文章编号:1671-4598(2017)10-0217-04 DOI:10.16526/j.cnki.11—4762/tp.2017.10.055 中图分类号:TP273 文献标识码:A

基于 AUTOSAR 的点到点安全通信的实现

钟再敏，黄 熙，章鸿滨
（同济大学 汽车学院，上海 201804）

摘要：功能安全的概念在汽车嵌入式系统领域越发到关注，汽车开放系统架构 AUTOSAR（Automotive Open System Architecture）是目前国际流行的标准软件架构，它在 AUTOSAR4.1 的版本中针对功能安全首次提出了点到点（End—to—End，E2E）的安全通信机制；为保证汽车各组件间的通信安全，对在 AUTOSAR 架构下的 E2E 安全通信机制进行了研究，采用 E2E Profile 2 的方法来实现 E2E 安全通信，旨在解决如何保证电子控制单元（Electronic Control Unit，ECU）之间以及 ECU 内部不同核之间，不同 SWC（software component）之间数据的安全通信的问题；基于 AUTOSAR 架构，通过在电子控制单元核内通信采用 E2E Protection Wrapper 的通信方式，跨电子控制单元核外通信采用 COM E2E Callout 的通信方式实现了通信机制的搭建；通过对 ECU 内部及跨 ECU 的通信测试，表明该方法能有效的检测通信过程中的重复发送错误、CRC（Cyclic Redundancy Check）校验和错误及发送序列错误等问题。

关键词：点到点通信；AUTOSAR；功能安全；ECU 核内通信；跨 ECU 通信

Realization of E2E Secure Communication Based on AUTOSAR

Zhong Zaimin，Huang Xi，Zhang Hongbin
（School of Automotive Studies，Tongji University，Shanghai 201804，China）

Abstract：The concept of functional safety has attracted more and more attention on the automotive embedded systems. The automotive open system software architecture standard AUTOSAR is the current international popular standard，it firstly proposes a End—to—End (E2E) security communication mechanism for functional security in the AUTOSAR4.1. In order to ensure the safety of communication between components of the vehicle，the E2E security communication mechanism under the framework of AUTOSAR is studied. By using E2E Profile 2 methods to realize the secure communication of E2E to solve the problem of how to ensure the secure communication between ECU or different SWC inside ECU. By using the E2E Protection Wrapper communication mode of the ECU nuclear communication and the COM Callout E2E mode of the ECU core communication，AUTOSAR architecture is built based on this communication mechanism. By testing the communication，the proposed method can effectively detect the repeated transmission errors，CRC checksum errors and send sequence errors .

Keywords：E2E；AUTOSAR；functional safety；ECU kernel communication；cross—ECU communication

0 引言

车载网络的安全通信一直倍受关注，早有主机厂和零部件供应商在车载安全通信方面做了探索和实现，普遍采用的方式是添加循环冗余校验的信息，信息在传递过程中一旦被篡改可以在信息接收端被识别，保证错误的信息不会被系统使用。但是这种方式，只能识别出信息被篡改，却无法发现信息的丢失和重复发送，因此后面加入了计数器信息，用于标识当前信息的序列号，在信息的接收端就可以发现信息的丢失和重复发送。不同的厂家之间安全通信的实现方式各不相同，因此不同厂家实现的控制器在同一个通信网络中通信存在兼容性问题。汽车开放系统架构标准 AUTOSAR，采用分层的软件架构，规范通信接口，可实现软硬件的分离，可极大提高软件的可移植性和可复用性[1-2]。在 AUTOSAR4.1[3-6] 的版本后陆续提出和完善了点到点（E2E）的安全通信机制，其囊括了多种的保护方式，并提出了安全通信标准的实现方式，不仅解决了控制器安全通信的问题，同时依托 AUTOSAR 软件架构可以保证软件的可移植性和厂家合作之间的兼容性。

本文采用 E2E Profile 2 通信安全机制，首先介绍了软硬件导致的常见的通信失效类型，提出 E2E 安全通信的必要性，其次介绍了 E2E 安全通信保护机制，之后介绍了 AUTOSAR 架构下 E2E 安全通信的具体实现，最后进行测试和验证。

