针对联网汽车的可实现无线攻击和车载局域网的安全协议

摘要：采用电子控制单元(ECU)控制车辆电气系统的车载信息融合技术是现代车辆快速兴起的一种模式，而控制器区域网络(Can)作为车载网络，则用于构建高效的车辆安全网络。不幸的是，控制器区域网络没有适当地处理安全问题，尽管它的控制信息可能会导致致命的危险。随着汽车联网的环境的出现，车载网络(例如 CAN)现在连接到外部网络(例如，3G/4G 移动网络)，使得对手能够使用 CAN 漏洞进行远程无线攻击。在本文中，我们证明了在联网的汽车环境中使用恶意的智能手机对真正的车辆进行远程无线攻击是可能的，并提出了一种针对当前 can 规范的安全协议，并使用了 CANoe 软件和 dsp-f28335 微控制器对所提出的安全协议的可行性进行了评估，结果表明我们所提出的安全协议与现有的安全协议相比，在认证延迟和通信负载方面表现地更好。

关键词：联网的汽车 控制器区域网络(CAN) 车载网络安全 密钥管理

1、介绍

为用户提供舒适的驾驶环境，响应汽车排放法规，最新车型的汽车继续追求与各种电控技术的融合。为了将电控技术应用于汽车，必须使用大量的汽车应用部件。在这些部件中，电子控制单元(ECU)是控制一个或多个电气系统和汽车的最基本部件。在车辆中的子系统中，最先进的车载体系结构可以由 70 多个 ECU 组成。它们通过诸如控制器区域网络(CAN)、本地互连网络(LIN)或 FlexRay 等异构通信网络互连。

作为最具代表性的车载网络，CAN 已经成为事实上的标准，因为它大大减少了所需通信线路的数量，并确保了更高的数据传输可靠性。不幸的是，在 CAN 的设计中没有考虑到信息的安全性，尽管传输的每一点信息都可能对司机的安全至关重要。例如，当数据通过总线网络广播时，无法确保 CAN 数据帧的保密和认证，从而为恶意对手轻易窃听数据或发起重放攻击铺平了道路。当车辆连接到汽车诊断工具上时，情况会变得更糟。为了在诊断过程中检查 ECU 的功能，诊断工具可以在没有加密和认证的情况下广播数据帧，以强制控制 ECU。这意味着对手也可以使用汽车诊断工具来轻松获得 CAN 数据帧，从而控制 ECU。

过去十年来，主要由欧洲资助的项目(例如 sevecom、preciosa、evita 和 overitor)积极开展了车辆安全研究。电子安全车辆入侵防护应用(Evita)项目明确规定了安全要求，并为车载车载网络开发了适当的解决方案。在这些解决方案中，evita-media-hsmm 是为了在 ecu 之间实现安全通信环境而开发的。然而，evita 并没有提供特定的安全解决方案。针对特定通信协议的体系结构。我们注意到，用于一般 it 环境的安全技术不能立即应用于 CAN，因为它具有独特的特性，例如有限的数据有效载荷。因此，即使在使用 Evita-Medi-HSM 的情况下，也有必要

设计一种高效的安全技术。

-[15]中的安全协议是在考虑 CAN 数据帧有限的数据有效载荷的情况下设计的，但是这些协议不适合部署在车辆环境中，因为它们不支持实时数据处理，也不考虑与外部设备(如汽车诊断工具)的连接。考虑到上述 CAN 的安全漏洞，将车载 CAN 连接到外部网络的联网汽车的出现使无线车辆攻击成为可能[16]。

本文演示了在汽车连网环境中，驾驶员的智能手机与车载设备相连时可实现的无线攻击。我们的攻击实验分为两个阶段：准备阶段和实际攻击阶段。在初始阶段，即在发动实际攻击之前，攻击者首先利用诊断工具获得一个 CAN 数据帧，以强制控制目标车辆。事实上，使用相同型号的车辆(更准确地说，汽车电子子系统配置相同的车辆)也可以。我们注意到，诊断工具用于获取能强制控制 ECUCAN 数据帧，并且在实际攻击时并不需要与目标车辆相连。攻击者也制造了一个恶意的自诊断应用程序，它伪装成一个正常的应用程序，并将其上传到应用程序市场。通过使用诸如“TORQUE”、“汽车量规 PRO”等自诊断应用程序和 OBD2 扫描工具(如“eml 327”、“plx kiwi”)，司机可以在驾驶时监视记录汽车局域网状态的信息。一旦目标车辆的司机下载了恶意的自我诊断应用，智能手机就在攻击者的控制之下。攻击者在实际攻击阶段，利用被感染的智能手机进行远程无线攻击，即如果 3G、4G 或 LTE 等移动网络可以通信的话，攻击者可以通过智能手机将数据帧独立注入到智能手机上。因此，智能手机不需要属于攻击者或机修师。我们提出的攻击模型只有在驾驶员下载恶意的自我诊断应用程序时才可能。即使在应用程序市场上大约有 300 个汽车自我诊断应用程序，它们在应用程序下载量上也不是最受欢迎的应用程序。这减少了基于我们提出的攻击模型的攻击的可能性。然而，我们提出的攻击模型仍然是一个现实的攻击场景，在不久的将来将是现实的，因为谷歌和领先的汽车制造商正在合作，将 Android 操作系统植入汽车和连网汽车服务，如镜像链接正在迅速增加。我们的攻击实验在第四部分中有详细的解释。

结合实际的无线攻击实验，我们还提出了一种安全协议来弥补 CAN 的漏洞，以满足以下要求。

- 数据加密和认证技术保证了车载 CAN 的实时数据处理。
- 使用消息认证码(Mac)的方法考虑了 CAN 数据帧的有限的数据有效载荷。
- 密钥管理技术支持外部设备和外部设备之间的安全连接。

此外，我们使用一个类似于实际 ECU 和一个常用的商业软件工具的制造安全-ECU 来评估所提出的安全协议的安全性和性能。以下是本文的主要贡献。

1） 演示了在联网汽车环境下使用恶意智能手机应用进行远程无线攻击的实际实验。

2） 设计了一种安全协议，该协议可以在 ECU 上实现，并能适应有限的可用资源和当前 CAN 数据帧格式。

3） 利用安全 ECU 和 CANoe 分析了所提出的安全协议的安全性和性能。

2、背景

A、控制器区域网络(Can)

