学校代号 10532　　学　号 S1310M1007

分 类 号 TP393　　密　级 普通

工程硕士学位论文

智能汽车网络安全若干关键技术研究

学位申请人姓名 朱立民

培 养 单 位 信息科学与工程学院

导师姓名及职称 李仁发 教授　唐涛 高级工程师

学 科 专 业 计算机技术

研 究 方 向 嵌入式系统安全

论文提交日 期 2017 年 3 月 7 日

学校代号：10532

学　　号：S1310M1007

密　　级：普通

湖南大学工程硕士学位论文

智能汽车网络安全若干关键技术研究

学位申请人姓名：　朱立民

导师姓名及职称：　李仁发 教授　唐涛 高级工程师

培　养　单　位：　信息科学与工程学院

专　业　名　称：　计算机技术

论 文 提 交日 期：　2017 年 3 月 7 日

论 文 答 辩日 期：　2017 年 3 月 18 日

答辩委员会主席：　谢鲲 教授

Research on Several Key Technologies for Network Security of the Smart Vehicles

by

ZHU Limin

B.E. (Dalian Neusoft University of Information) 2013

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Engineering

In

Computer Technology

in the

Graduate School

Of

Hunan University

Supervisor

Professor LI Renfa

Senior Engineer TANG Tao

March, 2017

湖 南 大 学

学位论文原创性声明

本人郑重声明：所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名：　　日期：　　年　月　日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权湖南大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密□，在_____年解密后适用本授权书。

2、不保密□。

（请在以上相应方框内打“√”）

作者签名：　　日期：　　年　月　日

导师签名：　　日期：　　年　月　日

摘　要

随着自动化技术、人工智能技术和无线通讯技术在汽车领域不断取得突破性进展，智能汽车运用而生。智能汽车在给消费者带来良好驾驶体验的同时也暴露出很多不足。因为其电子系统越来越依赖于信息共享和车载网络通信，所以汽车内部网络不可避免的与外界进行大量数据交换，使针对汽车的网络攻击成为可能。通过研究发现，以 CAN 总线为例的传统汽车总线已经不能满足智能汽车对安全性的需求，具有抗网络攻击的汽车总线成为发展趋势。ECU（Electronic Control Unit）的低处理能力对现代密码学的应用带来很大挑战，因此提出一种轻量级安全汽车总线十分重要。同时，虽然目前已经存在若干防御方法，但用于保护车内网络的密码系统仍有可能受到 DFA（Differential Fault Analysis）攻击等侧信道攻击的威胁。针对以上问题，本文的主要工作如下：

首先，针对普通 CAN 总线的安全缺陷，提出基于 AES-CCM 算法的安全 CAN 总线协议。分别通过对数据帧 CRC 场和扩展帧 ID 场的利用，提出两种不同的方案，使 CAN 总线具有机密性、可认证性和抗重放攻击的能力。

其次，提出了一种针对 AES 解密的 DFA 攻击方法。根据 AES 算法的性质推导验证了 DFA 攻击能在其解密过程产生威胁，并将密钥空间缩减至 2^{32}。进而，本文通过构建 S 盒分布表提出一种优化的攻击方法。相比基础攻击方法，具有相同攻击强度的条件下可大幅提高攻击速度。

最后，本文利用两块飞思卡尔 MC9S12XF512 开发板作为实验平台，实现了所提出的安全 CAN 总线协议，对其安全性和性能进行了分析与评测，并与相关研究进行对比分析。同时，本文在 VS 2010 开发平台上对所提出的针对 AES 解密的 DFA 攻击方法进行仿真实验，通过具体示例验证了攻击方法的有效性且用 2 对正确明文和故障明文可以在 70 毫秒内破解完整密钥，与前人工作相比减少了 79.5%的计算时间。

关键词： 智能汽车；CAN 总线；AES 算法；DFA 攻击；CCM 模式

Abstract

With the continuous breakthrough in the fields of automation technology, artificial intelligence technology and wireless communication technology, there comes up the smart vehicles. Smart vehicles have provided good driving experience to customers, but at the same time they exposed some shortcomings. Their electronic system is increasingly dependent on the information sharing and vehicle network communication, so the automobile network inevitably exchange a large amount of data with the outside world, which made it possible for the smart vehicles to be under network attacks. Research shows that CAN bus, as a sample of internal network of vehicles, can no longer meet the security requirements of smart vehicles. The vehicle buses, with ability to defect network attacks, are becoming a trend. The low processing ability of ECUs (Electronic Control Unit) makes a great challenge to the application of modern cryptography on vehicles, so it is important to propose a lightweight vehicle security bus. Although there are already a number of related defense methods, the cryptosystem using for protecting automobile network is still under the threat of side channel attacks like DFA (Differential Fault Analysis) attack. To solve the problems above, the main works of this paper are as follows:

Firstly, we propose a AES-CCM algorithm based security CAN bus protocol for offsetting the security defects of traditional CAN bus. By using the data frame's CRC field and extended frame's ID field respectively, we propose two different methods to make CAN bus have the abilities of confidentiality, authenticity and anti-replay attack.

Secondly, we propose a DFA attack method for AES decryption. According to the nature of AES algorithm, we verify that DFA attacks can be performed in AES decryption process, and reduce the key searching space to 2^{32}. We also propose an optimized attack method by creating a S-box distribution table in this paper. Comparing with the basic attack method, the optimized attack method can greatly improve the attack processing speed with the same attack intensity.

Finally, we accomplish the proposed security CAN bus protocol by using

two Freescale MC9S12XF512 development boards as the experimental platform. We analyze and evaluate the protocol's security level and performance. The current research are also compared with other related researches. At the same time, the simulation experiment of DFA attack method for AES decryption is carried out on VS 2010 development platform, the effectiveness of this proposed attack method is verified through a concrete example. The full cipher key can be retrieved in 70 ms with two pairs of fault-free and faulty plaintexts, which has reduced 79.5% computational time compared with the exiting work.

Key Words: Smart vehicle; CAN bus; AES algorithm; DFA attack; CCM mode

目 录

学位论文原创性声明.....I
摘 要.....II
Abstract.....III
插图索引.....VII
附表索引.....VIII
第 1 章 绪论.....1
1.1 研究背景.....1
1.1.1 智能汽车的概述及特点.....1
1.1.2 国内外研究现状.....2
1.2 研究问题.....3
1.2.1 网络安全.....3
1.2.2 网络体系结构.....4
1.2.3 汽车总线安全.....5
1.3 研究内容与贡献.....6
1.3.1 研究内容.....6
1.3.2 本文贡献.....7
1.4 结构安排.....7
第 2 章 研究基础及相关进展.....9
2.1 安全 CAN 总线协议的研究基础与相关研究.....9
2.1.1 CAN 总线协议.....9
2.1.2 高级加密标准.....10
2.1.3 汽车网络安全的相关研究.....14
2.2 差分故障分析攻击的研究基础与相关研究.....15
2.2.1 AES 加密模式.....15
2.2.2 差分故障分析攻击技术.....20
2.2.3 故障分析攻击的相关研究.....23
2.3 小结.....24
第 3 章 基于 AES-CCM 算法的安全 CAN 总线协议.....25
3.1 设计前提.....25
3.2 设计思路.....25
3.3 具体方案.....26

3.3.1 方案一 27
3.3.2 方案二 29
3.3.3 对比分析 30
3.4 实验分析 31
3.4.1 实验平台 31
3.4.2 性能分析 32
3.4.3 工作对比 34
3.4.4 安全性分析 35
3.5 小结 36
第 4 章 一种针对 AES 解密的高效 DFA 攻击方法 37
4.1 故障模型 37
4.2 基础攻击方法 37
4.3 优化攻击方法 41
4.3.1 构建 S 盒分布表 41
4.3.2 攻击过程 42
4.4 实验及分析 43
4.4.1 实验平台 43
4.4.2 选取实验参数 43
4.4.3 密钥恢复结果 44
4.4.4 工作结果对比 46
4.5 小结 46
结论和展望 48
参考文献 50
致　谢 54
附录 A 攻读硕士学位期间发表的学术论文 55
附录 B 攻读硕士学位期间所参与的项目 56

插图索引

图 1.1 智能汽车整体网络结构 4
图 1.2 智能汽车车内网络结构 5
图 2.1 CAN 数据帧格式 9
图 2.2 N_b=4, N_k=4 时的轮密钥选择情况 12
图 2.3 AES 解密过程 13
图 2.4 电子密码本模式加密 15
图 2.5 电子密码本模式解密 15
图 2.6 密码分组链接模式加密 16
图 2.7 密码分组链接模式解密 16
图 2.8 密码反馈模式加密 17
图 2.9 密码反馈模式解密 17
图 2.10 输出反馈模式加密 18
图 2.11 输出反馈模式解密 18
图 2.12 计数器模式加密 18
图 2.13 计数器模式解密 19
图 2.14 典型的侧信道攻击示例 20
图 2.15 故障发生在第 10 轮的错误扩展图 23
图 3.1 方案一的安全 CAN 总线协议示意图 27
图 3.2 标准帧与扩展帧对比图 29
图 3.3 方案二的安全 CAN 总线协议示意图 30
图 3.4 安全 CAN 总线协议的实验平台 31
图 3.5 安全 CAN 总线协议的软件架构 32
图 3.6 USBCAN II 检测到的安全 CAN 总线协议数据帧 33
图 3.7 安全 CAN 总线数据帧验证通过示意图 33
图 3.8 各模块的性能对比结果 34
图 4.1 在 AES 解密过程第三轮输入状态位置注入一个故障的密钥扩展过程 38
图 4.2 四条对角线在 AES 解密第三轮的故障扩展图 39
图 4.3 故障注入在 D0 对角线对应的第 2 轮及第 1 轮故障扩展图 40
图 4.4 故障注入在 D1 对角线对应的第 2 轮及第 1 轮故障扩展图 40
图 4.5 故障注入在 D2 对角线对应的第 2 轮及第 1 轮故障扩展图 40
图 4.6 故障注入在 D3 对角线对应的第 2 轮及第 1 轮故障扩展图 41
图 4.7 攻击结果和攻击耗时 46

附表索引

表 3.1 设计方案的相关定义 27
表 3.2 密钥 K、随机数 N、明文 P 和附加消息 A 的值 32
表 3.3 密文 CipherText 和消息认证码 MAC 的值 32
表 3.4 ECU 配置参数 34
表 3.5 方案二与其他工作的对比结果 35
表 4.1 密钥 K 和密文 CT 的值 43
表 4.2 正确明文 PT 的值 43
表 4.3 故障明文 PT_1*和 PT_2*的值 44
表 4.4 PT_1*对应（K_{00}、K_{05}、K_{10}、K_{15}）组的所有候选值 44
表 4.5 PT_2*对应（K_{00}、K_{05}、K_{10}、K_{15}）组的所有候选值 45
表 4.6 PT_1*，PT_2*（K_{00}、K_{05}、K_{10}、K_{15}）组的正确密钥 45
表 4.7 与其他工作结果对比 46

第1章 绪论

智能汽车是汽车行业最前沿的科技，吸引着越来越多的汽车生产商和学者的关注。本章首先概述了论文的研究背景及研究现状，进而重点分析了智能汽车所面临的问题，最后本章提出研究内容与工作贡献并列出论文结构。

1.1 研究背景

1.1.1 智能汽车的概述及特点

随着科技进步和人们生活质量的提高，消费者对汽车在安全、舒适和娱乐等方面的需求越来越高。近年来，自动化技术、人工智能技术和无线通信技术在汽车上的应用取得突破性进展，智能汽车运用而生。智能汽车以传统汽车为基础，搭载雷达、摄像头和激光等传感器在网络环境下利用信息技术和智能控制技术进行操纵，使汽车具有信息交换和环境识别等功能，代替部分人的操作进而实现辅助/无人驾驶。例如自适应巡航控制系统可以实现根据道路条件自动跟车、避障和道路保持等功能，能够有效提高驾驶员对车辆的控制与驾驶能力，确保车辆行驶的安全和高效。同时，智能汽车将车辆本身信息与互联网操作系统、大数据、导航、多媒体和零部件服务信息整合成线上和线下，为用户提供智慧出行的良好体验。

智能汽车朝着电子化、网络化和智能化方向的快速发展使得整个汽车电子系统更加复杂。车内的电子控制单元（Electronic Control Unit, ECU）已经达到百余个，存有千万行的代码，其软件的容量有 230MB[1]。 ECU 之间通过各类总线连接在一起，电子系统又根据不同的功能划分成多个系统，包括底盘控制系统、动力控制系统和车身控制系统等。综上所述，智能汽车有如下特点[2]：

1、复杂分布式系统

在智能汽车中，百余个车载 ECU 之间是独立共存的，它们不存在主从关系而是由车内网络连接在一起，通过交互与协作来完成单个或者多个系统功能。因此智能汽车是一个典型的分布式系统。

2、异构系统

智能汽车的异构主要体现在网络上。ECU 的功能各不相同，对响应速度也有不同需求，因此汽车内部网络存在多种汽车总线。不同类型的汽车总线之间需要通过网关进行通信，同时动力 CAN 总线和舒适 CAN 总线之间也需要网关进行同步。

3、高实时性嵌入式系统

作为车内网络基础的 ECU 是典型的嵌入式系统。对于底盘控制系统和动力控制系统而言，ECU 的实时性和可靠性直接影响着车辆的安全性，例如防抱死系统和刹车系统等对 ECU 的实时性有着近乎苛刻的要求。

4、开放物理系统

安全性和舒适性一直是汽车生产商所关注的工程问题，为了使消费者享受更佳的驾驶体验，智能汽车通常搭载功能强大的娱乐系统和安全防护系统。这些强大的功能使得智能汽车不可避免的与外界进行广泛的数据交互。