收稿日期：2017-04-09； **修回日期**：2017-04-26。

基金项目：国家科技支撑计划(2015BAG17B00)；国家科技支撑计划(2015BAG03B00)。

作者简介：钟再敏(1973-)，男，博士，教授，主要从事车用电驱动控制技术方向的研究。

1 软硬件导致的通信失效类型

在 AUTOSAR 中 E2E 的应用场景相对集中，E2E 的实施是为了保证 ECU 之间以及 ECU 内部不同核之间，不同 SWC 之间数据的安全通信，即在系统运行过程中，动态地识别软硬件故障引起通信失效，保证系统的可靠运行。E2E 安全通信是点到点之间的检测，它可以发现点到点之间由于软硬件原因导致的通信失效。图 1 列举了 AUTOSAR 架构下 E2E 的几个应用场景。

图 1 中 S1～S4 代表的是软件导致的通信失效：S1 指的是 RTE 生成代码出错；S2 指的是 COM 服务层代码出错；S3 指的是通信接口层和通信的驱动层之间可能存在问题，出错原因与 S2 的原因类似；S4 指的是跨核通信中 IOC 出错。这 4 种通信失效的场景都可以采用 E2E 安全通信进行检测。图 1 中 H1—H3 代表的是硬件导致的通信失效：H1 指的是通信的物理网络存在故障；H2 指的是通信网络的接口存在电磁兼容性问题；H3 指的是微控制器故障。这 3 种通信失效的场景同样可以采用 E2E 安全通信进行检测。

2 E2E 安全通信保护机制

本文在 AUTOSAR 架构下采用 E2E Profile 2 来实现 E2E 安全通信。E2E Profile 2 包含 3 个保护方式：分别是循环冗余

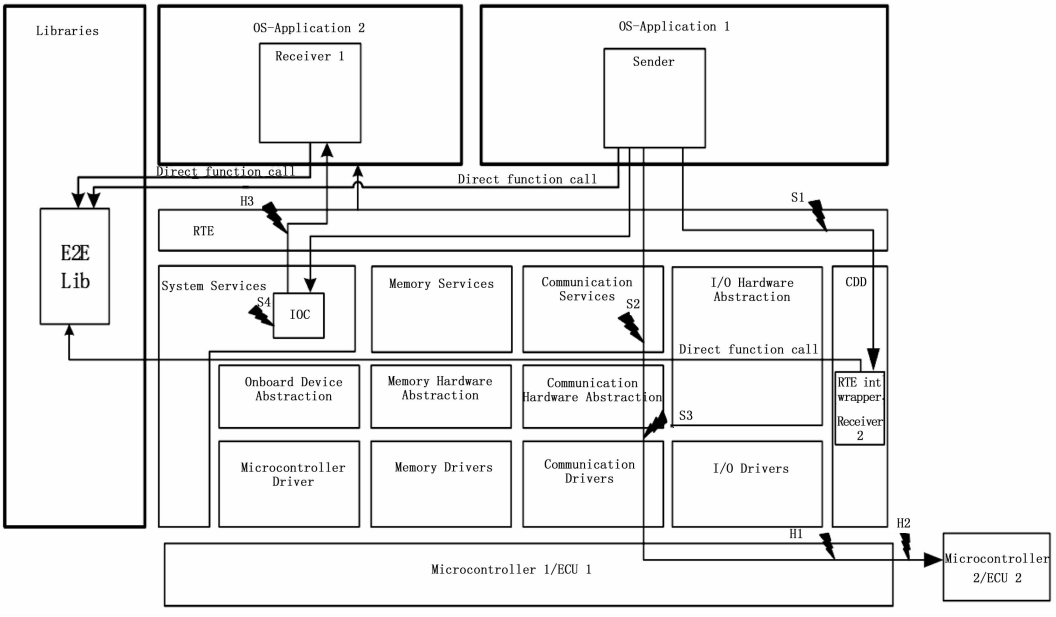

图 1 AUTOSAR 架构下 E2E 的应用场景

校验，计数器和数据 ID。通过 E2E Profile 2 处理后的数据 Buffer 分布如图 2 所示。E2E Profile 2 规定 Data［0］必须存放 CRC 校验和，Data［1］的低 4 个位必须存放 Sequence Counter。从 Data［2］开始存放需要保护的数据。

