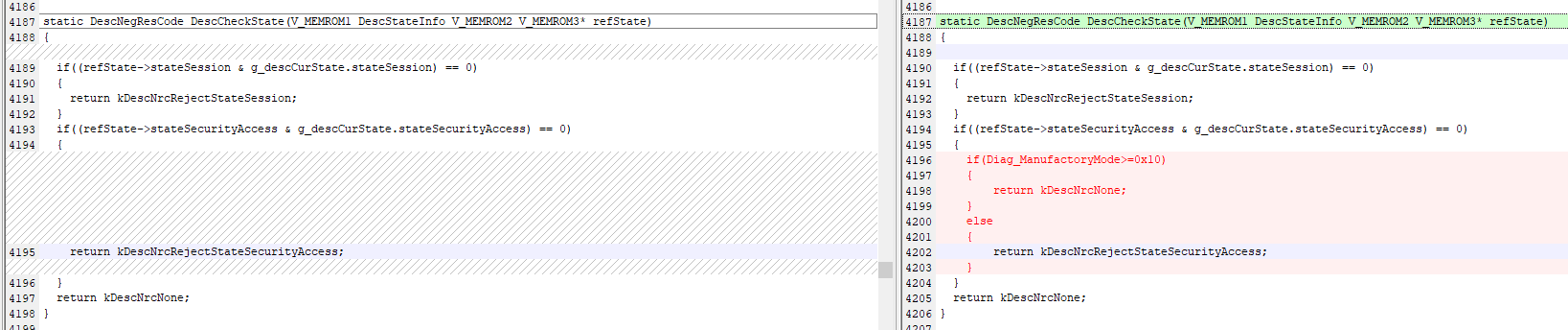
修改g\_descSvcInst表格中ApplDescWriteDidTransportation\_mode，设置成0；

第17条：ccl\_par.c文件中，屏蔽B-Can。改为单路Can。

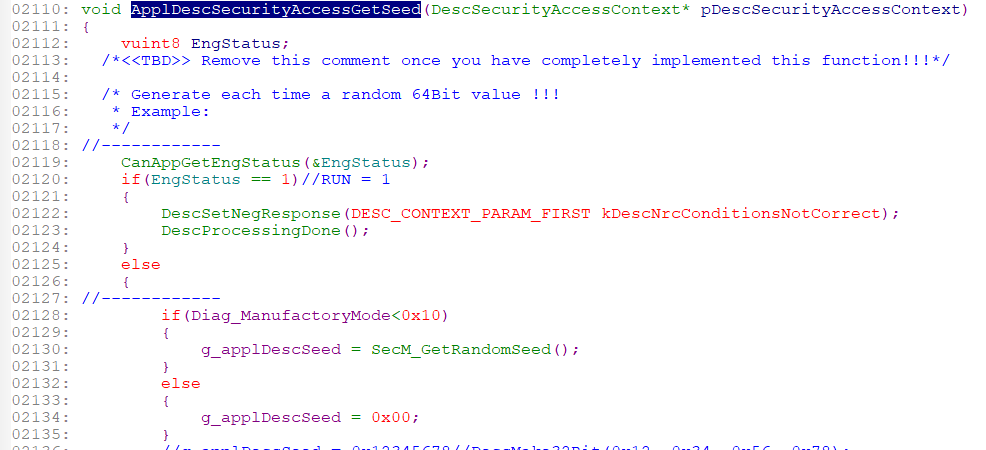


第18条：desc.c中，工厂模式默认安全解锁。

在DescCheckState（）函数中修改。参照右边修改。

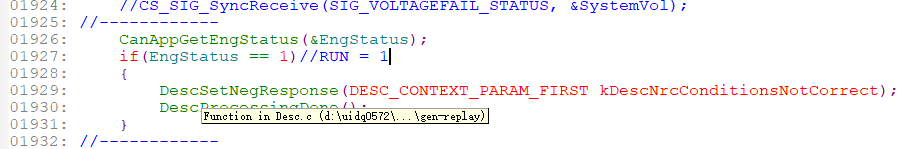


Appdesc.c中，函数ApplDescSecurityAccessGetSeed需要修改，如下截图。



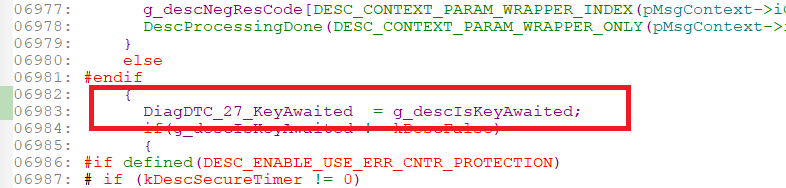
第19条：Appdesc.c中，回话时，EngStatus=run，回复NRC22；

修改函数ApplDescCheckSessionTransition；添加信号EngStatus判断；



第20条： desc.c中，在static void DescOemSecuritySeed(DescMsgContext\* pMsgContext, DescStateGroup requestedSecurityLevel) 函数中，

添加 DiagDTC\_27\_KeyAwaited = g\_descIsKeyAwaited;



第21条： desc.c中，添加函数void DescSecurityOnceInitPowerOn\_GAC(void)解决GAC的安全解锁3次后，10s计时结束，再钥匙OFF切换到On状态，需要清除安全解锁次数。

void DescSecurityOnceInitPowerOn\_GAC(void)

{

if(g\_descSecureNrOfAttempts==2)

{

g\_descSecureNrOfAttempts = 0;

}

}

第22条： desc.c中，DescSecurityOnceInitPowerOn和DescSecurityAccessTimer函数中，添加逻辑g\_descSecureTimer上电/复位delaytime，然后delaytime结束g\_descSecureNrOfAttempts设置为0。