CAN 是罗伯特·博世公司在 20 世纪 80 年代早期开发的一种高完整性串行数据通信技术，用于汽车应用之间的高效通信。CAN 是一种基于发件人 id 的多播广播通信总线系统，它允许 ECU 在数据传输速率达到 1Mb/s 的单线和双线网络上通信。CAN 协议使得汽车制造商可以降低车载网络布线的复杂性和成本，因此，国际标准化组织于 1993 将 CAN 作为国际标准[17]。

在 CAN 协议中，每个 ECU 使用数据帧向其他 ECU 传输信息。发送方 ECU 发送包含自己 id 的数据帧。其他 ECU 在识别数据帧中的发送方 ECU 的 ID 后选择性地检索数据帧。根据 ID 字段的长度，CAN 协议分为两种模式，即 CAN 2.0A 和 CAN 2.0B。在 CAN 2.0A 中，我们只描述了 CAN 2.0B，如图所示，CAN 2.0B 的数据帧格式如图所示。CAN 2.0B 有一个 29 位 ID 字段，分为两个部分：基础 ID 字段和扩展 ID 字段。ID 字段用于设置消息优先级。IDE 字段决定 18 位扩展 id 字段的使用。数据字段最多为 8 个字节，包含从发送方 ECU 发送到其他 ECU 的信息。循环冗余校验(CRC)字段用于传输数据帧的错误检测，其他字段与我们的工作无关，因此没有解释。

B、联网汽车环境

随着移动通信技术的飞速发展和智能设备及应用服务的普及，联网汽车作为电控汽车信息融合的下一代受到了广泛的关注。许多汽车制造商都在自主开发联网汽车技术，如通用汽车的 On Star 或 BMW 的联网驱动等。此外，随着按里程付费驾驶保险的普及，各种连接汽车的 OBD 2(车载诊断)端口，供智能手机应用使用的电子设备被大量销售。一般来说，联网的汽车是在驾驶时一直能连接到外部网络的车辆。如[16]和[18]中所述，联网汽车的组成部分如下：

- 具有电子控制单元（ECU）和车载网络的车辆；
- 为车辆提供各种服务的入口；
- 连接车辆和接口的通信链路。

附图.2 显示了一个联网环境的汽车包含的组件。在汽车中，大量电子控制单元被安装和连接到的汽车局域网上。入口可以划分为基于网页的和基于智能手机两种的应用服务。由于移动通信的高性能和普及，更多联网汽车环境是使用智能

手机。各种支持联网汽车的应用程序现在正在应用市场上销售，比如 google play 和苹果应用商店(例如， Send To Car, UVO Smart Control, and BMW ConnectedDrive)。

3、攻击模型和安全需求

在利用车载设备的漏洞方面，我们的攻击模型与 Koscher 等人的攻击模型相同[9]，[19]。然而，将恶意数据注入到入侵设备中的方法与其他相关的方法明显不同。我们的攻击模型是基于这样一个环境设计的：在车辆上安装 OBD 2 扫描工具之后，驾驶员使用自诊断应用程序监视状态信息，然后通过蓝牙将其与他/她的智能手机配对。当驾驶员在其智能手机上安装攻击者分发的恶意自诊断应用程序后，攻击者就可以发起实际攻击。攻击者可以从恶意自诊断应用程序中获取车辆的状态信息，并利用它将恶意数据注入车载网络。由于恶意自诊断应用程序和攻击者的服务器使用移动通信网络(如 3G、4G 或 LTE)进行通信，因此攻击不受距离的限制。此外，由于我们的攻击模型使用的是用于同一类模型的 ECU 强制控制数据(更准确地说，是使用用于同一模型、相同配置的汽车电子子系统)，所以没有必要事先实际占用目标车辆。

A、攻击模型

根据[8]所建议的计算机应急小组(CET)分类法，我们在表一中说明分类了我们提出的攻击模型。我们的攻击模型的假设如下：

攻击者的能力：在发动实际攻击之前，对手可以访问汽车诊断工具以获取 CAN 数据帧强制控制 ECU。攻击者可以使用恶意的自诊断应用程序窃听并将 CAN 数据帧注入到连接的汽车环境中的车载 CAN 网络。因此，攻击者不必在近距离攻击目标。应用程序可能会通过伪装成汽车的合法自我诊断应用在应用市场上广泛传播。

目标漏洞：目标车辆在电控单元之间使用 CAN 协议进行通信。如[3]、[7]、[8]和[9]所述，CAN 协议无法提供加密或数据帧验证等安全服务。这意味着可以在车载网络中实现窃听和重播攻击。未经授权使用汽车诊断工具也是一个安全漏洞，因为该工具存储了大量电子控制单元的控制命令。

受害者行为：目标车辆的受害者通过应用程序市场将恶意自诊断应用下载到他/她的智能手机上。受害者没有意识到该应用正在执行恶意行为，例如在车载 CAN 网络中进行窃听或重放攻击。在我们提出的攻击模型中，我们不认为这种

攻击会损害安装在车内的 ECU 或威胁到 ECU 的固件，因为这将需要长时间占据目标车辆以及专业的知识。

B、安全要求

以前的工作已经说明了在车辆上可能发生的各种类型的攻击。我们注意到，所有这些攻击最终都源自于车载CAN协议的漏洞，如第二节所讨论的。车载CAN协议有三个主要漏洞：1)薄弱的访问控制；2)无加密；3)不进行身份验证。为了构建安全的车载 CAN，必须消除这三个漏洞。然而，车载网络是一个几乎无法进行访问控制的通信环境。由于车载网络是基于发送者 ID 的多播广播通信环境，一个连接的节点可以接收发送的任何数据帧。此外，恶意节点可以通过窃取正常节点的 id 来传输数据帧(例如，数据帧的修改和重放攻击)。由于广播通信的性质在车载网络中进行访问控制实际上是不可能的，因此加密和验证数据帧，以防止修改和重放攻击是很有必要的。我们确认了提供安全的车载 CAN 协议要求如下：