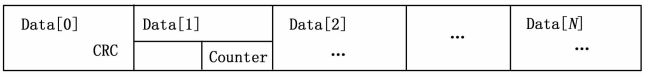

图 2 E2E Profile 2 处理后的数据 Buffer

其中参与 CRC 校验和计算的字段可以根据用户配置而定，我们采用的是下图 3 的方式，校验和包含 Data ID 部分和排除 CRC 存储字段外的所有的数据，虽然 Data ID 并没有显性的通过报文发送出去，但是它的校验信息包含在 CRC 中，其中需要填充的部分统一填充为 0xF。

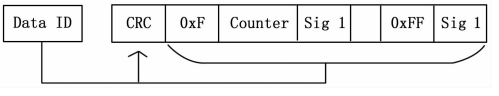

图 3 CRC 校验和的计算方式

采用 E2E Profile 2 安全通信需要额外的引入两个数据元素，分别是 Sequence Counter 与 CRC 校验和，它们必须随着被保护数据一起从发送端传送到接收端。在进行软件组件接口定义的时候必须定义相关的数据元素，尤其是在跨 ECU 通信中还必须在 DBC 中预留出相关的信号用于传输 Sequence Counter 与 CRC 校验和。

E2E Profile 2 的发送端和接收端，分别维护着一个 E2E 发送端的状态机和一个 E2E 接收端的状态机。在发送端，E2E 会根据发送端的状态机，首先计算 Sequence Counter 并写入数据 Buffer，然后根据被保护的数据、静态配置好的数据 ID 列表以及 Sequence Counter，计算 CRC 校验和，并写入发送 Buffer。完成上述操作后，E2E 就返回数据 Buffer 给调用者，最终由调用者将 CRC 校验和、Sequence Counter 与被保护的数据一起发送出去。被保护的数据并不会被 E2E 改变，E2E 并不负责加密。

在接收端，E2E 根据接收端状态机先计算出预想的 Sequence Counter，再根据静态配置好的数据 ID 列表、Sequence Counter 和接收到的数据，计算出 CRC 校验和。这里接收到的数据仅仅指发送端发送的需要被保护的数据。计算出 CRC 校验和后，E2E 会与发送端发过来的 CRC 校验和进行比较，如果一致，则继续判断 Sequence Counter 是否与期望的 Sequence Counter 一致。如果校验结果完全正确或者在容忍的范围内，则 E2E Profile 2 返回正确的返回值；如果校验结果有问题，则需要进一步解析出错的类型，并将错误状态更新到 E2E 的状态机。E2E 只做数据准确性的校验，至于出现数据丢失或者重复，应用层是否采用此数据，那是应用层数据访问的策略，与 E2E 无关。

在 AUTOSAR4.2.2 标准中，E2E 的通信有 3 种不同的实现方式：第一种是实现方式 E2E Transformer；第二种实现方式是 E2E Protection Wrapper；第三种实现方式是 COM E2E Callout。本文中，同一个 ECU 内部的 SWC 之间的 E2E 通信采用 E2EPW 的实现方式，跨 ECU 的 E2E 通信采用 COM E2E Callout 的实现方式。在跨 ECU 的通信中，E2E 通信添加的控制数据需要通过 CAN 网络进行传输，故需要修改 DBC 文件，对需要保护的信号添加相应的校验和字段和序列号字段。

3 AUTOSAR 架构下 E2E 安全通信的实现

3.1 应用层架构设计

应用层开发的流程和需求主要包括以下六步，可参看图 4：第一步，DataType 的定义；第二步，Interfaces 的定义；第三步，SWC 的定义；第四步，System 的定义；第五步，ECU Extract 的生成；第六步，OS 与 RTE 的配置[7-9]。本文只介绍和 E2E 通信相关的配置项。