1.1.2 国内外研究现状

1、国外研究现状

随着人工智能技术和通信技术的快速发展，辅助驾驶技术和无人驾驶技术等技术正加速从实验室走入消费者的生活之中。例如近两年，特斯拉和谷歌相继进行了无人驾驶技术的相关测试，沃尔沃推出的配有第三代城市安全系统的 S90 系列轿车也在汽车市场销售火爆，像宝马、奔驰、本田和奥迪等各大整车企业都拥有了比较成熟的自动驾驶技术。另一方面，车联网技术是目前解决交通问题的重要方案，在解决交通拥堵和交通事故上具有卓越贡献。汽车制造商、零部件提供商以及通讯设备制造商纷纷通力协作，联手开发以车与车信息交互为主的系统，美国和欧洲还把专业的通讯技术和移动蜂窝网络进行结合为一体，促成网协调统一[3]。诸多大型互联网企业都专注于智能汽车的开发，他们将人工智能技术，大数据技术、导航、通讯和多媒体这样一些信息融合发送至汽车的娱乐系统，使得消费者有着更好的驾驶体验。例如著名互联网公司谷歌和苹果都分别发布可关于智能汽车的驾驶操作系统，并且今年已经在若干车型上得到了应用。

2、国内研究现状

汽车产业的发展速度是衡量一个国家的工业发展水平的重要指标，智能汽车的快速发展必将促进各行各业的繁荣昌盛，因此我国十分重视智能汽车领域的发展。智能交通系统是我国交通运输领域的发展目标，智能汽车的发展可以有效的带动该领域的发展进程[4]。因此，我国政府对智能汽车的关注度在不断的提升。“十二五”期间，国家 863 计划对车联网的关键技术进行了研发，已经取得了初步的成果，基于车路交互的系统在实际道路上进行了应用实验。例如，国防科技大学和解放军军事交通学院，均完成了部分路段的无人车实验。工信部在发布的《中国制造 2025》中明确指出，到 2025 年我国将掌握自动驾驶的总体技术以及各项关键技术，建立比较完善的智能汽车研发体系、生产体系和产业群。所以在这种政策的市场的双重作用下，智能汽车的发展在我国发展迅猛，具有高速自动巡航、堵车辅助和自动泊车等功能的智能汽车不再是仅仅出现于学术研究领

域，目前已经逐步被消费者所熟知，并成为市场上的宠儿。国内传统汽车制造商纷纷通过合作、自主研发等方式加入了智能汽车研发领域，例如上汽乘用车与阿里巴巴合作推出了首款互联网汽车荣威 RX5。长安汽车与华为技术有限公司在车联网和智能汽车领域开展了多项深入的交流与合作。同时，部分互联网企业如百度和乐视等也耗费巨资力争在智能汽车市场上抢占先，并已经取得了较为可观的阶段性成果。

1.2 研究问题

1.2.1 网络安全

智能汽车的飞速发展给消费者带来前所未有的驾驶体验和安全性能的同时也面临着问题和挑战。智能汽车配备的各类无线设备（如 Wifi、蓝牙和蜂窝网络等）能为其增添丰富的功能，比如云端服务、空中固件升级和远程诊断等，而这些设备却是引发智能汽车网络安全问题的隐患。汽车内部网络对功能安全的要求极为严格，一旦出现故障就很有可能造成灾难性后果。智能汽车需要与其他车辆、互联网终端和基础服务设施进行大量的数据交换，这迫使汽车内部网络完全暴露于外界。当黑客通过逆向技术获得了汽车 ECU 的访问权限，可以利用该 ECU 控制连接在车内网络中的任何 ECU，对智能汽车的安全造成了极大威胁。随着车联网技术的不断拓展，智能汽车的网络安全问题会日益突出。针对智能汽车的网络攻击，目前主要存在以下两点途径[5]：

1、物理接入攻击

当黑客（可以为朋友、汽车维修工、汽车租客等）进入到汽车内部接触到其电子系统时，通过将预制的恶意设备连接到智能汽车的车载诊断（On Board Diagnostics, OBD）接口，该设备即可永久的连接到汽车的内部网络上。黑客可以随时向该恶意设备发送消息，以此方式来控制整个汽车电子系统。当消费者不经意购买到由黑客冒充设备供应商提供的恶意设备时，也会存在汽车可能被黑客控制的安全隐患。

2、无线接口攻击

智能汽车配有多种类型的无线接口，其中一些为类似 Wifi、蓝牙等短程无线接口，另一些为 4G 等近乎无限距离的无线接口。这些接口使车内 ECU 与外界网络相连，当黑客利用逆向技术通过此类接口获得车内 ECU 的访问权限时，即可向车内网络接收/发送控制信息从而对汽车进行操纵。2015 年 Miller 博士在全球黑帽大会上表示攻击者可以通过 3G 网络入侵切诺基吉普车的内部网络实现远程攻击[6]，包括控制汽车如引擎、刹车和方向盘等核心功能，迫使克莱斯勒公

公司在美国紧急召回 140 万辆可能受影响的汽车，引起业内广泛关注。

汽车作为重要的交通工具，安全始终是其最热门的话题。在汽车生产商不断提升功能安全和被动安全的同时，智能汽车的网络安全也需要得到同样的重视。安全作为一个整体，功能安全和网络安全两者相辅相成，任何一方面的缺失都会给智能汽车带来严重灾难。网络安全是智能汽车安全研究领域的新的分支，目前正处于初步阶段。在智能汽车网络安全的研究领域中抢得先机，缩小与欧美发达国家在汽车研发技术上差距，可以有效促进我国汽车产业的蓬勃发展。

1.2.2 网络体系结构

由于移动通讯技术、智能设备和应用服务的快速发展，信息技术与汽车的融合愈加密切。与传统汽车相比，智能汽车的网络体系结构发生了巨大变化。下面从整体网络结构和内部网络结构两方面来分析智能汽车网络体系结构的特点。

随着汽车电子系统与互联网的日益融合，汽车生产商已经开始独立研发汽车联网技术，例如通用汽车的安吉星系统或者宝马汽车的联网驾驶服务。这些联网技术已经不再是简单的通过无线通信技术连接到互联网，而是互联网与汽车的深度互联，即远方服务端可以进入汽车内部网络。Kleberger 将智能汽车的整体网络结构分成三部分[7]：

1、车辆本身，包含由大量 ECU 通过各类总线构成的车内网络；

2、可以为汽车提供各种服务的应用中心；

3、应用中心与车辆间的通讯链接；

图 1.1 为包含车辆本身、应用中心和通信链接的智能汽车整体网络结构。

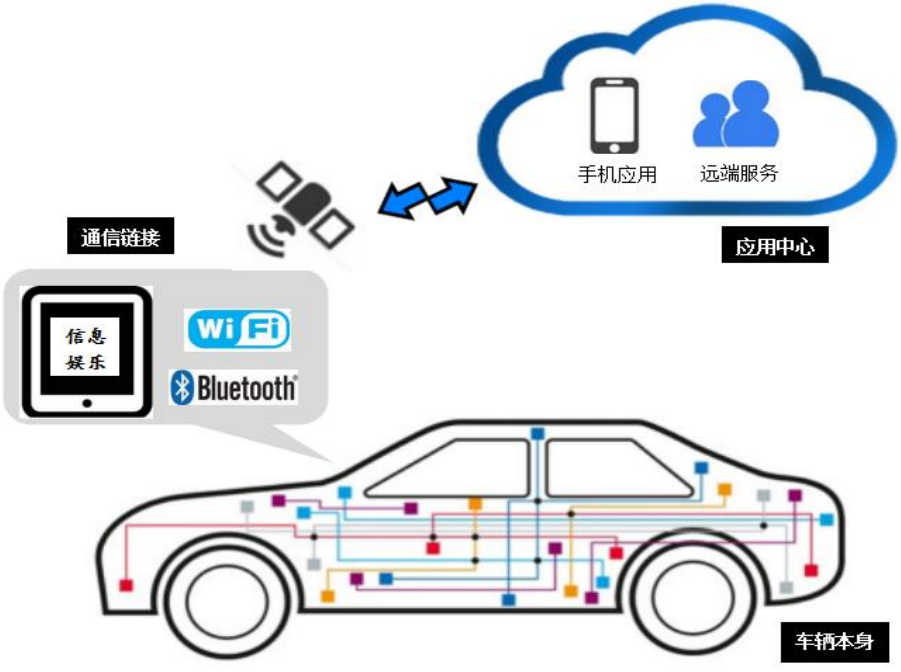

图 1.1 智能汽车整体网络结构

根据汽车各单元对控制速度和复杂性要求的不同，汽车内部网络可以进一步划分成不同的子网络，各子网络之间通过网关连通，如图 1.2 为智能汽车车内网络结构。例如，动力传动系统对网络响应速度要求很高，一般由高速、可靠的 CAN 总线或 FlexRay 总线[8]来实现。而车身控制系统，如空调、车灯和门窗等等不需要很高的实时性，因此可以采用如 LIN（Local Interconnect Network）总线[9]的辅助型总线。MOST（Media Oriented System Transport）总线[10]具有高速和高带宽的特点，主要应用于高档汽车上如多媒体系统、GPS 导航和车载电话等娱乐装置。

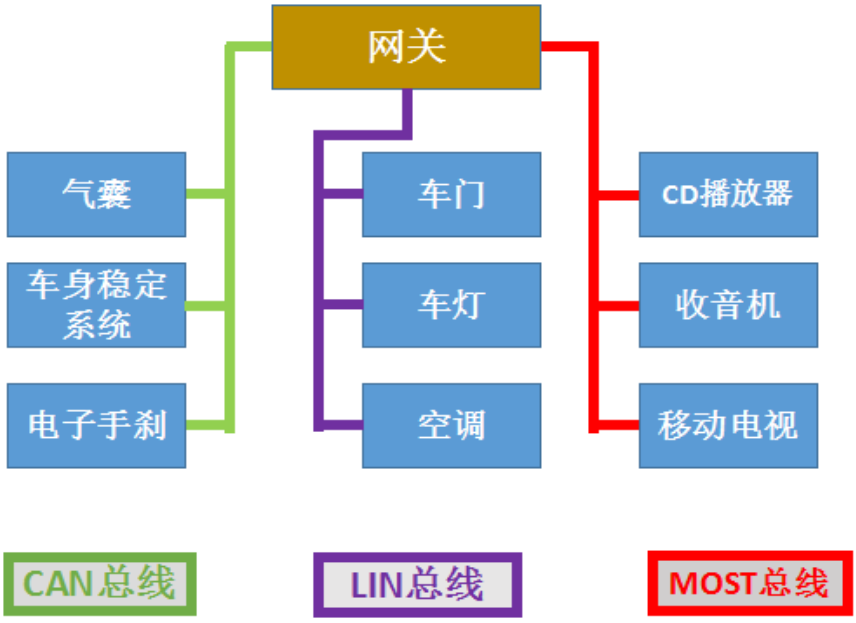

图 1.2 智能汽车车内网络结构

1.2.3 汽车总线安全

综合上述对智能汽车网络体系结构的分析可以发现，随着智能汽车的发展，外界网络与车内网络高度融合。然而，各类汽车总线的应用场景为封闭的局域网络，总线中不含有任何抵御外来网络攻击的功能，因此各类汽车总线安全性的缺失是造成智能汽车网络安全的重要原因。以 CAN 总线为例，汽车总线主要存在以下缺陷[5]。

1、广播性质

在 CAN 总线上的消息是以广播的形式传输的，当其中一个 ECU 发送数据帧时，CAN 总线上的任何 ECU 都可以监听到总线上的数据。因此当 CAN 总线上存在一个恶意节点，那么这个节点可以监听 CAN 总线上任意节点发出的消息，并且该节点可以向各节点发送消息。

2、易受拒绝服务攻击（Denial of Service Attack）

根据 CAN 总线的仲裁机制，拒绝服务攻击成为可能，以最高的优先级发送

数据，那么总线上的其他节点不再有控制权。同时，如果利用某 ECU 持续发送错误消息，根据 CAN 总线的错误检测机制，那么该 ECU 会被从总线上移除，使其失效。

3、弱认证

CAN 总线上的数据帧只通过 ID 进行标识，数据帧与 ECU 没有必然的联系，即接收端 ECU 不能够判断数据帧是否来由系统内部的 ECU 产生发送，这就使得重放攻击成为可能。当攻击者窃取到有效的数据帧时，攻击者可以通过某 ECU 持续发送这些数据帧来实现重放攻击。

1.3 研究内容与贡献

1.3.1 研究内容

通过上文从智能汽车整体网络结构和车内网络结构两方面特点进行深入的分析，发现汽车总线安全性的缺失是引发智能汽车网络安全问题的关键因素。本文着重从实时性、可靠性和防御能力角度出发，对安全汽车总线的实现问题进行深入研究，具体的研究内容概括为以下 2 个方面。

1、安全 CAN 总线协议的设计

CAN 总线是目前应用最广泛的汽车总线，因此对安全 CAN 总线协议的研究理应作为首要任务。对 CAN 总线安全性问题的分析可以发现，CAN 总线主要存在着三个缺陷，包括弱访问控制、无机密性保护和无可认证性保护。根据 CAN 总线的性质，使 CAN 总线具有较强的访问控制是很难实现的，但如果 CAN 总线协议具有机密性保护和可认证性保护可使 CAN 总线有效的抵御如消息窃取、消息修改和重放攻击等破坏方式。同时，嵌入式设备 ECU 对实时性有着很高的要求，安全 CAN 总线协议的设计必须在保证满足实时性的条件下进行。因此在高实时性嵌入式系统环境下提出高效的安全 CAN 总线协议，可以有效的解决智能汽车网络安全问题。