保密性：CAN 中的每一个数据帧都应该被加密以提供保密性。也就是说，数据帧的明文应该只会被合法的 ECU 或它的同类可用。由于 CAN 的性质，所有数据帧都被广播，使得攻击者能够很容易地窃听 CAN 数据帧。特别是，可以通过汽车诊断工具获得强制控制 ECU 的数据帧，因此很容易分析相关数据帧的含义。

认证性：在 can 协议中，控制数据帧仅通过数据帧发送者的 ID 确认，这就会导致存在重放攻击的可能。这意味着，拥有完整控制数据帧的的攻击者可能会伪装成合法数据帧发送者。为了抵挡这种类型的攻击，协议应该同时提供认证和完整的数据。目前使用的 CAN 协议规定只使用 CRC 循环校验码来进行数据传输错误的校验，而不是拿来进行认证。

4、可实施攻击的实验

基于前面提到的攻击模型，这部分内容讲述了一个对汽车进行远距离攻击的实验方案以及我们的实验结果。我们的可实施无线汽车攻击方案分为两个阶段：准备阶段、实施攻击阶段。在实验中我们使用到的工具都列举在表 2 中。

A、准备阶段：

使用自诊断工具获得可以强制控制车辆的 CAN 数据帧：在【9】中 Koscher

指出了多种确认数据帧控制车辆的关键部分的技术。在他提到的这些技术中，使用车辆自诊断工具是最简单的，因为工具本身存储着电子控制单元的控制指令。为了便捷的诊断故障，全球的汽车制造商们提供着各种各样的自诊断工具。在表 3 中，我们展示多种汽车自诊断工具。我们利用自诊断工具获得信息的过程如下：

1、 自诊断工具连接到车辆的 OBD 2 接口，如图 3.a 所示

2、 在笔记本电脑连接到另一个外加的接口后，车载 CAN 总线数据就会被监测到。

3、 使用自诊断工具执行可以强制驱动确定的 ECU 的命令。

上述的步骤完成后，连接到外加接口的笔记本电脑就能获得强制控制 ECU 的 CAN 数据帧。在实验过程中，我们获得了大量各种可以控制 ECU 的 CAN 数据帧。尤其是，我们获得了能控制停止加油并且关闭引擎的喷嘴的 CAN 数据帧。同时，我们也分析了汽车正常行驶时产生的 CAN 数据帧。我们重复分析了汽车在典型驱动状态时产生的数据帧的特征（例如，突然的加速，执行智能汽车辅助停位的功能等）。表 5 展示了我们在实验中分析出来能强制控制 ECU 的 CAN 数据帧（为了避免我们研究结果被恶意使用，分析的数据帧完整的字节信息已经被删掉了）。在汽车正常行驶过程中获得的数据帧显示在表格第 1-6 行，通过汽车自诊断软件获取的数据帧显示在表格的第 7、8 行。车载 ECU 帧 ID 使用的范围从 0x000 到 0x5FF。车辆自诊断工具使用的帧 ID 从 0x700 到 0x7FF。由于使用 ID 的范围不同，很容易通过自诊断工具生成数据来确定控制 ECU 的 CAN 数据帧。

编写散布攻击使用的恶意 APP：到 2013 年 12 月为止，分布在市场上提供给汽车使用的手机 APP 已经有超过 300 种。我们生成了一个恶意 APP 并伪装成为车辆提供服务的自诊断软件。恶意软件显示车速和 ECU 错误码的同时使用手机的通信网络（例如 3G/4G 网）传送从车载 CAN 获得的 CAN 数据帧到攻击者的服务器。此外，也有可向车载 CAN 网络注入从攻击者的服务器获得的 CAN 数据帧。为了制造恶意 APP，我们在基于 Windows-7 的电脑上安装了 JAVA JDK、Android ADT 和 Android SDK 并且使用 eclipse 软件开发了这款软件。我们确认过这款 APP 可以在真正的智能手机上正常地运行。图 4 展示了我们设计的恶意 APP 的状态图和屏幕截图。实际上，我们并没有在 APP 市场上上传分享我们设计的恶意 APP。然而，根据论文【19】里说的，在 APP 市场上上传分享恶意 APP 是很容易做到的。然而，恶意 APP 使用的攻击模型有一个薄弱点。一旦攻击者使用 APP 控制了一辆汽车，软件的声誉就会开始下降，因为 Android 的使用者会很快发现这个软件是个恶意软件。加强攻击模型的一个未来研究课题可能是隐藏行为，以避免攻击被用户注意到。

B、实际攻击阶段

通过恶意智能手机的强制驱动攻击：如图 5(A)所示，使用 Android 智能手机，服务器，OBD 2 扫描工具，和一辆中型轿车，我们组织了一个环境，并进行了一次实验性攻击。图 5(B)示出了提出攻击方案的步骤。在准备阶段完成后，实际攻击是在驾驶员在他或她的智能手机上的下载并使用恶意 APP 之后开始。诊断工具是在攻击过程中没有与目标车辆有物理上的相连。OBD 2 接口扫描工具在安装上目标车辆后和司机的智能手机通过蓝牙进行配对。然后恶意自我诊断 APP 和攻击者的服务器通过手机移动通信网络进行连接。整个实验按照以下步骤实行。

1、 受害者的智能手机上安装了恶意 APP。
2、 受害人通过蓝牙或 wifi 将智能手机连接到目标车辆。恶意应用程序为受害者提供正常功能，伪装成一个自诊断 APP。
3、 恶意应用程序使用移动通信网络向攻击者服务器传输车载 CAN 网络的数据帧。攻击者的服务器检查目标车辆的状态并通过恶意软件向车载 CAN 网络发送 CAN 数据帧以强制 ECU。
4、 由于攻击者服务器发送的异常控制数据目标车辆发生物理故障。