1）Interface 定义

Interface 定义如表 1 所示，在同一个 ECU 内部的两个 SWC 通信引用的接口为 E2E _ DataInterface _ 0。在跨 ECU 通信中，将 CAN 的发送和接收接口定义为 E2E _ Can _ Input 和

E2E _ Can _ Output。它们均为 S－R 端口，内部包含 3 个数据元素，分别为被保护的数据、序列号 Counter 及 CRC 校验和。

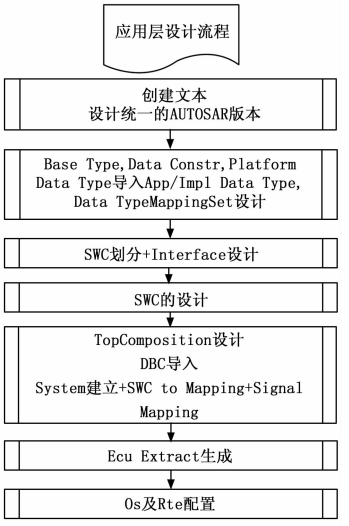

图 4　应用层的开发流程和需求

2）SWC 的定义

本文应用层需要在 ECU 内部实现两个 SWC 之间的 E2E 通信，这两个 SWC 分别是 CPT _ E2E _ InterSWC _ 1（作为发送 SWC）和 CPT _ E2E _ InterSWC _ 2（作为接收 SWC）。

表 1　Interface 定义

接口名称	数据名称	数据类型
E2E_DataInterface_0	CKS	Unit8
	Alive_Cnt	Unit8
	DATA	Unit8
E2E_Can_Input	DE_Signal_Rx_CRC	Unit8
	DE_Signal_Rx_Counter	Unit8
	DE_Signal_Rx_Data0	Unit16
E2E_Can_Output	DE_Signal_Tx_CRC	Unit8
	DE_Signal_Tx_Counter	Unit8
	DE_Signal_Tx_Data0	Unit16

与此同时，还有一个 SWC 与其他 ECU 的 SWC 通过 CAN 网络进行 E2E 通信，即将 CPT _ E2E _ SWC 这个 SWC 的接收发送端口上的数据 MAPPING 到 CAN 网络的信号上。

SWC 的定义如表 2 所示。以发送端 SWC 功能定义为例，E2E _ InterSWC _ 1 包含一个 PPort 端口和一个内部行为。内部行为包含两个运行实体，其中一个运行实体关联一个周期性的事件运行。另外一个是初始化的运行实体。此外还需要为周

表 2　SWC 的定义

SWC	Port	Runnable
E2E_InterSWC_1	PPortPrototype_0	RE_E2E_InterSWC1
		RE_E2E_InterSWC1_Init
E2E_InterSWC_2	RPort_E2E_DataInterface_0	RE_E2E_InterSWC_2
		RE_E2E_InterSWC2_Init
E2E_SWC	RPort_E2E_Can_Input	RE_E2E_SWC
	PPort_E2E_Can_Output	RE_E2E_SWC_Init

期性运行的可运行实体添加数据访问点，用于发送数据。其余两个 SWC 定义类似。

3.2　基础软件层及 OS 配置

应用层架构设计工具与基础软件配置工具进行信息交互的文件是 ARXML 文件，主机厂只要把前一阶段生成的 ECU Extract of System Description 交付给一级供应商，一级供应商就可以进行底层基础软件的配置。基础软件层代码库及其配置工具来自某商用公司开源的 AUTOSAR 基础软件层解决方案。该部分在 AUTOSAR 软件架构 RTE 层的下面，MCAL 层的上面。基础软件层配置后，需要将基础软件层配置的 XML 文件拷贝到 AUTOSAR 应用层配置软件中，为接下来的 OS 配置和 RTE 配置做好准备[8-10]。该部分本文不作介绍。在应用层配置软件中可进行操作系统的配置，用户可以根据需要将不同的 SWC 的 runnable 和基础软件层的 MainFunction 映射到操作系统的任务中[11]。