2、密码系统的攻击与防御

为汽车总线添加机密性保护和可认证性保护，目前的主流方法是将现代密码技术应用于汽车 ECU 上，与传统汽车总线技术相结合来实现安全汽车总线。其中，ECU 在密码学的应用过程中承担着重要使命，是密码系统的重要组成部分，它不仅需要对密码算法进行处理而且需要对密钥进行保护。但是本身作为嵌入式系统，很容易受到差分故障分析攻击（Differential Fault Analysis Attack, DFA）的威胁。因此安全汽车总线的实现同时需要对 DFA 攻击与防御进行研究，从而提升密码系统自身的防御能力。

1.3.2 本文贡献

本文的主要贡献及创新点总结如下：

1、基于 AES-CCM 的安全 CAN 总线协议

针对传统 CAN 总线的设计缺陷，现代密码学的应用可以使 CAN 总线具有机密性和可认证性能够有效抵御各种网络攻击。高级加密标准[11]（Advanced Encryption Standard, AES）具有加密速度快和安全等级高的特点，很适合应用于智能汽车的高实时性嵌入式系统中。CCM（Counter with Cipher Block Chaining-Message Authentication Code）模式[12]是一种全新的加密模式，能够够实现类似公钥密码算法的可认证功能。本章结合 AES 算法和 CCM 模式的特点，提出基于 AES-CCM 算法的安全 CAN 总线协议。与其他工作相比，该协议具有设计简单、安全性高且在高实时性的智能汽车电子系统环境下具有良好的性能表现。

2、一种针对 AES 解密的 DFA 攻击方法

安全汽车总线的实现离不开密码系统的支持，ECU 作为密码系统的重要组成部分，担负着密码算法处理和密钥保护的作用，但是很容易受到 DFA 攻击的威胁。通过对相关 DFA 攻击的研究发现，模型复杂、攻击效率低和实现难度大是 DFA 攻击方法的主要特点。同时，虽然 AES 加密与解密同等重要，但鲜有针对 AES 解密过程的 DFA 攻击研究，因此本文通过提出一种针对 AES 解密的高效 DFA 攻击方法。与其他工作相比，本文所提出的攻击方法，不但可以应用于 AES 解密过程，同时具有攻击模型简单、所需故障明文数量少和攻击速度快等特点，对智能汽车的 ECU 具有很强的破坏性。

1.4 结构安排

本文的结构如下：

第一章 绪论部分

首先介绍了智能汽车的概念及研究现状，然后对智能汽车的网络体系结构和现有设计存在哪些问题进行深入的分析和总结，最后简述了本文的研究内容及贡献，并列出组织结构。

第二章 研究基础和相关进展

从安全CAN总线协议和故障分析攻击两个不同的研究视角出发，概述了所涉及的研究基础，然后对相关的研究工作进行调研和综述，并对其中存在的不足进行了分析和总结。

第三章 基于AES-CCM算法的安全CAN总线协议

针对传统CAN总线的设计缺陷，本章提出基于AES-CCM算法的安全CAN总

线协议。根据CAN总线协议的特点，分别提出利用CRC场的方案一和利用扩展ID场的方案二，均使得传统CAN总线同时具有了机密性和可认证性，可以抵御多种类型的网络攻击。最后，利用2块飞思卡尔HCS12 开发板作为实验平台，对所提出的基于AES-CCM算法的安全CAN总线协议进行实现与评测。

第四章 一种针对AES解密的高效DFA攻击方法

ECU作为安全CAN总线协议实现的重要部件，很容易受到DFA攻击的影响，因此对DFA攻击的研究变得十分重要。AES的解密过程与加密过程同等重要，但是目前DFA攻击的研究往往集中于AES的加密过程。本章将DFA攻击扩展到AES的解密过程，并利用构建S盒分布表提出优化攻击方法来提高攻击效率。最后在VS 2010开发平台上实现所提出的DFA攻击方法，并在密钥空间和攻击速度等方面与前人工作进行了分析与比较。

结论和展望

对本文的工作以及创新点进行总结，并对今后的研究工作进行展望。

第2章 研究基础及相关进展

本文研究的安全汽车总线，根据视角不同分为安全 CAN 总线协议与故障分析攻击两个研究内容，本章分别对其研究基础和相关进展进行介绍。

2.1 安全 CAN 总线协议的研究基础与相关研究

2.1.1 CAN 总线协议

随着电子技术在汽车行业的飞速发展，汽车电子系统更加多样化、异构化和复杂化，传统的线束直连已经无法满足如此复杂的布线连接，总线技术由此引入汽车领域。CAN 总线，即控制器局域网络，由德国 Bosch 公司在 20 世纪 80 年代后期推出，是国际上应用最广泛的现场总线之一。 CAN 总线具有很高的实用性和可靠性，在汽车控制系统中有广泛的应用。同时，CAN 总线可以应用于其他领域。

CAN 总线协议是串行通讯协议，能够有效的支持具有高安全级别的分布实时控制。按传输速率可以将 CAN 总线分为低速 CAN 和高速 CAN，低速 CAN 速率低于 125kbps，高速 CAN 的速率处于 125kbps 和 1Mbps 之间。CAN 按标识符长度划可分成 CAN2.0A 和 CAN2.0B，前者标识符为 11 位，而后者扩展了 18 位，计 29 位。

CAN 协议共定义了四种不同的帧格式[13]：

1、数据帧：数据帧用于进行节点间的数据传输。

2、错误帧：当节点检测到错误时就会发送错误帧。

3、远程帧：某节点发送远程帧，请求发出具有相同标识符的数据帧。

4、过载帧：过载帧主要用于在连续的两个远程帧或者数据帧之间提供一段延迟。

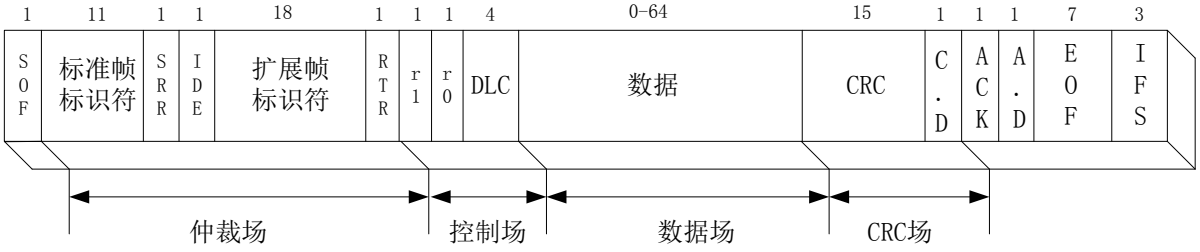

图 2.1 CAN 数据帧格式

如图 2.1 为 CAN 数据帧格式，共由起始符、仲裁场、控制场、数据场、CRC 校验场、确认场和帧尾组成。起始符 SOF（Start of Frame），用于表示 CAN 报文的开始，以一个显性位表示。仲裁场包含标识符和远程发送请求 RTR（Remote Transmission Request）位，用于表示消息的优先级和远程帧的开启判定。在标准格式中，标识符的长度为 11 位。而扩展格式中，标识符长度为 29 位，包括 11 位的基本 ID 和 18 位的扩展 ID，其中 IDE（Identifier Extension）定义了是否为扩展帧。控制场由 6 位数据组成，标准帧和扩展帧的控制场格式不尽相同。其中 r0、r1 作为保留位，DLC（Data Length Code）用于表示数据的长度。数据场由数据帧中发送的数据组成，按照高位优先进行传输。CAN 场包括 CRC 序列和 CRC 界定符。最后是 ACK 确认场和帧尾结束符。

CAN 总线网络中，所有节点均处于对等的关系，因此仲裁机制对于 CAN 总线的通信具有十分重要的作用。CAN 采用载波侦听多路访问/冲突检测的仲裁机制，其特点为在发送节点进行数据发送前对总线进行侦听，当总线处于未被占用的条件下进行数据发送，在发送的过程中同时接收本节点所发送的数据，当接收的数据与自身发送的数据不一致时则停止发送数据。这表明该节点已经失去仲裁，需要退出总线的访问进行等待。

2.1.2 高级加密标准

1、AES 加密

AES 是对称密码的高级加密标准，主要用于确保数据机密性、来源认证和数据完整性保护。AES 是一个迭代加密算法，共有三个版本。第一个版本包含 128 位的明文和密钥进行共 10 轮的变换。其他两个版本中，都是 128 位的明文输入，但是分别支持 192 位、256 位的密钥进行 12 轮或 14 轮的迭代变换。在变换过程中，16 字节的临时结果用 4×4 二维数组表示，通常称为状态组数。例如第 i 轮的状态数组 $\mathrm{M}^{i}=(M_{0}^{i},...,M_{15}^{i})$，可以由公式（2.1）表示。

$$\mathrm{M}^{\mathrm{i}}=\begin{pmatrix}\mathrm{M}_{0}^{\mathrm{i}} & \mathrm{M}_{4}^{\mathrm{i}} & \mathrm{M}_{8}^{\mathrm{i}} & \mathrm{M}_{12}^{\mathrm{i}}\\ \mathrm{M}_{1}^{\mathrm{i}} & \mathrm{M}_{5}^{\mathrm{i}} & \mathrm{M}_{9}^{\mathrm{i}} & \mathrm{M}_{13}^{\mathrm{i}}\\ \mathrm{M}_{2}^{\mathrm{i}} & \mathrm{M}_{6}^{\mathrm{i}} & \mathrm{M}_{10}^{\mathrm{i}} & \mathrm{M}_{14}^{\mathrm{i}}\\ \mathrm{M}_{3}^{\mathrm{i}} & \mathrm{M}_{7}^{\mathrm{i}} & \mathrm{M}_{11}^{\mathrm{i}} & \mathrm{M}_{15}^{\mathrm{i}}\end{pmatrix} \tag{2.1}$$

加密过程，每一轮都顺序进行字节替代（SubBytes）、行位移（ShiftRows）、列混合（MixColumns）和轮密钥加（AddRoundkey）4 个操作。但是在第一轮变换前，首先进行明文与原始密钥进行一次轮密钥加。在最后一次轮变换时，列混合操作被省略。字节替代、列混合和轮密钥加可以被定义为在有限域 $GF(2^{8})$ 上的数学函数，此有限域的多项式可以表示为 $\mathrm{X}^{8}\oplus\mathrm{X}^{4}\oplus\mathrm{X}^{3}\oplus\mathrm{X}\oplus1$，且加密过程中处理的数据均可以看作是此数据域中的元

素。

字节替代，状态数组中字节进行独立的非线性替换操作。通常利用置换表（S-Box）作用于输入的每个字节以此来得到输出。字节替代对 AES 加密算法的安全性起着至关重要的作用，可以有效的抵御各种数学分析方法。

行位移，状态数组中各行进行不同长度的补位位移。第一行保持不变，第二行左移 1 个字节长度，第三行左移 2 个字节长度，最后一行左移 3 个字节长度。

列混合，这里可以被看作是在 $GF(2^8)$ 有限域上，各列与特定二项式 $a(x)=3x^3+x^2+x+2$ 进行模 x^4+1 的乘法。

轮密钥加，即 16 字节的状态数组中每个字节与对应密钥数组中的每个字节进行独立的异或计算。

这里需要注意，如果改变轮密钥加、字节替代和行位移输入中的一个字节，那么结果中也会发生一个字节的变化。但是如果改变列混合输入中的一个字节，那么输出结果中会改变这个字节所在位置的一整列数据。

2、密钥扩展方案

AES 算法利用原始密钥根据密钥扩展算法来产生轮密钥[11]。密钥扩展算法法首先需要一组 N_b 字节长度的初始密钥，并且 N_r 轮操作中的每一轮都需要产生 N_b 字节长度的密钥数据，因此密钥扩展共产生 $N_b\times(N_r+1)$ 个字节长度的轮密钥。

轮密钥由线性数组构成，用 w_i 表示，i 的范围为 $0\le i<N_b\times(N_r+1)$。对于不同的 N_k 值密钥扩展算法会有所不同，N_k 的可能值为 4、6 和 8。当 N_k=4 或者 N_k=6 时，扩展方案如算法 2.1 所示。

算法 2.1 当 N_k=4 或 N_k=6 时的密钥扩展

输入：CipherKey，W

输出：W

```
1.  for(i=1; i<Nk; i++)
2.  W[i]=CipherKey[i];
3.    for(i=Nk; i<Nb*(Nr+1); i++)
4.    {
5.      Temp=W[i-1];
6.      if(i%Nk==0)
7.      Temp=SubWord(RotByte(Temp) ⊕Rcon[i/Nk]);
8.      W[i]=W[i-Nk]⊕Temp;
9.    }
```

CipherKey 表示原始密钥，长度为 N_k 的一维数组。W 则为存储轮密钥的一维数组。SubWord()函数是将输入字节根据 S 盒而置换产生输出字节。对于 RotWord()函数，则是将字[a0,a1,a2,a3]作为输入展开循环排列，返回字[a1,a2,a3,a0]。其中涉及到轮常数 Rcon[i]=(Rc[i], ‘00’,’00’,’00’)，而 Rc[i]则

表示在有限域 GF(2^8)中 x_{i-1} 的值 Rc[1]=1(即'01')，Rc[i]=x(即'02')·(Rc[i-1])=x_{i-1}，当 N_k=8 时，密钥扩展方案如算法 2.2 所示。

算法 2.2 密钥扩展方案

输入：Key, W

输出：W

```
1.  for(i=0; i<Nk; i++)
2.   W[i]=(key[4*i], key[4*i+1, key[4*i+2],key[4*i+3]);
3.   for(i=Nk ;i<Nb*(Nr+1)); i++)
4.   {
5.     temp=W[i-1];
6.     if (i%Nk==4)
7.        temp=SubByte(RotByte(temp))^Rcon[i/Nk];
8.     else if(i%Nk==4)
9.        temp=SubByte(temp);
10.    W[i]=W[i-Nk]^temp;
11.  }
```