我们的攻击实验在电视新闻上有报道。新闻视频剪辑可在网址 http：/goo.gl/mo33ay 找到。URL 上的文本被翻译成英文并附于附录。附录解释了新闻视频中 28s~1 min(46)s 的实验剪辑。这则新闻视频展示了四种攻击实验：仪表板失真、发动机停机、手柄控制和加速。在新闻录像中，我们制作并使用了仅用于广播的攻击者应用程序。运行在攻击者的智能手机上应用程序，发送攻击消息(比如可以强制控制 ECU 的 CAN 数据帧)到攻击者的服务器，然后服务器通过驾驶员受感染的智能手机将攻击消息发送并注入到目标车辆的车载 CAN 网络中。在我们的攻击模型中，攻击者的服务器直接注入攻击消息，而不使用攻击者的智能手机(攻击者的应用程序)。

5、防御

A、设计目标

为了设计一个考虑到 ECU 的低性能和 CAN 数据帧有限的数据有效负载的安全有效的协议，这些目标应该实现。

CAN 数据帧的加密和认证：为广播的数据提供认证，数据应该加密后传输。此外，若要验证所传输的数据帧，应该使用 mac 进行身份验证。产生并传播。

然而，这并不容易。在 CAN 数据帧中有效地包含 MAC。如图所示在图 1 中，can 数据帧由 120 位组成，包括 64 位数据字段。如果数据字段用于 MAC，则 CAN 数据帧传输总量至少增加两次：一个原始数据帧和至少一个负责 MAC 的数据帧。这样的方法不合适，因为它会增加 CAN 总线负载迅速。因此，安全协议需要有效的数据身份验证技术，可应用于当前 CAN 数据帧格式。

一种高效的密钥管理：为了安全通信，一个 CAN 安全协议应该为数据帧加密和认证提供一个安全的、快速的密钥分发机制。此外，还需要一个高效且安全的会话密钥更新协议，以增强会话密钥和缩短的 mac 的安全性，还需要支持外部设备和车辆之间的连接。对于安全会话密钥更新，应该确保两个属性：1）向前和向后的保密：在 CAN 中，每个 ECU 都应该使用认证会话密钥 AKn 和加密会话密钥 EKn，以确保在第 n 个会话中能够验证和机密的数据帧。密钥 AKn 和 EKn 不应该用于在 n+1 会话和第 n 次会话（向前和向后保密）中广播的 CAN 数据帧的解密和身份验证。如果不能保证向前和向后的保密，不管密钥的更新周期是什么，都不能保证机密性和身份的验证。2）密钥的新鲜度：加密密钥 EKn 和认证密钥 AKn，用于在第 n 个会话中对 CAN 数据帧进行加密和认证，应该是新生成的。为了确保密钥的新鲜度，一个用于密钥生成的参数应该使用一个随机数或计数器不断地更改。密钥的新鲜度对防止重放攻击是至关重要的。

B、提议的安全协议

为了描述我们所提议的安全协议，我们假定以下的安全协议。首先，网关和通用 ECUs 预先共享了长期对称密钥 K 和 GK。其次，长期对称密钥的加载是通过一个安全通道完成的。第三，ECUs 使用 CAN 的消息过滤功能。每个 ECU 在其 ID 表上注册 sender-ECU 的 ID，并且每个 ECU 只有在它的 ID 表上注册了 sender ID 时才会收到数据包。对于其他数据包，它执行过滤。第四，发送方和接收方 ECUs 同步数据帧与计数器。当一个数据帧被完全传输到接收器时，两者都增加了一个数据帧计数器。（在 CAN 中，发送方和接收方会使用 ACK 位字段对 CAN 数据帧的传输状态进行检查。因此，在发送方和接收方之间，可以管理和同步一个数据帧计数器。）第五，ECU 的计算能力比一般的 ECU 要高。第六，设备证书被加载到网关 ECU 和外部设备上。

我们提议的安全协议分为五个阶段。阶段 1 和 2 使用一个众所周知的安全技术（长期对称密钥和经过身份验证的密钥交换协议 2（AKEP2））来构建初始会话密钥分发。在我们提议的安全协议中，主要的新贡献在于阶段 3、4 和 5。本文中所使用的符号列在表 v.1 中。

1）加载了长期对称密钥：ECUi 加载长期的对称密钥 Ki 和 GK 到安全存储。GECU 装载 N 个 Ki 和一个 GK 到安全的存储器中。（值 N 是安装在车上的 ECUs 的数量。）在制造车辆或更换 ECU 时，只执行长时间对称的键加载阶段。

2）初始会话密钥的分发：在启动一辆车后，每个 ECU 以固定的顺序执行初始会话密钥的派生过程。虽然 GECU 在最初的会话密钥中使用了一个特定的 ECU，但其他 ECUs 不会进行通信，而是等待轮到它们。我们使用 AKEP2 在车载环境中构建安全有效的密钥派生过程，因为它提供了相互实体身份验证和隐式密钥分发[21]。当[21]明确允许从协议中删除冗余部分而不影响安全性时，我们删除了冗余部分，并为密钥确认添加了一个值。初始会话密钥的分布如图 6 所示。

（A）ECUi 选择随机数 Ri 并将其传输给 GECU。

（B）GECU 选择随机数字 Seed1，并使用 Ki 为 ID、ID GECU、Ri 和 Seed1 生成 MAC1，并将其和 Seed1 一起发送给 ECUi。

（C）ECUi 验证 MAC1。

（D）ECUi 计算初始的会话密钥如

KDFGK(Seed1)=EK1||AK1||KEK1||KGK1||UK

（E）ECUi 为 IDi and Seed1 使用 Ki 生成 MAC2。它还为 IDi 、EK1、AK1、KEK1、KGK1 和 UK 生成 MAC3，使用 AK1。MAC2 和 MAC3 被传送到 GECU。

（F）GECU 验证 MAC2。通过验证 MAC2，可以确认 ECUi 是否正确地接收了 Seed1。在 MAC2 验证之后，GECU 按照（1）计算初始会话密钥。

（G）在生成初始会话密钥之后，GECU 使用 AK1 验证 MAC3。通过验证 MAC3，可以确认 ECUi 正确地生成了初始会话密钥。

3）可数据帧的加密和认证：在初始会话密钥分发过程完成后，每个 ECU 对在车辆正常驾驶过程中生成的数据帧进行加密和认证。在我们的协议中，我们使用高级加密标准 128（AES-128）和 Keyed-Hash MAC。我们提出了两种方法：基本的和增强的。数据帧加密和身份验证在图 7 和图 8 中显示。