3.3　芯片驱动设计

芯片驱动设计目的是使上层软件与微处理器型号无关，包含 MCU 中内部外设的驱动和使用 MCU 内存映射的外部设备的驱动。本部分开发工作主要面向控制器抽象层（Microcontroller Abstraction），属于 AUTOSAR 软件架构的最底层，包括 Microcontroller Driver、Memory Driver、Communication Driver 以及 I/O Driver 等。具体需要配置的模块有 MCU、CAN、Port。Mcal 模块配置完成就可以生成相关的配置代码，它是以源文件的形式生成。除了配置代码，还需要配置 Mcal 的静态代码，二者配合才能集成编译。

3.4　基于 E2EPW 通信保护的代码实现

发送端的软件组件 E2E _ InterSWC _ 1 内部的可运行实体 E2E _ InterSWC _ 1 _ REProc 周期性被操作系统任务 AppTask _ 10ms 调度，如下所示：

```
FUNC(void,E2E_InterSWC_1)E2E_InterSWC_1_REProc(void)
{
if(E2E_TestActive==0)
{
E2EPW_error_SWC1=E2EPW_Write_E2E_InterSWC_1(&E2E_InterSWC_Var1);
}
}
```

函数 E2EPW _ Write _ E2E _ InterSWC _ 1 的参数是一个结构体指针，该结构体的类型为应用层配置软件中 E2EImplementationDataType _ 0，如下所示：

```
typedef struct {
    uint8 Impl_CKS;
    uint8 Impl_Alive_Cnt;
    uint8 Impl_Data;
} E2EImplementationDataType_0;
```

在函数 E2EPW _ Write _ E2E _ InterSWC _ 1 内部调用 E2E _ P02Protect 函数，进行发送端 E2E 控制数据（CRC，Counter）的计算，之后通过 RTE 提供的 API 函数，将控制数据和需要保护的数据发送给接收端 SWC，如下所示：

```
FUNC(unit32,E2E_InterSWC_1)E2EPW_Write_E2E_InterSWC_1
(E2EImplementationDataType_0 * E2E_InterSWC_var1)
{unit32 E2EPW_error=E2E_E_OK;
static boolean FirstRun=TRUE;
Std_ReturnType E2E_error;
E2E_error=E2E_P02Protect(&E2E_P02_StaticCfg_SWC1,&E2E_P02SenderState_SWC1,E2E_InterSWC_var1);
```

```
Rte_Write_E2E_InterSWC_1_PPortPrototype_0_Alive_Cnt(E2E_InterSWC_var1->Impl_Alive_Cnt);
Rte_Write_E2E_InterSWC_1_PPortPrototype_0_CKS(E2E_InterSWC_var1->Impl_CKS);
Rte_Write_E2E_InterSWC_1_PPortPrototype_0_Data(E2E_InterSWC_var1->Impl_Data);
return((RTE_E_OK)|(E2EPW_error<<8)|(E2E_error<<16));
}
```

接收端软件组件 E2E _ InterSWC _ 2 代码原理与发送端类似，可运行实体 E2E _ InterSWC _ 2 _ REProc 的内部通过函数 E2EPW _ Write _ E2E _ InterSWC _ 2 调用 RTE 提供的 API 函数来读取控制数据和 E2E 保护的数据，再调用 E2E _ P02Check 函数来校验接收的数据（CRC，Counter），最后将校验结果返回。此外，在 ECU 的启动时，需要在 E2E 安全通信之前初始化发送端和接收端 E2E 状态的结构体，如下所示。

```
FUNC(Std_ReturnType, E2E_InterSWC_1) E2EPW_WriteInit_E2E_InterSWC_1(void)
{Std_ReturnType
E2EPW_error=E2E_E_OK;
E2E_P02SenderState_SWC1.Counter=0;
return E2EPW_error;
}
FUNC(Std_ReturnType,E2E_InterSWC_1)E2EPW_ReadInit_E2E_InterSWC_2(void)
{ Std_ReturnType E2EPW_error=FALSE;
E2E_P02ReceiverState_SWC2.LastValidCounter=0;
E2E_P02ReceiverState_SWC2.LostData=0;
E2E_P02ReceiverState_SWC2.MaxDeltaCounter=0;
E2E_P02ReceiverState_SWC2.NewDataAvailable=FALSE;
E2E_P02ReceiverState_SWC2.WaitForFirstData=TRUE;
E2E_P02ReceiverState_SWC2.Status=E2E_P02STATUS_NONEWDATA;
E2EPW_error=TRUE;
return E2EPW_error;
}
```