相比与 $N_k \le 6$ 的扩展算法，$N_k > 6$ 的扩展算法在 i 为 4 的整数倍时，需要现将 w[i-1]进行 SubWord 函数变换，可以在密钥扩展中附加对部分字的 SubWord 函数变换，密钥扩展的安全程度得到提升。当 N_k=8 时，密钥相对较长，如果只对 Nk 整数倍的字节做 SubWord 函数变换，会使得 SubWord 函数变换的密度较稀，安全性低。

当进行轮密钥加时，密钥长度应与分组长度对应。因此第 i 轮的密钥与分组长度有关，并且由密钥扩展的字 w[Nb*i], w[Nb*i+1], ..., w[Nb*(i+1)]构成。图 2.2 描述了 N_b=4, N_k=4 时的轮密钥选择情况。

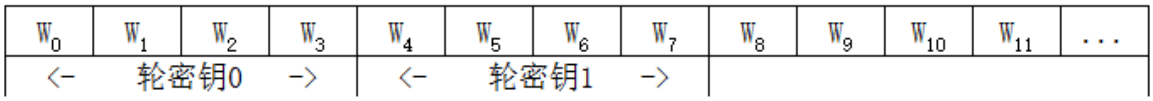

图 2.2 N_b=4, N_k=4 时的轮密钥选择情况

3、AES 解密

AES 解密过程与 AES 加密相似，利用密钥（密钥长度有 128bit、192bit、256bit 三种）将 128bit 长度的密文解密至 128bit 长度的明文。它总共执行 11 次轮迭代，每一轮顺序进行下面 4 个操作：逆行位移（Inverse ShiftRows）、逆字节替代（Inverse SubBytes）、逆列混合（Inverse MixColumns）和轮密钥加。

逆字节替代，是状态数组中每个字节依次进行独立的非线性置换操作。通常利用逆置换表（Inverse S-Box）作用于输入的每个字节以此来得到输出。

逆行位移，状态数组中各行进行不同长度的补位位移。第一行不进行位移，第二行右移 1 个字节长度，第三行右移 2 个字节长度，最后一行右移 3 个字节长

度。

列混合，这里可以被看作是在 GF(2^8)有限域上，各列与特定二项式 $a(x)=3x^3+x^2+x+2$ 进行模 x^4+1的乘法。

轮密钥加，即 16 字节的状态数组中每个字节与对应密钥数组中的每个字节进行独立的异或计算。

这里需要注意的是，AES 解密中每轮的 4 个操作除了轮密钥加，其余 3 个操作都是对应加密部分的逆操作。与加密相比，每轮的 4 个操作中首先进行的是逆行位移而不是逆字节替代。

为了与前人的工作进行清晰的对比，也为了便于展开 DFA 攻击，本文根据 AES 算法规范[11]中的方法，对密钥扩展函数进行修改，即使 AES 解密过程中每每轮依次进行的操作顺序与加密部分对应的操作顺序一致。所有的子密钥均由此修改过的密钥扩展函数产生，将原始密钥扩展至标准长度的子密钥。解密和加密类似，按顺序包含下面 3 个步骤：

1) 密文与原始密钥的异或计算；

2) 连续 9 次轮变换迭代；

3) 再进行一次轮变换，但是不包含逆列混合操作。

AES 解密过程如图 2.3 所示：

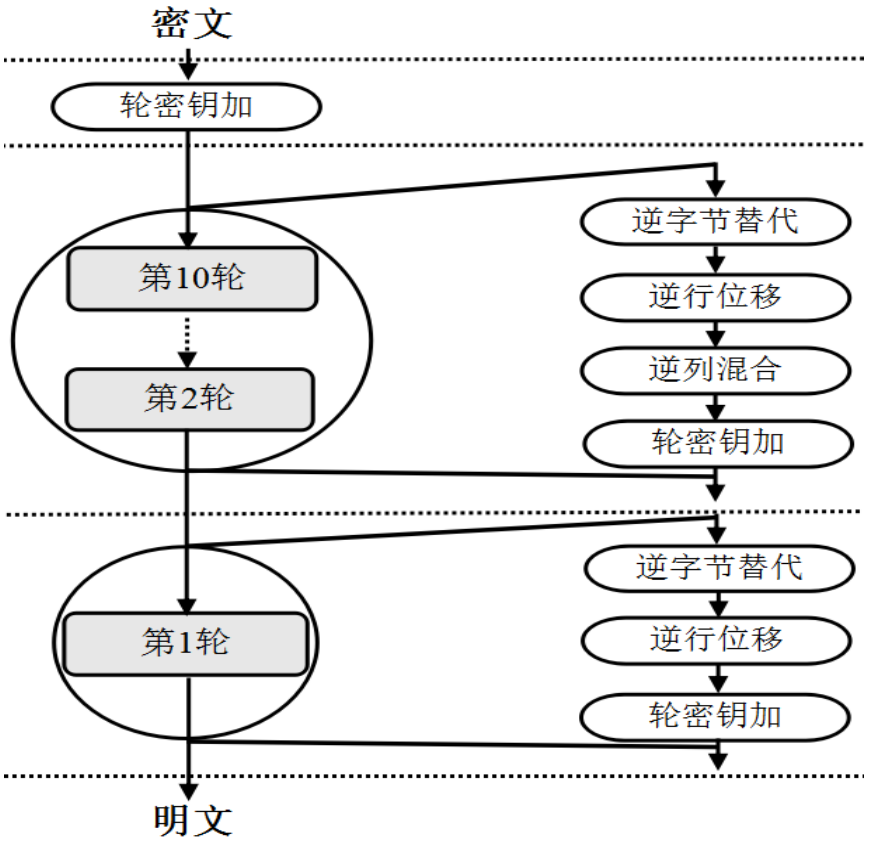

图 2.3 AES 解密过程

2.1.3 汽车网络安全的相关研究

智能汽车的快速发展给消费者带来极大便利，同时也带来了诸多网络安全隐患，引起了国内外学者的高度重视。

1、国外相关研究

2007 年来自波鸿鲁尔大学的 Wolf 等人展开了针对汽车网络安全的研究，分析了汽车可能遭受的网络攻击类型并提出了对应的防御措施[15]。华盛顿大学的的 Koscher 提出了特定的无线网络攻击技术，进而通过实际实验展示了智能汽车面临的网络安全问题。即当乘客的智能手机与汽车蓝牙设备相连接，并且在手机上下载安装了特定的恶意软件，那么展开短距离无线攻击变成可能[5]。Brooks 等人将可以接入汽车网络的通讯手段进行分类，进而利用 CERT 分类法来分析针对车载服务的攻击。分析结果表示当 ECU 中的固件进行升级时，需要特定的安全保护，并且需要提防由于智能汽车与汽车网络中心融合所带来的安全隐患[16]，如远程诊断服务。Schweppe 等人[17][18]建议利用 EVITA-HSM 建立安全通信结构，考虑到 CAN 数据帧有限的数据域，他们采用 32 位的消息认证码（Message Authentication Code, MAC），并指出由于车内网络的总线的特性，32 位的 MAC 可以抵挡住持续 35 周的碰撞攻击，因此是足够安全的。然而作者提出的安全架构过于抽象，既没有表述如何产生这 32 为的 MAC，也没有考虑到数据机密性和外部设备的连通性。为了保护车内网络免于遭受重放攻击，Groza 和 Marvay 提出一种类似 TESLA 协议具有消息认证功能的 CAN 总线协议[19]。Woo 展开了一次远程无线攻击车辆的具体实验，并根据 CAN 总线的特性性提出一种安全 CAN 总线协议，结果显示该协议有着较好的通信延迟和负载的性能表现[20]。

2、国内相关研究

2010 年王亚猛于吉林大学提出基于 Openssl 的车载网关认证系统用于对车内网络数据进行保护，但是其系统对资源需求极大，不适合应用于资源受限的车载电子环境[21]。Lin 等人在 2012 年利用 ID 表、消息计数器和双向对称密钥提出 MAC 生成技术[22]，次年提出优化混合整数线形设计公式来保证信息安全和和功能安全，不过两篇文献中均没有考虑到数据机密性[23]。中国科学技术信息研究所的王喜文通过研究日本推出的汽车信息安全模型，介绍了针对汽车电子系统有哪些攻击途径和不同汽车功能模块的信息安全对策，但是文章并没有提出具体的防护手段[24]。Wang 提出可以用于保护汽车电子系统的框架，该框架有良好的兼容性并且可以有效抵御消息篡改，但是数据机密性问题仍然没有解决[25]。

2.2 差分故障分析攻击的研究基础与相关研究

2.2.1 AES 加密模式

对称密码已经广泛应用于数据的保密和完整性验证。不同的分组密码在具体运用时，都会选择一种具体的加密模式。加密模式的主要目的是解决密钥的产生和使用，通常是基本密码、反馈和数学运算的组合。这些运算是简单的，因为加密算法的安全性主要依赖于密钥而不是依赖于加密模式。加密模式不会损坏算法的安全性。常见应用于 AES 加密算法的加密模式有[26]：电子密码本模式（ECB）、密码分组链接模式（CBC）、密码反馈模式（CFB）、输出反馈模式（OFB）、计数器模式（CTR）和 CCM 模式下面将逐个进行介绍。

1、电子密码本模式

图 2.4 和图 2.5 分别为电子密码本加密和电子密码本解密。在 ECB 加密模式中，明文被分割成分组长度相等的块，然后每个块独立进行加密。该模式具有可并行计算、速度快和故障不会传递等优点，但是存在明文长度必须是分组长度倍数和相同明文块对应其相同密文快的特点，在很大程度上影响了分组加密的安全性。一般此模式适用于较短的明文加密。

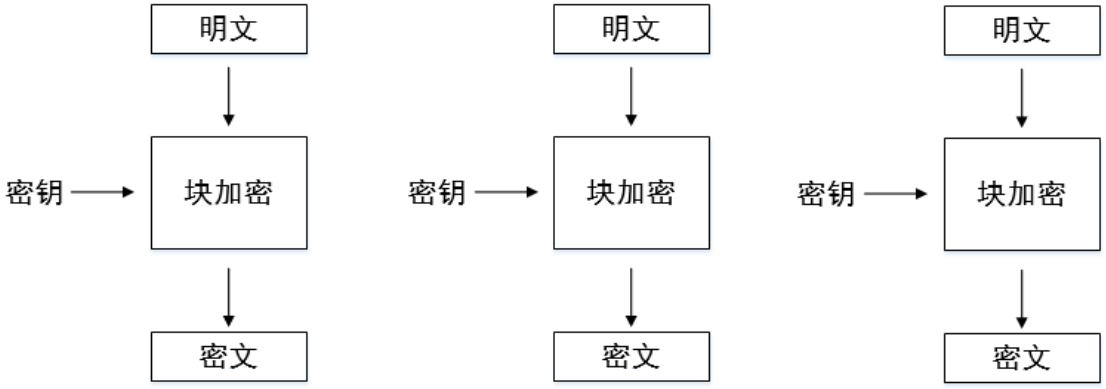

图 2.4 电子密码本模式加密

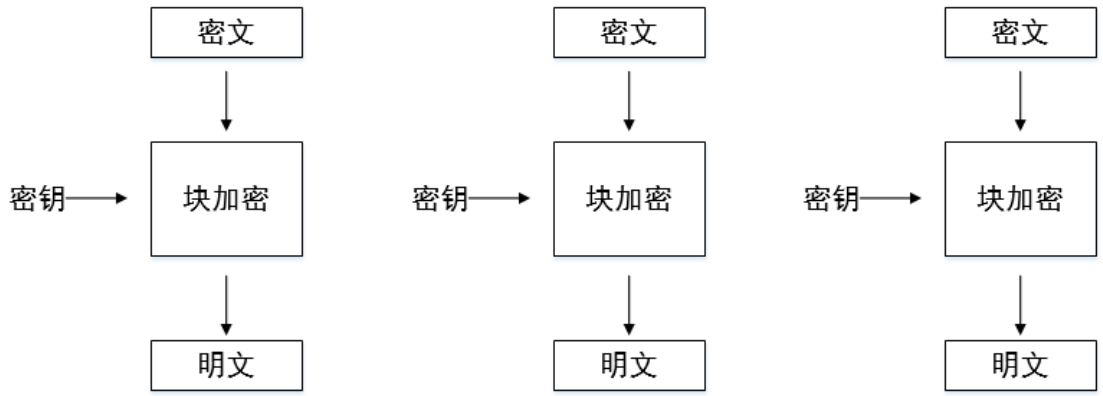

图 2.5 电子密码本模式解密

2、密码分组链接模式

图 2.6 和图 2.7 分别为密码分组链接模式加密和密码分组链接模式解密。在 CBC 模式下，每个明文块在加密前与上个密文块进行异或操作，密文块依赖于此前所有的明文，同时在第一个明文块被加密前必须采用初始化向量 IV。可以用 $C_0=IV$ 和 $C_i=E_k(P_i \oplus C_{i-1})$ 来表示 CBC 模式的加密过程。同理用 $C_0=IV$ 和 $P_i=D_k(C_i) \oplus C_{i-1}$ 表示 CBC 模式的解密。