基本方法：

消息传输（kth 会话）发送者 ECUs 管理自己的数据帧计数器值 CTRECUs。在传输数据帧时，ECUs 使用 CTRECUs 从 C=EEKk（CTRECUs）生成密文。当使用 AES-128 算法时，EEKk（CTRECUs）的结果是 128 位。由于 CAN 数据有效负载的最大大小是 64 位，所以只有前 64 位用于生成 ciphertext（C）.ECUs 为 CAN 数据框架生成一个 MAC 值，包括 ciphertext（C）和 CTRECUs，如下：MAC=H2，AKk（IDsCCTRECUs）。由于数据帧有效负载是 8 字节，所以我们使用一个截断的 32 位 MAC 和一个除法方法来传输它。如图 1 所示，CAN 2.0 B 的 ID 字段被分为两个子字段。我们提出的安全协议使用扩展 ID 字段的前 16 位和用于 MAC 传输的 16bit CRC 字段。在 18 位扩展的 ID 字段中，未使用的两个位被设置为零。CRC 用于接收数据的完整性。由于 MAC 同时提供了数据验证和完整性，所以它仍然确保接收到的数据没有被修改。可能发生 MAC 验证延迟; 然而，延迟并不严重，也不会干扰数据的实时处理。我们在实验中证明，MAC 的性能与正常的性能相似。第七节详细描述了绩效评估实验。图 7 显示了分割过程的细节。一旦 ECUs 完成前面提到的步骤，它就会传输安全的数据帧和增量 CTRECUs。

消息接收（kth 会话）

（A）接收器 ECUr 管理 ECUs 的计数器。ECUr 通过 CTRECUs 和 AKk 来验证它的安全。

（B）当 stepa）被正确完成时，ECUr 通过以下方法进行解密并获得明文（M）：M=EEKk（CTRECUs）c。（4）

（C）在完成对接收的数据帧的验证和解密后，ECUr 增加了 ECUs（CTRECUs）的 CAN 数据帧计数器。

改进算法：

在 AES-128 算法中增强的方法，EEKk（CTRECUs）的结果始终是 128 位。在增强的方法中，两个 CAN 数据帧使用一个 AES 加密的结果进行加密。由于 CAN 数据有效负载的大小是 64 位，所以一个 128 位的 EEKk（CTRECUs）的计算被两次划分为两个部分（左和右最多 64 位）。因此，AES 的计算成本是基本方法的一半。

4）密钥更新：一个 32 位的截短的 MAC 不足以保证安全。此外，一个对手可能使用外部设备连接来泄露操作键。因此，每个会话使用的加密和认证密钥应该定期更新。在第 k 次会议上，推荐的密钥更新阶段如下。

GECU 执行每个预定义周期（T）的密钥更新，如下所列。（图 9 显示了密钥的更新过程。）

（A）在选择了西 k+1 的随机值后，GECU 生成一个密钥的请求消息并将其广播到车内的 CAN。这条信息是由以下内容组成的。

（B）接收到密钥请求消息的每个 ECUi 分别对 MAC 和 AKk 和 KEKk 进行解密（C）的验证。然后，ECUi 将使用 KGKk 的预定义函数 KDF（）来获得 k+1 会话中的会话密钥。每个数据帧计数器也被初始化为零。

（C）在步骤 B 之后），每个 ECUi 会生成一个密钥的响应消息，并将其发送给 GECU，以确认他们收到了正确的密钥请求消息。

（D）在密钥确认之后，GECU 将 ECUi 的每个数据帧计数器初始化为零。当 GECU 初始化自己的数据帧计数器时，密钥的更新阶段就完成了。

在外部设备被释放连接（kth 会话）时，密钥更新阶段。如果外部设备和车辆之间的连接被终止，GECU 执行一个会话密钥更新过程，以保持保密。（A）GECU 生成种子的随机值。然后，它生成一个密钥的请求消息。

然后，GECU 将密钥的请求消息发送到车内。（B）接收到一个密钥请求消息的每个 ECUi 分别对 MAC 和使用 AKk 和 UK 的加密文本（C）进行解密。其余的步骤与一般的密钥更新阶段是相同的。5）与外部设备共享一个会话密钥：我们建议在将车辆与汽车诊断工具等外部设备连接起来时，为额外的认证和密钥分发提供一个阶段，假设它是一种可靠的设备。在第 kth 会话中，外部设备认证和会话密钥分发如下。

（A）在将外部设备连接到车辆后，外部设备向 GECU 发送身份验证请求。

（B）在收到请求后，GECU 生成一个随机数 r1（r1 ∈ Z* q),）和 r1P 上的签名。然后，它通过证书将这些值传递给外部设备。（在这个椭圆曲线组中，P 是 G 的发电机点，它的顺序是一个大的 q.）

（C）如果由 GECU 传输的证书和签名被成功验证，外部设备会生成一个随机数 r2(r2 ∈ Z* q)和 r2P 的签名。然后，它用证书将这些值传递给 GECU。在完成传输之后，外部设备产生一个临时会话密钥（(SK = r1r2P）并删除 r2。

（D）在验证由外部设备传送的证书和签名之后，GECU 生成一个临时会话密钥（(SK = r1r2P）并删除 r1。然后，GECU 将 Seedk 与 SK 加密，并将其传输到外部设备。外部设备派生出在 kth 会话中使用的会话密钥，然后可以进行通信。

6、相关工作

随着联网汽车的商业化，在过去被认为是一个封闭的网络的车载 CAN 网络，现在正在连接到外部网络并为远程诊断提供有用的服务,如[18]、[23],固件无线更新[24]、[25],和实时产品碳排放数据数据分析[26]。另一方面，这种与外部网络