3.5 基于 COM E2E Callout 通信保护的代码实现

首先在基础软件配置工具中将 PDU 的发送和接收的回调函数配置好，会生成一个函数指针，并声明该函数具有的参数和返回值类型。

Com _ RxCbk _ msg _ RxCycle500 _ 20 _ E2E 和 Com _ RxCbk _ msg _ TxCycle500 _ 20 _ E2E 分别为发送和接收的回调函数。通过基础软件层配置工具生成配置代码，其只是声明了回调函数的形式，具体回调函数的定义需要用户自己完成。

下面介绍 COM 端的代码实现：发送端在 PDU 发送回调函数中直接调用 E2E _ P02Protect 函数，如图 10 所示。

```
FUNC(boolean,COM_CODE)Com_RxCbk_msg_TxCycle10_0_E2E(VAR(PduIdType,AUTOMATIC)id,P2VAR(unit8,AUTOMATIC,COM_APPL_DATA)ptr)
{
Std_ReturnType error,temporary_return;
if(E2E_ProfileSelector==0x02){
error=E2E_P02Protect(&E2E_P02_StaticCfg_msg_TxCycle10_0_E2E,&E2E_P02_SenderState_msg_TxCycle10_0_E2E,ptr);
count_TX_P02++;
if(error==0){
temporary_return=TRUE;
}
else{
temporary_return=error;
}
return temporary_return;
}
```

接收端在 PDU 接收回调函数中直接调用 E2E _ P02Check 函数，如下所示：

```
FUNC(boolean,COM_CODE)Com_RxCbk_msg_RxCycle500_20_E2E(VAR(PduIdType,AUTOMATIC)id,P2CONST(unit8,AUTOMATIC,COM_APPL_CONST)ptr)
{Std_ReturnType error,temporary_return;
if(E2E_ProfileSelector==0x02){
E2E_P02ReceiverState_msg_Rxcycle500_20.NewDataAvailable=TRUE;
error=E2E_P02Check(&E2E_P02_StaticCfg_msg_RxCycle500_20_E2E,&E2E_P02_ReceiverState_msg_RxCycle500_20_E2E,ptr);
count_RX_P02++;
if((error==0)&&((E2E_P02ReceiverState500_20.Status==E2E_P02STATUS_OK)||
(E2E_P02ReceiverState500_20.Status==E2E_P02STATUS_OKSOMELOST)||
(E2E_P02ReceiverState500_20.Status==E2E_P02STATUS_INITIAL))){
temporary_return=TRUE;
}
else{
temporary_return=FALSE;
}
```

4 测试及验证

4.1 基于 E2EPW 实现 ECU 内部通信测试

根据上文建立的模型，通过实验测试 ECU 内部 SWC 之间的 E2E 通信，对于可能出现的错误的通信进行模拟。我们通过在发送端 SWC 发送数据时注入错误，查看接收端 SWC 的 E2E 状态结构体，看 E2E 的通信检测机制是否可以检测出相应的错误。

E2E _ TestActive 是一个全局变量，我们添加在 TRACE32 上位机中作为标定量，表示程序是否进入测试。E2E _ TestCase 也是一个全局变量，我们将其作为标定量，表示不同的测试案例。此外，还需要在 TRACE32 上位机中添加一个观测量 E2E _ P02ReceiverState _ SWC2，该变量是接收端 E2E 的状态结构体，存放当前数据校验的结果。测试的流程如图 12 所示。