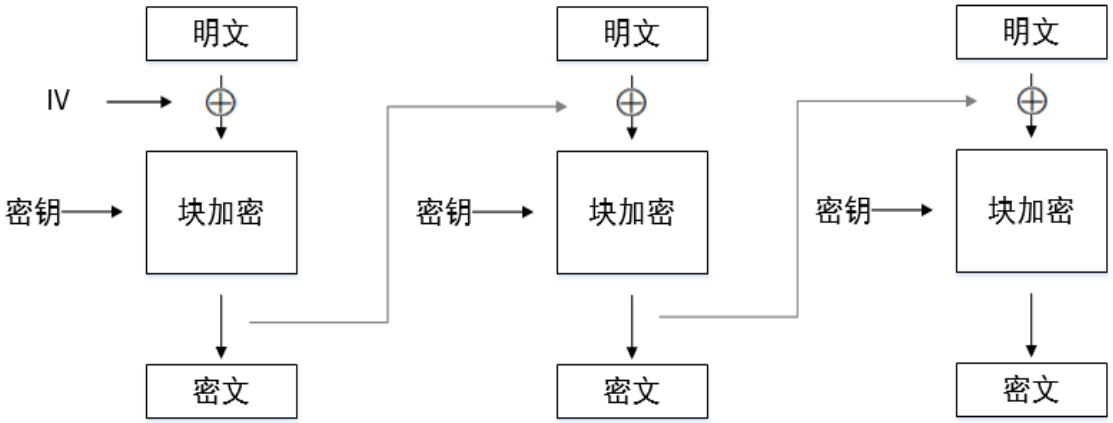

图 2.6 密码分组链接模式加密

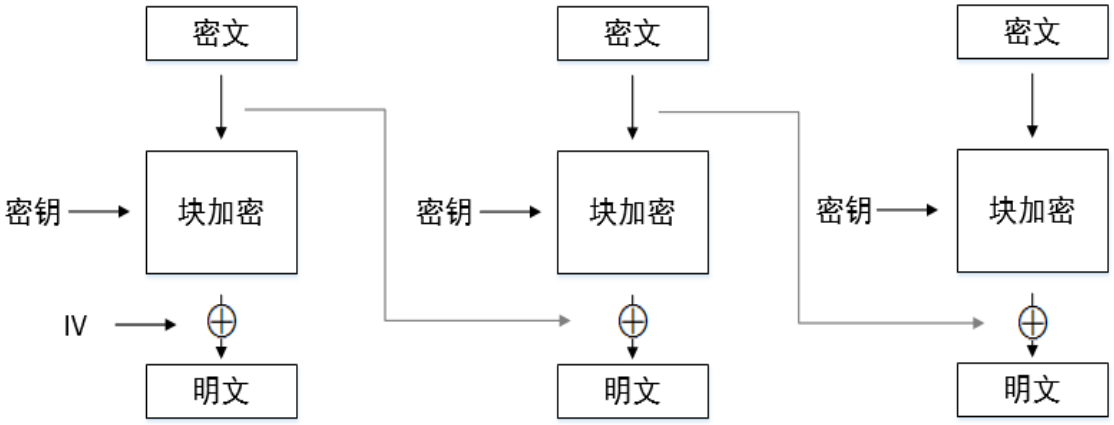

图 2.7 密码分组链接模式解密

CBC 模式是常用的一种加密模式，它的主要缺点是分组的加密是连续的，因此不容易并行加密，而且明文的长度必须补位到分组长度的整数倍。

在解密过程中，如果 IV 是错误的，那么明文的第一个分组将会是错误的，但是后续的明文块将是正确的。这是因为每个分组与之前的密文，而不是明文进行异或。这就意味着一个明文分组可以由两个毗邻的密文组经过计算而得到，因此解密是可以并行处理的。

3、密码反馈模式

图 2.8 和图 2.9 分别为密码反馈模式加密和密码反馈模式解密。CFB 模式与 CBC 模式相似，它将一个分组加密变成了具有自同步功能的流加密。明文中的

故障可以永久的传播到密文中去，而且加密过程不能并行。同样，CFB 模式在解密过程可以并行。在解密过程中，密文中的一个位故障可以影响到两个明文块，对应的明文分组的一个位故障和前一个明文块的彻底破坏，而后面的明文块则会正常解密。CFB 模式的优势在于块加密只在加密方向采用，而明文不需要补位到分组大小的整数倍。

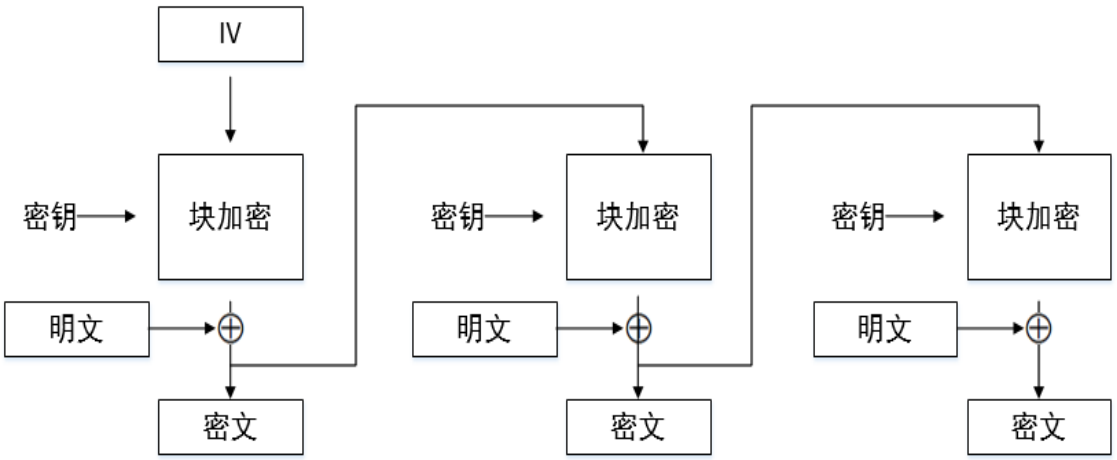

图 2.8 密码反馈模式加密

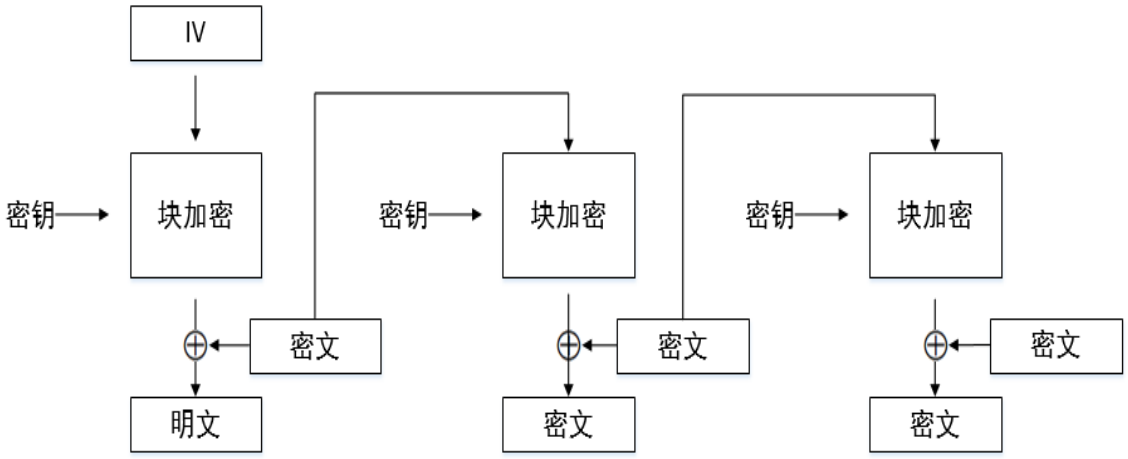

图 2.9 密码反馈模式解密

4、输出反馈模式

OFB 模式使分组密码成为可以自同步的流密码。它产生密钥流分组，之后与明文块进行异或产生密文块。和其他流密码的特性一样，在密文中改变一个比特位状态可以使得对应明文位置的比特位状态发生改变。在加密和解密的两端，分组算法都以加密模式使用，这种方法有时也叫内部反馈，因为反馈机制独立于明文和密文存在。如图 2.10 和图 2.11 所示为输出反馈模式加密和输出反馈模式解密。

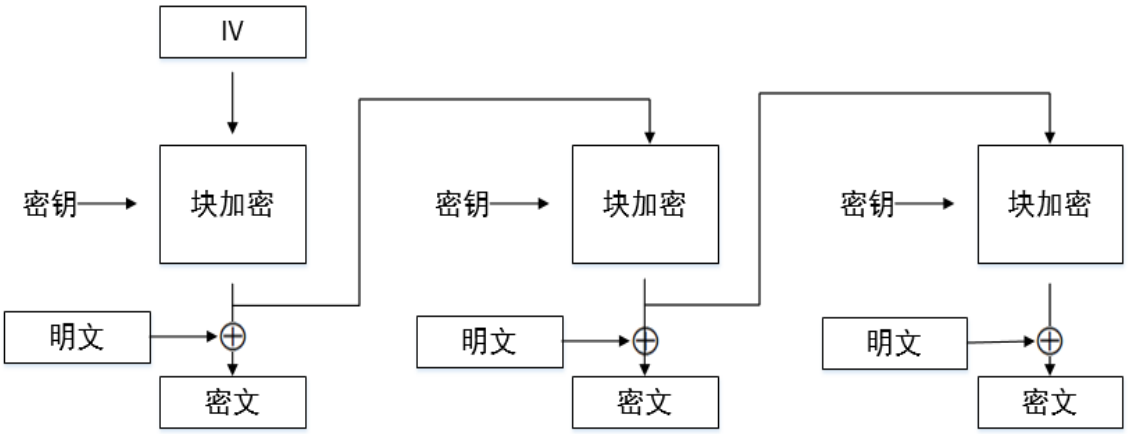

图 2.10 输出反馈模式加密

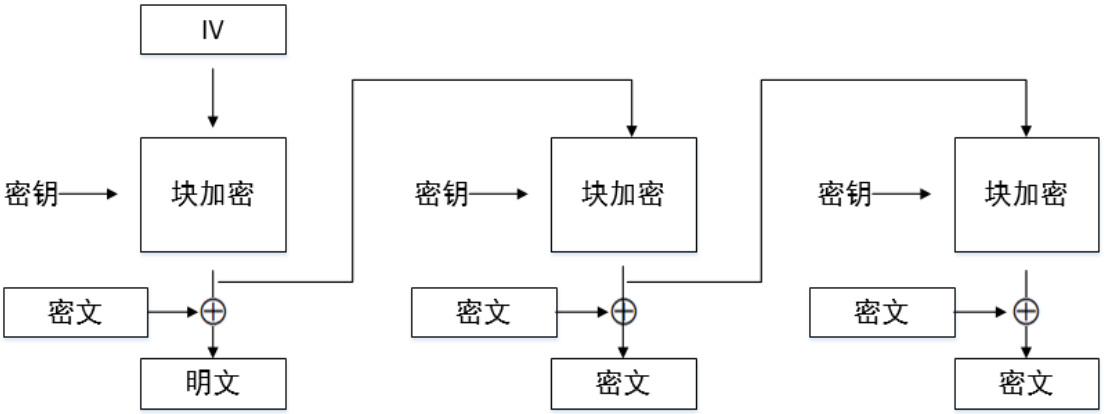

图 2.11 输出反馈模式解密

5、计数器模式

图 2.12 和图 2.13 分别为计算器模式加密和计算器模式解密。CTR 模式通过加密连续的计数值来产生密钥流，这些计数值可以由任何一种函数产生，但实际应用中采用最简单的不重复的自增序列。采用随机数 Nonce 与计数值结合可以产生用于加密的唯一计数分组。简单的将随机数与计数值进行异或在选择明文攻击的很多情况下会降低算法安全性，因为攻击者可以操作整个随机数-计数值来产生碰撞。

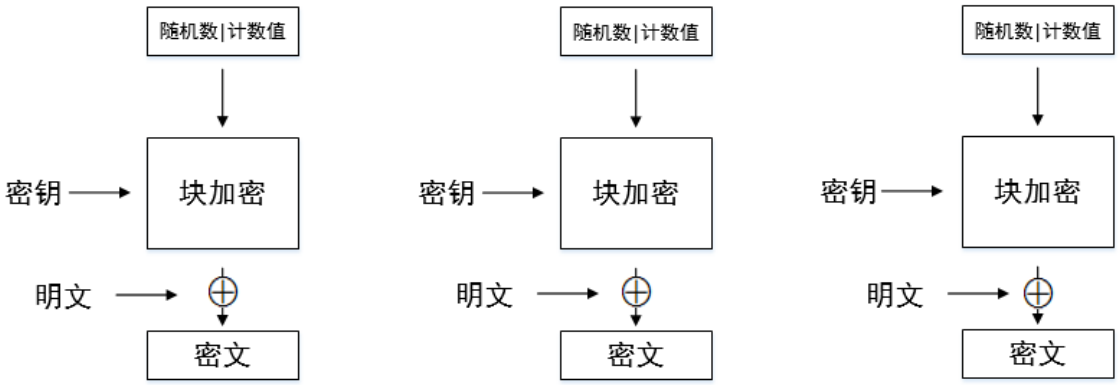

图 2.12 计数器模式加密

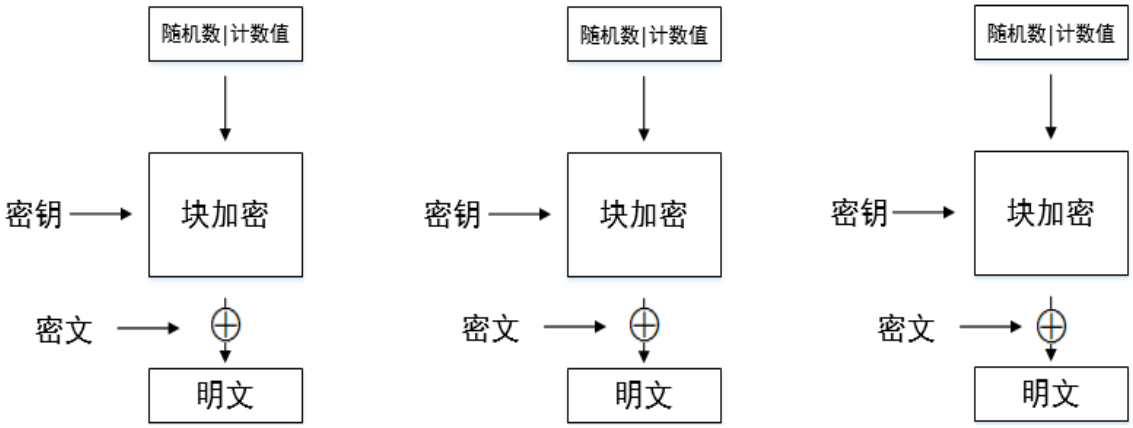

图 2.13 计数器模式解密

6、CCM 模式

由 NIST 在 2004 年提出的 CCM 模式是一种通用的分组加密模式，但它仅可以应用在分组长度为 128 比特位的加密算法，如 AES 加密算法。因此 CCM 模式不可以应用于 3-DES 加密算法[27]，因为它的分组长度为 64 比特位。相比比于其它加密模式，CCM 模式有很强的优势，它不但可以提供加密功能而且可以提供鉴别功能。其中加密功能由计数器模式（Counter Mode）提供，而鉴别服务由 CBC-MAC（Cipher Block Chaining-Message Authentication Code）模式提供，算法产生的 MAC 可以提供比校验和与错误检测码更强有力的可认证保障，校验和与错误检验码仅用来检测意外的传输故障，而 CCM 算法中的 MAC 则用于检测内部未授权的修改，同时也可以检测意外传输故障。可以理解为 CCM 模式结合了 CTR 模式与 CBC 模式两者各自的优点。CCM 模式还具有同步性和一定程度的并行性（在加密过程具有并行性，而在鉴别过程没有），同时故障不会被传播。