的连接为车辆带来了一种新的安全威胁。Koscher 等人提出了特定的无线攻击技术，并尝试了远程和远程无线攻击。当安装在车上的蓝牙设备与驾驶员的智能手机配对时，就有可能发生短程无线攻击。由于在 aqLink 协议中身份验证功能的脆弱性，远程无线攻击是可能的。然而，在[19]中要在复杂和先进的汽车电子设备上进行无线攻击，比如反向工程，就需要分析汽车电子设备。此外，远程无线攻击仅适用于使用 aqLink 协议的车辆。之前关于车辆安全的研究指出，车内的漏洞可能是网络攻击的主要原因，[8], [13].。特别是，在[9]中提到缺乏数据帧认证和加密是 CAN 协议最严重的漏洞。

为了建立一个安全的车内 CAN 网络，在过去的十年里进行了各种各样的研究和研究项目。作为一个由欧洲资助的项目，EVITA 开发了一个用于车载网络安全的硬件安全模块（HSM）。根据使用的领域，HSMs 可分为三种类型。

1）适用于车辆网络（内部或内部）的全 HSMs。

2）适合于汽车内部网络的中型 HSMs。

3）适用于传感器和执行器的轻型 HSMs。

Schweppe 等人建议在[11]和[12]中使用 EVITA-HSM 的车辆进行通信安全架构。他们使用了一个被截断的 32 位 MAC，考虑到 CAN 数据帧的有限数据有效负载，并解释说，由于车载网络的有限属性（可以总线负载和带宽），32 位 MAC 在 35 周内安全受到碰撞攻击。然而，Schweppe 等人的安全架构非常抽象。它并没有提供详细的描述，包括所有的和传输的。它也不考虑数据机密性和与外部设备的连接。

为了提供一个可以防止重放攻击的车载 CAN 通信环境，[14]和[15]提出一种考虑了一个 CAN 数据框架的有限数据有效负载的数据认证技术。Groza 和 Marvay 在[14]中提出了一种可以使用类似 TESLA 协议的数据验证协议。在 TESLA 协议中，一个发送者连接到每一个数据，一个 MAC 用一个只知道发送者的 key k 来计算。不久之后，发送方将 k 发送给接收者，后者可以对数据进行身份验证。我们注意到，在类似于 TESLA 的协议中，关键的信息披露延迟应该被最小化，以确保能够进行实时处理。然而，延迟越短，总线负载就越大。下一节的模拟显示，类似于 TESLA 的协议[14]发现很难在 CAN 中提供实时处理。Groza 等人还提出了一个单一的主案例，以尽量减少关键的信息披露延迟。在一个主案例中，发送者生成一个带有一个长期秘密密钥的 MAC，与通信主机和发送器和响应 MAC 共享给主服务器。然后，主将数据和 MAC 传输给接收者。然而，由于发送方和通信主之间的密钥不能被更改，因此在监听传输的 CAN 数据帧和 MAC 之后，重复攻击是可能的。

Lin 等人提出了一种使用 ID 表、消息计数器和双向对称密钥（PWSK）的 MAC 生成技术。接收者的 ID 在发送者的 ID 表上注册。假设发送方与 ID 表中的接收者共享一个 PWSK。他们的 MAC 生成技术与我们的相似，因为它在 ECUs 中使用了同步的消息计数器。然而，Lin 等人的协议使用的是 PWSK，而我们使用的是组会话密钥。使用 PWSK 意味着发送方必须在通信组中生成尽可能多的 mac，并将它们分别传输给每个接收方。这将迅速增加公共汽车的负载，因此不切实际。此外，他们的安全技术不考虑数据机密性和与外部设备的连接。在第七节中，我们对提出的安全协议和[14]、[15]的安全协议进行比较评估。

7、安全与性能分析

安全分析

•机密性：我们提出的安全协议使用 AES 加密算法来确保 CAN 数据框架的机密性。当一辆汽车启动时，每一个 ECU 使用长期对称密钥（Ki）和（GK）执行初始会话密钥派生过程。虽然存储长期对称密钥（Ki）和（GK）的合法 ECUs 可以计算初始会话密钥，但对手可以使用它。这意味着 annotarycannot 获取加密密钥（EK1）和从 Seed1 派生出的身份验证密钥（AK1）。在随后的会话中，对手无法获得任何会话密钥，因为 Seedi 已被 KEKi 1 加密。由于 AES 算法的安全性已经在 27 中得到验证，所以很明显，如果没有会话密钥，对手就无法获得 CAN 数据。

•身份验证：我们使用 32 位截短的 MAC，与 evita-中型 hsm 使用的 MAC 相同。根据[28]的说法，如果一个对手能够访问 32 位的 MAC 一代 oracle，那么一个 32 位的 MAC 可以在 O（216）的查询中被破解。这意味着，对于能够访问 ECU 固件的对手来说，这是有可能的。然而，我们并不认为这种类型的对手，如第 iii 部分所解释的那样。攻击者也可以使用已知的输入结构到 MAC 来生成有意义的消息。然而，攻击者仍然不能在不知道 MAC 键的情况下生成对应于有意义消息的 MAC。用于生成 MAC 的密钥在提议的协议中是安全共享的。对于攻击者来说，在 232 个可能的 MAC 值中选择一个 32 位的字符串是唯一的选择。虽然一个 32 位的 MAC 可以在几秒钟内在一个通用的 IT 环境中被伪造，但是它可以访问 MAC 一代 oracle，而一个对手在每 10 个 ms 中传输 232 个数据帧就需要大约 11930 个小时。如果一个对手将一个恶意的数据帧传输到一个车载系统中，那么这个网络将会产生一个“CAN 总线关闭”的错误状态，指示通信失败（这个攻击可以被入侵检测系统检测到）。我们还设计了一个用于 32 位 MAC 安全性的会话密钥更新协议。

•向前和向后的密钥保密：在拟议的安全协议中，一个会话密钥被外部设备公开，该设备与车载设备进行通信。然而，外部设备很难获得前进会话的键。如果外部设备和车载设备之间的连接失效，GECU 就会播放包括种子新在内的密钥请求信息。因为密钥请求消息是由英国加密的，所以外部设备无法知道用于转发会话的键。要获得一个逆向会话的密钥也很困难。例如，尽管 kth 会话密钥的 EKk、AKk 和 KEKk 被公开，但是不可能获得 EKk 1、AKk 1 和 k 第 1 个会话密钥中的 KEKk 1，因为 seedk+1 和 seedk 之间没有关联。