首先测试正常通信，使 E2E _ TestActive=0，此时状态位为 E2E _ P02STATUS _ OK，表示通信正常。之后每次测试前应保证通信正常。使 E2E _ TestActive=1，此时状态位为 E2E _ P02STATUS _ REPEATED，表示发生重复发送错误。使 E2E _ TestActive=2，此时状态位为 E2E _ P02STATUS _ WRONGCRC，表示 CRC 检验和出错。使 E2E _ TestActive=3，此时状态位为 E2E _ P02STATUS _ WRONGSEQUENCE，表示发送序列出错。

（下转第 243 页）

相应的授权操作，有效实现了对不同用户的访问行为进行控制的目的。总结而言，该方案不但可以有效实现访问控制，同时能够极大提高系统效率，具有很好的发展前景。

参考文献：

[1] 赵雪良. 电力云存储中基于属性和策略的访问控制研究 [D]. 保定：华北电力大学，2014.

[2] 甘玉芳. 面向智能电网云存储的基于属性角色的访问控制研究 [D]. 保定：华北电力大学，2015.

[3] 罗　超. 基于云存储的智能配用电系统及其访问控制方法研究 [D]. 保定：华北电力大学，2014.

[4] 张鸿辉，刘　伟，李永强. 应用于电网企业的云存储访问控制增强策略 [J]. 计算机应用与软件，2014 (2)：17－20.

[5] 冉　军. 基于云存储的智能电网访问控制研究 [D]. 保定：华北电力大学，2014.

[6] 关志涛，杨亭亭，徐茹枝，等. 面向云存储的基于属性加密的多授权中心访问控制方案 [J]. 通信学报，2015，36 (6)：116－126.

[7] 明　镜. 智能配电网云存储中基于属性的访问控制研究 [D]. 保定：华北电力大学，2015.

[8] 厉优栋. 基于电力系统统一平台的访问控制机制的研究与实现 [D]. 上海：上海交通大学，2012.

[9] 王保义，王蓝婧. 电力信息系统中基于属性的访问控制模型的设计 [J]. 电力系统自动化，2007，31 (7)：81－84.

[10] 王保义，邱素改，张少敏. 电力调度自动化系统中基于可信度的访问控制模型 [J]. 电力系统自动化，2012，36 (12)：76－81.

[11] 张　强，刘雪艳，王维洲，等. 基于 CP－ABE 的智能电网访问控制研究 [J]. 计算机工程，2014，40 (12)：83－88.

[12] 夏明超，吴俊勇，吴命利. 基于角色访问控制在电力监控系统中的应用 [J]. 电力系统及其自动化学报，2008，20 (2)：46－50.

[13] 孙中伟，张荣刚. 智能配电网通信系统访问控制研究 [J]. 电力系统保护与控制，2010，38 (21)：118－121.

[14] 宋燕敏，杨争林，曹荣章，等. 电力市场运营系统中的安全访问控制 [J]. 电力系统自动化，2006，30 (7)：80－84.

[15] 孙中伟，张荣刚. 智能配电网通信系统访问控制研究 [J]. 电力系统保护与控制，2010，38 (21)：118－121.

[16] 丁　杰，奚后玮，韩海韵，等. 面向智能电网的数据密集型云存储策略 [J]. 电力系统自动化，2012，36 (12)：66－70.

(上接第 220 页)

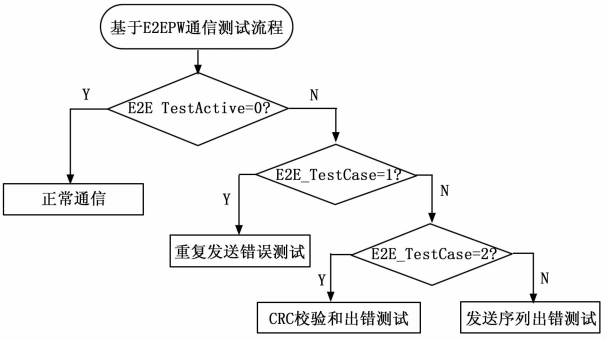

图 12　基于 E2EPW 的 E2E 通信的测试流程图

4.2　基于 COM E2E Callout 实现跨 ECU 通信测试

由于条件限制，本文通过开发工具模拟一个节点，捕捉开发板发送出来的 CAN 报文，修改 CAN 报文的 ID，然后重新发给开发板。通过这样的方式，模拟跨 ECU 之间的通信。