与其他加密模式一样，CCM 模式在数据进行加密之前必须建立一组唯一的密钥。因此，CCM 模式应该在一个良好设计的密钥管理结构下实现，其安全性取决于密钥的长度。CCM 模式适用于数据帧环境，因为数据帧在加密之前都已经准备就绪。相反，CCM 模式不适用于处理部分数据和流数据的处理。CCM 模式的输入包含 3 个部分：有效载荷，其数据应该既被加密又被认证；相关信息，它是一个消息头，应该具有可认证性，但是不需要加密；随机数，用于分配给有效载荷和相关信息。

在相同密钥条件下，CCM 模式要求对每组消息采用不同的随机数 Nonce。通常采取的方案是依次为每组消息编号，并且将这个编号作为随机数 Nonce。按序编号的消息可以检测重放攻击。因此，在很多情形下，依次排序的方法都是有效的。

通常 CCM 模式主要存在两个参数需要用户进行设定。第一个参数为鉴别段 M 的长度，鉴别段越长 CCM 模式所能提供的安全性就越强，但是当消息中搭载

过长的鉴别段则会使有效的数据空间受到限制，因此用户需要在消息扩展程度和被黑客攻击可能性之间进行权衡，其有效参数为 4、6、8、10、12、14 和 16 字节。第二个参数为有效载荷的长度，有效参数为 2 字节至 8 字节。有效载荷与随机数共享一段长度，提高有效载荷长度的同时，影响安全级别的随机数长度就会受到影响，用户可以根据需要在最大载荷长度和随机数长度之间权衡得到最优的参数配置。

2.2.2 差分故障分析攻击技术

传统的分组密码分析攻击从算法的数学结构入手，利用算法本身的数学和统计学特性来进行分析。但是目前主流的分组密码都很安全，已经很难从数学分析上对算法产生破坏。与此不同，侧信道攻击是一类新兴的密码分析方法。侧信道攻击不在关注密码算法本身的数学结构，而是利用密码设备工作时产生的侧信道信息来进行辅助分析，从而提高攻击效果。如图 2.14 为典型的侧信道攻击示例。

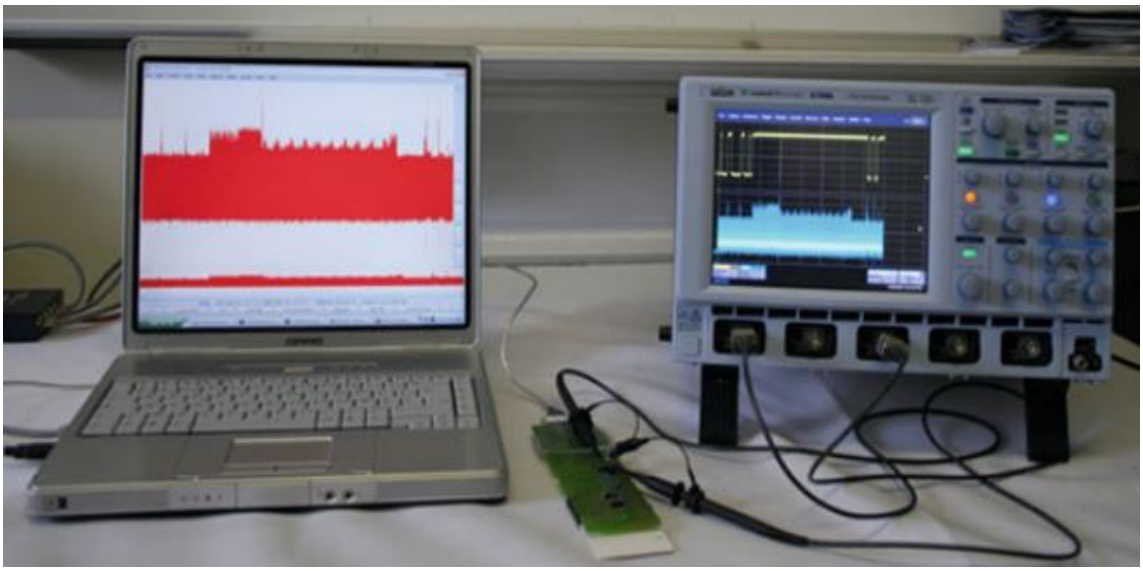

图 2.14 典型的侧信道攻击示例

自 1995 年 Kocher 提出第一种侧信道攻击以来[28]，各种新兴的侧信道攻击击方法逐渐涌现出来。例如：时间攻击、差分功耗攻击和差分电磁攻击等。其中的差分故障攻击分析攻击有着十分强大的破坏力，引起了学术界和工业界的广泛关注，下面将对其相关内容进行简单的介绍。

1、基本原理

差分故障分析攻击是侧信道攻击的一种。它是指密码设备在正常工作过程中诱导设备产生故障，通过对加密算法性质的利用，以此来获得密钥信息的技术。差分故障分析攻击有两大步骤。第一步就是要诱导密钥设备产生侧信道信息，这个过程称之为故障注入。故障注入一般是将密码设备置入不正常的工作环境中。

例如，当密码设备处于高温、强电流、高超频和强辐射等特殊条件时，处理器的操作可能会受到影响，产生故障结果。第二步就是利用采集到的侧信道数据进行分析。故障攻击的分析方法一般首先假设一个故障模型，而这些故障模型其实就是对侧信道信息的简单化和形式化的描述。在此之后就可以结合算法的性质和它的实现方式来逆向推导当前加密设备正在进行哪种操作，甚至得到其内部数据状态。

2、故障注入方法

故障注入是进行差分故障分析攻击的第一步，故障注入的结果直接影响故障模型、攻击方法和故障利用复杂度等。故障注入的方法有很多种，根据作用效果可以分为全局型的故障注入和局部型的故障注入，瞬时的故障注入和永久的故障注入。根据对密码系统的破坏情况分为入侵式故障注入和非入侵式故障注入。本节将对主流的故障注入方法和特点进行介绍[29]。

1) 电源和时钟故障注入

电源和时钟故障注入是指为密码设备的电源或时钟提供不正常的数据，例如过高或者过低的电压、电流或者时钟毛刺等。电源和时钟的故障注入是一种全局型的注入方式，故障影响整个加密设备的电路。故障注入可以在时间上有着准确的控制，但是无法对某一运算单元单独注入故障。由于完成电源和时钟注入需要的成本低，因此获得了广泛的应用与研究。

2) 电磁故障注入

电磁故障注入是指在加密设备的芯片上施加瞬间的强电磁场或者特定的谐波来影响电路的运转来注入故障。相对于电源和时钟注入，电磁注入可以控制注入的空间位置，产生局部故障。

3) 激光故障注入

激光注入是一种广泛应用的注入技术，激光故障注入需要十分昂贵的设备和更专业的操作，但此种故障方法往往可以获得很好的空间精度，获得一些全局故障注入无法获取的信息。激光故障注入技术采用很强且精确聚集的激光束对芯片进行故障注入。对晶体管进行激光照射可以其上的电介质形成短暂的电流通路，这样就可以精确的控制晶体管的状态。激光故障注入有着很高的空间和时间精度，可以较容易的对密码设备的芯片进行故障注入。

3、故障模型

故障模型是故障注入和分析攻击的中间桥梁，在故障分析攻击技术中具有十分重要的地位，也是当前的研究热点。故障模型不仅是故障注入的形式化描述，而且是后续分析攻击的前提。由于故障注入技术和攻击目标的多样化导致实际产生的故障类型数量巨大，在对特定算法进行攻击时，攻击者会根据算法的特性选取有利的攻击模型。因此有必要对故障模型进行系统的总结和归纳。故障模型可

以从不同的方面来划分，最终使用的故障模型则是综合这些不同的方面进行选定。下面将列出故障模型中常涉及的几个方面[29]。

1) 故障注入在算法的状态数组中还是在密钥扩展过程。故障注入在状态数组是指在算法进行加密或者解密的计算过程中，攻击者进行故障注入，最终导致计算过程出现故障，而故障注入本身并不影响密钥扩展的计算过程。

2) 故障注入影响的范围大小。一般有单比特位、单字节、单字，对应也有多比特位、多字节、多字的故障。

3) 受故障影响单元的分布状态。一般分为均匀分布和非均匀分布。例如通过将密码设备置入非工作电压的环境下而产生的故障，故障的分布往往是有偏差的非均为分布。

4) 故障注入的轮数。分组密码大部分采用的是迭代的方法进行加密和解密，故障的注入通常发生在具体的某一轮中。故障注入在不同轮，后续采用的分析方法也不尽相同。

5) 故障发生的位置。根据后续故障分析的需要，可以假定故障发生在每一轮的固定位置，或者假设故障发生的位置是随机的。

6) 是瞬时故障还是永久故障。对于瞬时故障是指故障的注入只影响本次加密或解密的结果，其后续的结果是正确的。相反，永久的故障是指攻击者破坏了电路本身，后续的每次计算结果都是错误的。

4、故障分析方法

差分故障分析攻击方法由 Shamir 等人针对 DES 算法而提出[30]，之后在针针对分组密码甚至流密码的攻击中都得到了广泛的应用。差分故障分析攻击方法对故障模型要求宽松。差分故障分析攻击中，对侧信道信息的收集是指故障注入后产生的密文，利用算法的数学性质通过对比正确密文和故障密文的差值来完成攻击，恢复密钥的部分或者全部值。差分故障攻击主要采用了数学分析攻击中的差值攻击的思想，但是差分故障攻击是在中间过程中引入的差分值。对 AES 算法进行差分故障分析攻击为当前的研究热点。AES 算法是目前分组加密的高级标准，安全性很高，也因此得到了广泛的应用。AES 算法的分组长度为 128 比特位，而密钥的长度为 128、192、256 比特位。根据 AES 密钥扩展的性质，如果攻击者获得了 AES 算法的最后一轮密钥，那么就可以完整的逆向推导获得 AES 算法的原始密钥。

下面通过一个简单的例子来描述差分故障分析方法。假设一个随机单比特故障 e 注入在 AES 加密过程的最后一轮输入的位置，且故障发生在状态数组第 1 个字节上，经过字节替代、行位移和轮密钥加操作得到故障密文，如图 2.15 所示。

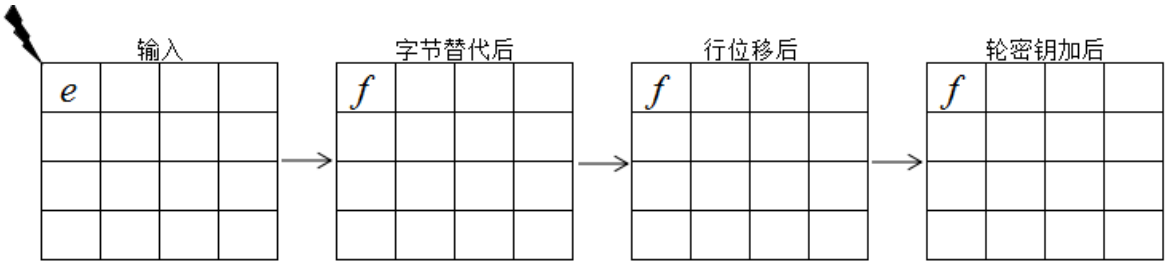

图 2.15 故障发生在第 10 轮的错误扩展图

如果用SB( )来表示字节替代，K_0^{10}表示第 10 轮密钥的第 0 个字节，C*表示故障密文。那么可以根据 AES 加密的数学性质，可以得到公式组(2.2)。

$$\begin{aligned} C &= SB\left(M_0^{10}\right)\oplus M_0^{10} \\ C^* &= SB\left(M_0^{10}\oplus e\right)\oplus K_0^{10} \end{aligned} \tag{2.2}$$

用$SB^{-1}( )$表示逆字节替代，根据公式组(2.2)推导出公式(2.3)。

$$e = SB^{-1}\left(C\oplus K_0^{10}\right)\oplus SB^{-1}\left(C\oplus f\oplus K_0^{10}\right) \tag{2.3}$$

其中 e 的可能值为 0x01、0x02、0x04、0x08、0x10、0x20、0x40、0x80，而 f 和K_0^{10}的可能值为 0x01 至 0xff。对变量 e、f 和K_0^{10}进行遍历，筛选出满足等式的K_0^{10}值。对状态数组中的任意字节，利用 3 个故障密文能以 97%的概率恢复密钥字节。因此对于 AES-128 算法，共需要不到 50 个故障密文即可恢复 AES 算法的完整密钥[31]。