•密钥新鲜度：每个会话使用的密钥都来自于随机生成的值，因此，它们之间没有关联。换句话说，seed1、seed2、seed3 和 seedn 是不同的值。

•重播攻击：在我们提出的安全协议中，发送方和接收方管理数据帧计数器。数据帧计数器在它的发送方和接收方之间的同步和管理以生成 MAC。如图 7 和 8 所示,第五部分,发送方使用数据帧计数器以生成 MAC。这样,由于数据帧计数器是用于生成 MAC 的一部分,所以我们建议的安全协议对再现攻击是安全的。

性能评估

为实现对所提议的安全协议的性能分析，我们制造了一个与真实的 ECU 具有相似功能的安全系统，然后执行一个基于硬件的模拟。用独木舟软件将安全系

统插入到安全系统中。模拟环境如图 10 所示。表六显示了用于评估的设备的规格。

1）基于硬件的评估（F28335 微控制器）：我们在德州仪器的 DSP-F28335 微控制器上制造了一个安全装置，用于评估。加密和验证的执行时间是通过实现算法（AES-128，MAC）实现的，在安全系统固件中实现。将 DSP-F28335 微控制器的 CPU 时钟频率更改为 150、120、90 和 60 MHz，我们分析了所提出的安全协议的执行时间。为了更准确的评估，我们重复了 10 000 次协议，获得了平均执行时间，如图 11（a）所示。

如果使用了增强技术，当 CPU 时钟频率为 60 MHz 时，可以在 378 秒内执行 CAN 数据帧的加密和身份验证。我们注意到，如果提议的安全协议是在特定于应用程序的集成电路（ASICs）上实现的，那么执行时间将比我们的实现结果 29、30 更快。

2）软件——基于硬件的评估：构建一种类似于汽车的评估环境，我们使用的是向量 Co.独木舟，是用于开发或测试嵌入式系统的网络仿真软件。图 10 所示，使用安全-ecu、独木舟和 VN7600 接口进行环境评估。在独木舟虚拟 ECU 节点上也实现了拟议的安全协议。我们实现了我们的安全协议，并构建了一个 DLL（动态链接库），以便在独木舟中应用它。然而，我们实际上不能将 32 位的 MAC 值加载到扩展 id 和 CRC 字段，因为像 DSP-F28335 这样的微控制器通常是这样的，所以扩展 id 和 CRC 字段的独特功能是不可更改的。因此，我们实现了基于硬件的评估的结果，作为软件的执行时间延迟——基于硬件的评估。在将执行时间延迟设置为在传输之前和接收到数据帧之后，我们进行了软件——基于硬件的评估。

•通信响应时间：我们用独木舟连接安全-ECU（接收器 ECU）来测量通信响应时间。在这个实验中，我们改变了虚拟 ECUs（发送方 ECUs）的数量，通过 5、10、15 和 20 来传输到 CAN 数据帧，并测量了安全 ECU 的通信响应时间。虚拟 ECUs 以 10 ms 的循环播放 CAN 数据帧。在接收到数据帧之后，安全 ECU 执行了身份验证和解密，然后传输了一个响应 CAN 数据帧.在图 11（b）中，我们根据虚拟 ECUs 的数量绘制了安全 ECU 的响应时间。当假定安全 ECU 的 CPU 时钟频率为 150 MHz 时，在实现所提议的协议时，一般和修改的罐子之间没有显著的区别。然而，尽管图 11（b）中没有显示时钟频率低于 90 MHz 时，当虚拟 ECUs 的数量超过 15 时，接收到的数据帧中出现了丢失。数据帧的丢失是因为接收到的数据帧的循环比在数据帧传输和接收过程中解密和认证所需的执行时间要快。

然而，我们也进行了性能评估实验，同时设置了比非典型的车辆更大的通信负荷。一个典型的车载系统可以分为三个子网络：1）动力传输系统和底盘；2）车体的电子单元；3）车载影音娱乐系统。每个子网络由少于 15 个 ECUs 组成。在沃尔沃 XC90 的案例中，超过 40 个 ECUs 安装在两个以上的子网络上。特别地，最大的子网络是身体电子功能，其中 13 个 ECUs 相互通信 5、6。此外，在车辆开发中，已经安装了超过 150 MHz 的微控制器，因此可以使用我们提出的安全协议，而不需要在一般的车载环境中使用数据帧丢失。

•初始会话关键推导时间：假设这些 ECU 被用作 GECU，我们用 ECUs 的数量和 CPU 时钟频率来测量初始会话的关键推导时间。GECU 的 CPU 时钟频率固定在 150 MHz。图 11（c）显示了从我们的实验中得到的平均初始会话的关键推导时间的结果。我们提出的安全协议使用 AKEP2 来获得初始会话密钥。为了以

一种安全的方式建立会话密钥，AKEP2 执行了一个经过身份验证的三方握手。一旦某个 ECU 与 GECU 进行了三次握手，通信响应时间延迟发生两次。此外，当下一个 ECU 在确认了三向握手的最后三分之一后，开始了一个初始的会话关键的推导，如果 N ECUs 的与 gecun-1 通信响应时间延迟的握手方式发生了。换句话说，当 N ECUs 对 GECU 进行初始会话的关键推导时发生的通信响应时间延迟是这样的：（3N+1）*（通信响应时间延迟）。

图 11（c）的结果与图 11（a）、（b）的结果相结合，表明初始会话关键的派生执行时间的差异来自于 MAC 函数执行时间的差异。在经过身份验证的三次握手中，MAC 函数被使用了 6 次。当 ECU CPU 时钟频率为 150 或 60 MHz 时，验证的三向握手所需的 MAC 函数执行时间的差异大约为 720 μs。对结果的综合分析表明，与 MAC 函数执行时间相比，通信响应时间延迟对初始会话关键的推导时间有更大的影响。此外，如图 11（c）的结果所示，60 个 ECUs 完成初始会话关键派生过程的时间少于 235 ms。因此，在将我们的安全协议应用于性能较低的 ECUs 的车辆时，它的可用性可能得到充分的保证。