4.2.1　E2E 跨 ECU 通信测试

1）常通信测试：在 TRACE32 中添加观测量 E2E _ P02ReceiverState _ msg _ RxCycle500 _ 20，这是 COM 层上 E2E 通信的状态结构体，其内部保存对最新接收到的 PDU 的校验结果。用 CANoe 和开发板配合构造一个闭环的通信过程来测试 E2E 跨 ECU 通信是否正常工作。此时接收端状态结构体中 Status 为 E2E _ P02STATUS _ OK，表示接收端校验正常，Rte _ ImplicitBufs 会跟随 CAN 报文内容动态变化。

2）异常通信测试：本文通过构造一个开环的通信过程测试 E2E 的检错能力，将 CAN 卡和开发板连接，此时上位机不对接收到的报文做处理，而是将我们自定义的报文发给开发板。发送端的 Counter 是 0－15 循环，如果我们通过 E2E 保护的数据没有动态改变，则 CAN 报文在每个循环中只要 Counter 相同，那 CAN 报文的内容完全一致。首先保证接收端校验正常，再人为的在报文中注入我们想要的错误来测试不同的错误。连续发送两帧 Counter 为 2 的 CAN 报文来测试重复发送错误，此时接收端状态结构体中 Status 为 E2E _ P02STATUS _ REPEATED。改变 Counter 为 2 的 CAN 报文的内容，使其 CRC 校验和出错，然后发送给开发板，此时接收端状态结构体中 Status 为 E2E _ P02STATUS _ WRONGCRC。改变发送序列，在本应该发 Counter 为 3 的时候，发送 Counter 为 4 的给开发板，接收端状态结构体中 Status 为 E2E _ P02STATUS _ OKSOMELOST。

5　结论

本文在 AUTOSAR 架构下，采用 E2E Profile 2 实现了 E2E 安全通信。介绍了基于 E2E Profile 2 实现 E2E 安全通信的原理。在 ECU 内部通信采用 E2E Protection Wrapper 的通信模式，在 ECU 跨核通信采用 COM E2E Callout 的通信模式，描述了这两种通信模式的搭建过程，并搭建实验来模拟在这两种模式下的通信错误，实验表明，该方法能快速有效的检测通信过程中的重复发送错误、CRC 校验和错误及发送序列错误等问题，保障汽车的通信安全。

参考文献：

[1] 魏学哲，戴海峰，孙泽昌. 汽车嵌入式系统开发方法、体系架构和流程 [J]. 同济大学学报（自然科学版），2012，40 (7)：1064－1070.

[2] 高焕吉. 基于 AUTOSAR 的汽车电子控制系统嵌入式软件开发 [J]. 汽车电器，2010 (05)：11－14.

[3] AUTOSAR Administration. Specification of SW－C End to－End Communication Protection Library V4.2.2 [EB/OL]. http：//www.autosar.org/.

[4] AUTOSAR Administration. Specification of Module E2E Transformer V4.2.2 [EB/OL]. http：//www.autosar.org/.

[5] AUTOSAR Administration. Specification of Communication V4.2.1 [EB/OL]. http：//www.autosar.org/.

[6] Moessinger J. AUTOSAR－The Standard for Global Cooperation in Automotive SW Development [Z]. 2008.

[7] Voget S. AUTOSAR and the Automotive Tool Chain [C]. 2010.

[8] AUTOSAR Administration. Specification of Operating System V5.0.0 [EB/OL]. http：//www.autosar.org/.

[9] ETAS GmbH. RTA－OS User Guide [Z]. 2014.

[10] AUTOSAR 基础软件介绍 [Z]. 2013.

[11] 秦美峰. AUTOSAR 软件构件运行实体映射模型的研究与设计 [D]. 成都：电子科技大学，2012.