2.2.3 故障分析攻击的相关研究

故障分析攻击是侧信道攻击的一种，对嵌入式密码系统有着极大威胁。目前对故障分析攻击的研究已有二十载，国内外研究硕果累累。

1、国外相关研究

1996 年，Boneh 提出一种针对智能卡通过注入故障来破解密钥的新型方法，这种攻击方法对现代密码的安全展现出很强的破坏性，引起了人们的广泛关注[32]。自此之后，故障分析攻击在密码分析中处于十分关键的位置，大批学者者加入到此领域的研究之中。在 1997 年，Biham 正式提出 DFA 的概念，并将其应用在 DES 算法上[30]。2000 年，Rijndael 被选为高等加密算法(AES)。虽然学术界已经开展相关工作用于保护 AES 算法[33-35], 但是仍有很多将 DFA 攻击成功应用在 AES 算法上的研究。Chien-Ning 等人在 2002 年开展了针对 AES 算法的 DFA 攻击研究，在智能卡加密过程中注入故障，使之发生比特级故障，产生故障密文。利用 AES 的性质，分析比较正确密文和故障密文的关系，逆向推导出 AES 密钥[36]。但是由于比特级的故障注入很难实现，随后他们通过改进攻击算法将故障类型扩展至字节级，提高了攻击可行性[37]。2003 年，Blomer 描述一种方法，需要 128 至 256 个故障密文来破解最后一轮轮密钥[38]。Giraud 称利

用注入的 50 个字节级故障可以破解轮密钥[31]。Dusart 等人在第 9 轮 ShiftRows 变换后注入故障字节，大概需要 40 个故障密文来恢复正确密钥[39]。Piret 和 Quisquater 在第 8 轮和第 9 轮之间注入故障，并且只需要 2 个故障密文即可获得密钥[40]。Saha 描述一种支持多字节故障的攻击方法，并且只需要 1 个故障密文文就可以破解密钥。当故障全部被注入到第 8 轮输入状态的四个“对角线”之一，密钥空间被缩减至 2^{32} 大小[41]。Mukhopadhyay 通过利用 AES 在密钥扩展过程的性质，得到 2^{8} 的密钥空间[42]。

2、国内相关研究

相对于国外的研究，国内的研究起步较晚。2005 年来自于浙江大学的叶世芬开始从事于对 DFA 攻击的研究[43]，2006 年杜育松针对 128 位 AES 算法进行行 DFA 攻击方法的研究，他提出了攻击算法并分析了算法成功的概率[44]。刘祥忠在 2012 年提出分别针对 AES 第 7 轮和第 8 轮的两种 DFA 攻击方法[45]。2013 年来自复旦大学的孙维东分析了 AES 和 DES 对故障分析攻击的安全性，描述一种仅需要 20 次故障注入就可以获取完整 AES 密钥的攻击方法[46]。

2.3 小结

本章从安全 CAN 总线协议和差分故障分析攻击两方面对所涉及的研究基础进行简单概述，同时介绍了对应的最新的研究进展。

第3章 基于 AES-CCM 算法的安全 CAN 总线协议

智能汽车上通常搭载百余个 ECU 通过各类数据总线连接在一起，如 CAN、LIN、MOST 和 FlexRay 总线，其中 CAN 总线应用最为广泛。然而，CAN 总线不具备数据的机密性和可认证性，连接在总线上的恶意节点可以监听总线上的所有消息，甚至可以向其他节点发送恶意消息。因此，本章提出基于 AES-CCM 算法的安全 CAN 总线协议，使传统 CAN 总线具有机密性和可认证性，可以抵御来自外界的各类网络攻击。

3.1 设计前提

针对智能汽车特殊的应用场景，安全 CAN 总线设计需保证以下 3 个前提。

1、 安全协议不能增添过大的载荷。如图 2.3 所示，CAN 数据帧的长度为 16 字节，其中数据域只有 8 个字节。对目前的 CAN 总线来说，这 8 个字节已经有很高的利用率了。因此，安全协议所需的额外载荷越少越容易被实现。例如，为了实现消息的可认证性，需要产生一个 MAC，并将其随着原始消息同时发送出去。为了确保足够的安全性，MAC 的长度至少为 4 个字节。如果将每个数据帧中加入完整的 4 字节 MAC，则 CAN 数据帧的有效载荷将减小一半。因此，该方法不符合智能汽车的应用场景。

2、安全协议不能依靠很强的计算能力和存储能力。通常的车载 ECU 都是具有有限资源的微控制器，如果设计中需要大量的数据计算和存储，普通 ECU 将难以在保证实时性的前提下完成任务。例如非对称加密算法 RSA 算法不但具有很强的机密性而且具有认证性[47]，非常符合本章的设计目的，但是受到 ECU U 有限资源的限制故不能应用于 CAN 总线上。

3、 安全协议不能进行任何硬件修改。若对安全协议进行硬件修改，则会使成本增加、可移植性降低以及设计复杂化。安全协议的实现应该在传统 CAN 控制器和 CAN 接收器的硬件基础上对协议本身进行修改。相比在 EVITA 项目中提出在汽车设计时增添硬件安全模块的策略[17]，单纯的协议修改更容易被汽车车制造商所接受。

3.2 设计思路

通过总结当前存在的针对智能汽车的网络攻击类型[5]，Samuel Woo 指出通通过修改 CAN 总线等车内网络总线，使其具备机密性和可认证性即可抵御各种类型的网络攻击[20]。

实现 CAN 总线的机密性和可认证性，ECU 发送节点和 ECU 接收节点之间需要实现以下 4 点：

1、 ECU 发送节点与 ECU 接收节点之间必须以密文形式通信；

2、 包含 MAC 的数据帧只能由可信任的 ECU 发送节点生成；

3、 ECU 接收节点可以验证 MAC 的正确性；

4、 不能存在相同值的 MAC。

根据章节 2.1.2 和章节 2.2.1 分别对 AES 算法和 CCM 加密模式的介绍，相比于其他算法和模式，AES 算法与 CCM 模式的结合不但可以满足设计中对机密性和可认证性的要求，AES-CCM 算法还具有加密速度快、安全性能高、计算量低和所需存储空间小的优势。本着设计简单、高效和易实现的初衷，本章所设计的安全 CAN 总线协议将保持传统 CAN 总线的帧格式。

从机密性方面考虑，AES-CCM 算法可以利用 128 比特位的密钥来产生密文，其密钥长度在当前的计算能力下很难被破解，因此密钥的安全性可以得到保证。但是 AES-CCM 算法是分组加密算法，其分组大小为 128 比特位，而 CAN 总线的数据场最大只有 64 比特位大小，因此本章采用补位的方案。即当所需加密数据小于 128 比特位的时候，进行补零操作以满足 AES-CCM 算法的分组长度需求。

从可认证性方面考虑，Luk 和 Karlof 在其无线传感器安全通信的研究成果中表明，用 MAC 替换 16 比特位的 CRC 域既可以提供数据完整性也可以提供数据可认证性[48][49]，即 MAC 可以检测数据帧中的恶意数据破坏也可以检测数据数据传输错误。因此，CRC 场完全可以被 MAC 场取代。Handschuh 指出 16 位的 MAC 不能够使数据处于一个有效的安全级别，32 比特位长度的 MAC 为最低安全标准[50]。AES-CCM 算法可以产生所需要的 MAC，并且可以根据需要得到 4 字节、6 字节和 8 字节等不同长度的 MAC（在本章采用 4 字节长度）。AES-CCM 算法在 MAC 生成的过程中将随机数引入，即每次加密产生的 MAC 均不相同，因此本章设计的安全 CAN 总线可以抵御当攻击者嗅探到有效数据帧对汽车进行的重放攻击。

本章的设计中没有考虑密钥管理部分，但是为使安全 CAN 总线协议的正常工作，需要保证不同的 ECU 节点在进行加密或解密时共享相同的密钥。

3.3 具体方案

本节将对基于 AES-CCM 算法的安全 CAN 总线协议设计进行详细的介绍，为了使描述简洁，表 3.1 列出了设计方案中涉及的定义。根据应用背景不同，本节共提出两种设计方案来实现安全 CAN 总线协议。

表 3.1 设计方案的相关定义

定义	描述
ECU_i	第 i 个 ECU 实体
ID_i	ECU_i 的 ID
Enc_{aes-m}	采用 AES 算法的 m 模式进行加密
N	随机数 Nonce
P	输入明文
A	附加消息
C	输出密文
P_{len}	明文长度
T_{len}	MAC 长度
MSBs(X)	取二进制串 X 的从左 s 个二进制位
LSBs(X)	取二进制串 X 的从右 s 个二进制位
||	连接符号

3.3.1 方案一

1、数据帧发送

如图 3.1 为方案一的安全 CAN 总线协议示意图，图中共有 3 个 ECU（ECU1、ECU2、ECU3）接入 CAN 总线。总线呈多主机通信，每个 ECU 既可以作为发送节点，可以作为接收节点。当 ECU 需要发送数据帧且总线空闲时，ECU 即可将数据帧发送。

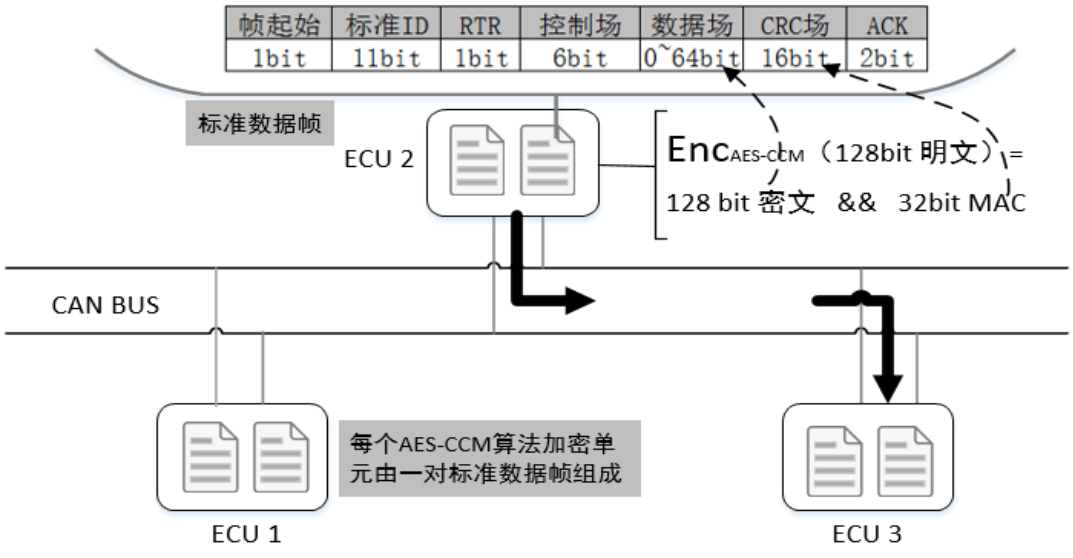

图 3.1 方案一的安全 CAN 总线协议示意图

数据帧发送之前，ECU 2 将至多 128 比特位长度的明文 P、附加消息 A 和随机数 N 组合进行格式化函数操作，然后利用 AES-CBC 算法生成 MAC（用 T 表示），如公式(3.1)所示。

$$T = Enc_{AES\text{-}CBC}(P \| A \| N) \tag{3.1}$$

T 的长度为 32 比特位，标准数据帧中 CRC 场的最大长度为 16 比特位，因此连续两个标准帧可以实现 T 替换 CRC 场。ECU 2 通过 AES-CTR 算法生成密文 C，如公式(3.2)所示。至多 128 比特位长度的密文 C 和 32 比特位的 T 经过简单处理组成消息 R，最终由连续的两个标准帧传送至目标节点 ECU 3。

$$C = Enc_{AES\text{-}CTR}(P) \tag{3.2}$$

通过 AES-CCM 加密算法对原始数据的处理，连续的两个标准帧构成了 1 个具有机密性和可认证性的安全 CAN 总线数据帧，完成方案一的安全 CAN 总线数据帧的发送。其中算法 3.1 描述了 AES-CCM 加密的过程。

算法 3.1 AES-CCM 加密

输入：N、A、P

输出：R

1. Formatting Function(N、A、P) -> $B_0, B_1, \ldots B_r$
2. $Y_0=CIPH_k(B_0)$
3. for i from 1 to r
4. $\quad Y_i=E_k(B_0 \oplus Y_{i-1})$
5. $T=MSB_{Tlen}(Y_r)$
6. Counter Generation Function(N) -> $Ctr_0, Ctr_1, ..Ctr_m$
7. for j from 0 to m
8. $\quad S_j=CIPH_k(Ctr_j)$
9. $S=S_1||S_2...||S_m$
10. Return $R=(P \oplus MSB_{Plen}(S)) \,||\, (T \oplus MSB_{Tlen}(S_0))$

2、数据帧接收

如图 3.1 所示，当目标节点 ECU 3 从 CAN 总线上接收到来自 ECU 2 连续的两个标准帧（单个安全数据帧）时，ECU 3 将数据场和 CRC 场中的内容组合成为 AES-CCM 解密算法中的一个输入参数 R，另外的两个输入参数为附加消息 A 和随机数 N（均预先存于 ECU 的 ROM 中）。通过 AES-CCM 解密算法，ECU 3 得到明文 P、MAC 和来自 ECU 2 的 T。将 MAC 与 T 进行一致性验证，如果通过验证那么返回明文 P；否则验证失败，丢弃明文并返回错误提示。其中算法 3.2 描述了 AES-CCM 算法解密的过程。