•密钥更新时间：我们通过实验测量了与初始会话密钥派生时间相同的环境中的密钥更新时间。图 11（d）显示了结果，可以看到密钥更新可以在 6 ms 内执行。密钥的更新时间是相似的，不管 CPU 时钟频率是什么，除了 60 MHz 之外，因为对一个键请求数据帧的接收和一个密钥响应数据帧的生成都是在所有 ECUs 上并行执行的。换句话说，密钥更新时间的增加与 ECUs 和通信速度的数量成正比，而不是 ECUs 的性能（CPU 时钟频率）。

安全性和效率比较，

这里，我们将我们的协议与[14] [15]中提出的的协议进行比较。为了方便描述，我们建议的安全协议被表示为我们的，而[14]的协议被表示为 EPSB（安全广播的有效协议）和[15]的协议被表示为 IDT&C（using ID Table &message 计数器）。EPSB 有两种模式：单主模式和多主机模式。我们只考虑单主模式，因为多主机模式需要高总线负载。在 EPSB 的单一主模式中，一个通信主对每个发送方传输的数据帧进行身份验证。为了进行详细的比较，我们将 EPSB 的单主模式划分为 EPSB 1 和 EPSB 2。EPSB 1 是一种既包含消息又包含 MAC 的模式。EPSB 2 是一种传输两个 CAN 数据帧的模式，一个用于消息，另一个用于 MAC。IDT&C 在通信组中产生的 MAC 消息和接收方一样多。在通信组中，生成的额外消息的数量是 M（m1），以使组中的每个 ECU 交换消息。我们使用表 7 中给出的评估环境，分析了我们的总线负载，EPSB-1，EPSB-2 和 IDT&C。关键的信息披露延迟只适用于 EPSB-1 和 EPSB-2。图 12 显示了我们的总线负载，EPSB-1，EPSB-2，和 IDT&C，在 ECU 的数量和关键的信息披露延迟方面。在一般情况下，总线负载必须保持在最大的 50%以下，以保持一个稳定的通信环境。如图 12（a）所示，我们的总线负载保持在 50%以下，尽管有 20 个 ECUs，因为它既不需要额外的 CAN 数据框架，也不使用密钥披露来进行消息身份验证。

然而，在 EPSB 1 和 EPSB 2 的情况下，由于额外的数据帧和关键的信息披露，总线负载增加了 50%以上。只有当少于 5 个 ECUs 和关键的信息披露延迟时，EPSB 才能保持低于 50%的总线负载。IDT&C 也会迅速增加总线负载，因为它还需要传输尽可能多的 MAC 消息作为接收方。只有在通信组中有 5 个 ECUs 时，IDT&C 可以保持 60%的总线负载。表 8 显示了我们的安全性和效率的总体比较，EPSB 和 IDT&C。表八中所列数据的详细比较评价如下。

1）安全措施（数据加密和认证）。

2）与外部设备的连接（即：与外部设备的密钥交换）。

3）所提议的技术的效率（是否能维持低于 50%的公共汽车负荷）。

4）来自我们的攻击模型的安全性。（我们的攻击模型使用了消息重放攻击。如果安全技术能够确保消息重放攻击的安全性，那么它们将从我们的攻击模型中得到保护。）

EPSB 和 IDT&C 为 CAN 数据帧提供认证，但他们不考虑在车辆操作期间外部设备的连接。此外，在 EPSB 和 IDT&C 中，有可能分析某些数据帧的含义，因为不保证数据帧机密性。如图 12 所示，EPSB 和 IDT&C 大大增加了公共汽车的负载。EPSB 也很容易受到消息重放攻击的影响，因为发送方和通信主机之间的密钥在每个会话中都没有更改。因此，在窃听包括 mac 在内的数据帧后，可以执行重放攻击。IDT&C 有缺点，因为 MAC 必须另外传输消息身份验证，并且在 MAC 验证之前不能使用消息。然而，它是安全的，不受重玩攻击。根据实验结果，在使用 IDT&C 对小于 5 个 ECUs 的通信组时，可以用我们提出的攻击模型来保证安全。与前面提到的相比，我们提出的安全协议既提供了安全性，又提供了效率。

我们支持外部设备的连接。此外，正如前面提到的，我们提供了数据框架的机密性和身份验证。我们提供的是机密性和身份验证，但很少增加总线负载，支持实时处理 CAN 数据帧。此外，在我们的情况下，重放攻击是不可能的，因为发送方和接收方之间的计数器被管理，用于加密和验证 CAN 数据帧。

8、结论

最近，许多关于车内易受攻击的研究已经完成。然而，这种攻击模式是不现实的，因为它们需要大量的努力和复杂的技术，如逆向工程和劫车。因此，在本文中，我们提出了一种实际的攻击模型，在联网汽车环境中使用恶意的智能手机应用程序，并通过实际的实验证明了这一点。在演示了攻击模型并分析了车内的易损性之后，我们设计了一种安全协议，可以应用到汽车环境中。此外，我们还通过对安全与独木舟的评估，分析了所提议的安全协议的安全性和性能。在未来，我们计划通过在硬件上实现加密和散列算法来优化安全协议的性能，以优化我们的安全技术。

9、附录

http://goo 的新闻报道的内容。gl/mo33ay 的总结如下。•（00:28）应用程序（app）用于诊断一个小静脉。•(00:31)无线连接汽车电子控制器,智能手机显示的信息不显示在仪表盘上,如车辆故障信息、准确一加仑汽油所行驶的里程信息,和旅游路线信息。•(00:43)有超过 200 车辆诊断应用的市场由许多司机使用。•(00:50)然而,问题是,如果一个黑客安装这个程序的恶意代码,他/她就可以 toexternally 控制汽车。•（00:59）我们在韩国大学的研究小组的支持下进行了一项实验。•（01:04），

一旦攻击者的智能手机按钮被按下，指示板立即触发警报。（01:12）汽车的点火突然关闭。（01:19）当使用智能停车系统时，方向盘是随机的。•（01:26）更危险的远程控制也是可能的。当快速加速按钮被压在智能手机上时，车辆的速度会立即增加，当车辆被提升时，它的速度会增加到 160 公里/时。•（01:46）以这种方式，黑客可以使用被其恶意代码感染的受害者智能手机来控制汽车。