算法 3.2 AES-CCM 解密

输入：N、A、R

输出：P or INVALID

1. if $C_{len}<T_{len}$
2. Return INVALID
3. else
4. Counter Generation Function() -> $Ctr_0, Ctr_1, ..Ctr_m$
5. for j from 0 to m
6. $S_j=CIPH_k(Ctr_j)$
7. $S=S_1||S_2...S_m$
8. $P=MSB_{Clen-Tlen}(R)\oplus MSB_{Clen-Tlen}(S)$
9. $T=LSB_{Tlen}(R)\oplus MSB_{Clen-Tlen}(S_0)$
10. if N or A or P not valid
11. Return INVALID
12. else
13. Formatting Function(N、A、P) -> $B_0, B_1, ...B_r$
14. $Y_0=CIPH_k(B_0)$
15. For i from 1 to r
16. $Y_j=CIPH_k(B_i\oplus Y_{i-1})$
17. if $T\neq MSB_{Tlen}(Y_r)$
18. Return INVALID
19. else
20. Return P

3.3.2 方案二

CAN 总线协议包含两类数据帧，标准帧和扩展帧。其不同点主要为 ID 场的长度，标准帧的 ID 场长度为 11 比特位，扩展帧的 ID 场长度为 29 比特位。如图 3.2 为 CAN 标准帧与扩展帧对比图。对于标准帧数据，可用扩展帧进行发送，即取其 ID 场中 11 比特位作为标准帧 ID 场，其余 18 比特位作保留位。本节中，方案二利用扩展帧发送标准帧数据，将保留位中的 16 个比特位发送 MAC，其余两个比特位清零。同时，CRC 场用于搭载另外 16 比特位长度的 MAC。最终由 1 个扩展帧构成的安全 CAN 总线数据帧可以发送 32 比特位长度的 MAC 和至多 64 比特位长度的密文。

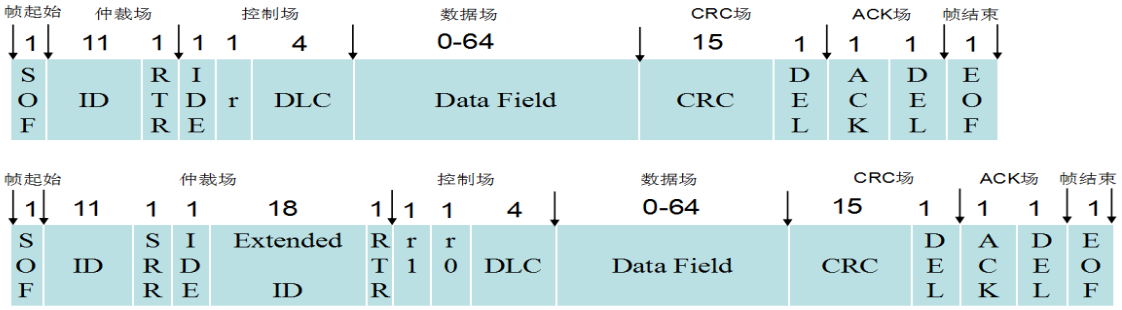

图 3.2 标准帧与扩展帧对比图

如图 3.3 为方案二的安全 CAN 总线协议示意图，在数据帧发送之前，发送节点 ECU 2 将至多 64 比特位长度的明文 P、随机数 N 和附加消息进 A 作为输入进行 AES-CCM 算法的加密，得到 32 比特位的 MAC 和至多 64 比特位的密文。利用 1 个扩展帧作为单个安全 CAN 总线数据帧进行消息发送，其中 32 比特位长度的 MAC 由扩展 ID 场和 CRC 场共同发送，8 字节长度的数据场用于发送密文。

对于安全 CAN 总线 数据帧的接收，当节点 ECU 3 在总线上监测到有扩展帧时，将 CRC 场和扩展 ID 场共同搭载的 32 位 MAC、随机数 N 和附加消息 A 作为 AES-CCM 解密算法的输入。解密过程中，ECU 3 对 MAC 进行认证，认证通过则返回正确明文，否则抛弃该数据帧并返回错误提示。

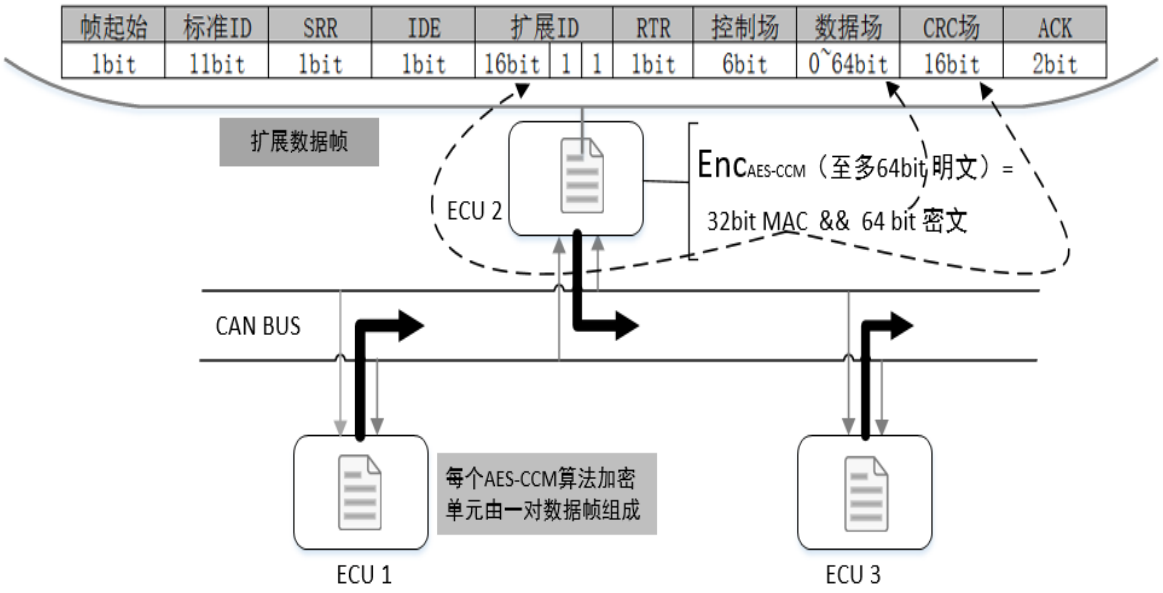

图 3.3 方案二的安全 CAN 总线协议示意图

3.3.3 对比分析

两种方案均在传统 CAN 总线协议的基础上，利用 AES-CCM 算法作为加密算法实现安全 CAN 总线协议。在功能上，两种方案均具备机密性、可认证性和抗重放攻击的能力。在安全等级上，两种方案均采用 128 比特位长度的密钥和 32 比特位长度的 MAC，可以高效的确保数据安全。两种方案均利用传统 CAN 总线协议的 CRC 场来发送 MAC，使安全 CAN 总线协议在保留传统总线协议所具备的传输错误检测功能的前提下，又增添了数据可认证功能。两种方案均为轻量级修改方案，未对传统 CAN 总线添加任何硬件模块且没有对协议的数据格式进行修改。因此，安全 CAN 总线具有良好的兼容性。

在方案一中，单个安全 CAN 总线数据帧由两个标准数据帧组成，消息的可认证性需要发送节点和接收节点进行同步操作，数据帧 MAC 认证需要接收节点

取得完整的安全 CAN 数据帧单元后才可以进行，即存在认证延迟。同时，如果在同步过程中有数据帧丢失，那么将造成认证失败。而在方案二中，通过同时对传统 CAN 总线协议的 CRC 场和扩展 ID 场的利用，密文和 MAC 的发送仅需要一个扩展帧即可完成，因此可以有效避免认证延迟和因帧丢失造成认证失败的问题，并且其通讯负载也将有所降低。然而，方案二的实现会导致扩展帧 ID 场长度由 29 比特位降低至 11 比特位，即仅支持标准 ID 场长度。

综上所述，两种方案有着相同的安全功能和安全等级，均可以有效的防御目前存在的多种汽车网络攻击手段。两者的区别主要体现在是否存在认证延迟，方案二对传统 CAN 总线协议进行高效的利用，可以通过 1 个扩展帧来实现安全 CAN 总线协议数据帧，消除了认证延迟。因此对于标准数据帧的发送，方案二更加具有优势。

3.4 实验分析

本节采用两块飞思卡尔 MC9S12XF512 开发板作为实验平台实现了本章提出的方案二，并从可行性和通信延迟两方面对安全 CAN 总线进行评测。

3.4.1 实验平台

本章采用两块 16 位飞思卡尔汽车开发板 MC9S12XF512 作为硬件开发平台，软件开发平台为飞思卡尔工具包 CodeWarrior Version 5.0。开发板通过 BDM 下载器与 PC 机连接下载二进制代码，同时通过 USBCAN II 来采集 CAN 总线上的数据。开发板通过 LCD 显示屏和串口调试软件来显示内部数据。图 3.4 为安全 CAN 总线协议的实验平台。

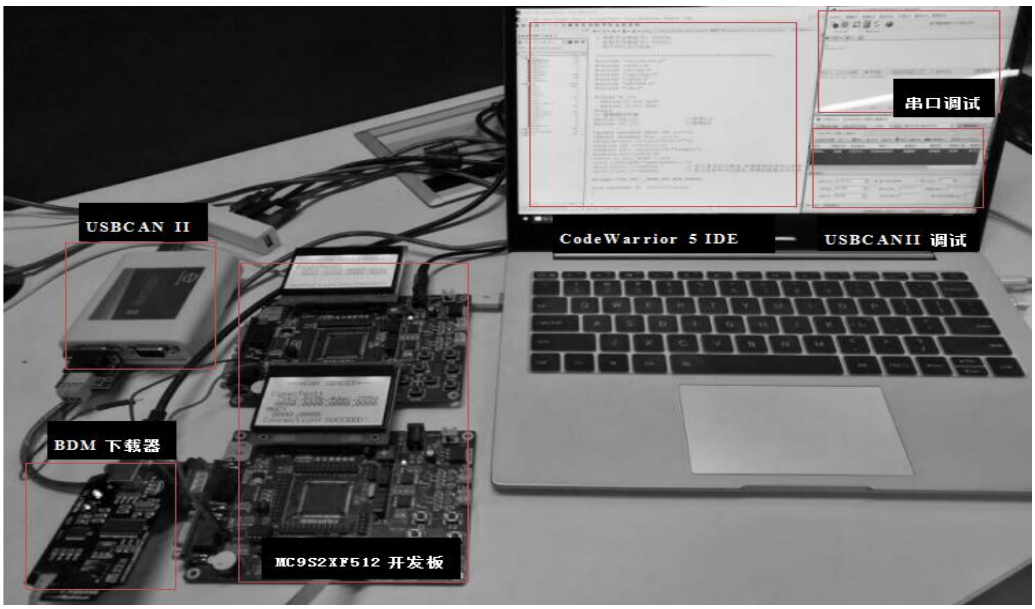

图 3.4 安全 CAN 总线协议的实验平台

如图 3.5 所示为安全 CAN 总线的软件架构，在传统 CAN 总线通信的接口层与应用层之间加入安全层，使其具有安全特性。

图 3.5 安全 CAN 总线协议的软件架构

3.4.2 性能分析

通常 ECU 的 CRC 接口在出厂时会被屏蔽，用户不可以对 CRC 场进行任何修改。为了完成对安全 CAN 总线协议的性能评测，本节实验对章节 3 的方案二进行适当调整。通常车载 ECU 发送各种功能的 CAN 消息为 8 字节，本节设定 CAN 消息为 6 字节，其余的 2 个字节用于仿真 CRC 场发送 MAC。

本节通过选取 1 组具体示例来验证方案的可行性。设定密钥 K 长度为 128 比特位，随机数 N 长度为 56 比特位，明文长度为 48 比特位，附加消息长度为 64 比特位，密钥 K、随机数 N、明文 P 和附加消息 A 的值如表 3.2 所示。

表 3.2 密钥 K、随机数 N、明文 P 和附加消息 A 的值

参数名称	参数值(16 进制)
K	40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
N	10 11 12 13 14 15 16
P	20 21 22 23 24
A	00 01 02 03 04 05 06 07

根据 AES-CCM 加密算法进行理论推导，得到 6 字节密文 CipherText 和 4 字节 MAC，密文 CipherText 和消息认证码 MAC 的值如表 3.3 所示。

表 3.3 密文 CipherText 和消息认证码 MAC 的值

参数名称	参数值(16 进制)
CipherText	71 62 01 5B C0 51
MAC	CE 10 4F 15

在本节实验中，发送节点 ECU 和接收节点 ECU 在启动后立即进入初始化模式，待初始化完毕后则进入工作模式，并通过 LCD 显示已成功接入 CAN 总线。发送节点 ECU 在应用层将明文 P 送入至协议的安全层与密钥 K、随机数 N 和附加消息 A 通过 AES-CCM 算法生成密文 CipherText 和消息认证码 MAC，通过 CAN 接口层发送安全 CAN 总线协议数据帧（帧 ID 为 10001000100， 占用扩展帧的第 4~14 比特位）至总线。全部 6 字节密文和 MAC 前 2 字节通过扩展帧数据场发送，MAC 后 2 字节通过扩展帧的第 15~30 比特位发送。

如图 3.6 为 USBCAN II 监测到的安全 CAN 总线协议数据帧。扩展帧 ID 的第 15~30 比特位的值为 0x4f15，8 字节数据后 2 字节为 0xce10，前 6 字节为 0x7162015bc051。监测到的数据帧数据同预期的密文 CipherText 和消息认证码 MAC 的值相同。

图 3.6 USBCAN II 检测到的安全 CAN 总线协议数据帧

接收节点 ECU 通过 CAN 接口层获取安全数据帧后，经过安全层将密文 CiphetText 和消息认证码 MAC 组合，并与随机数 N、附加消息 A 和密钥 K 进行 AES-CCM 解密验证。如果验证通过则返回明文至应用层，并通过 LCD 提示验证通过，否则丢弃明文并提示验证失败。如图 3.7 为安全 CAN 总线数据帧验证通过示意图。

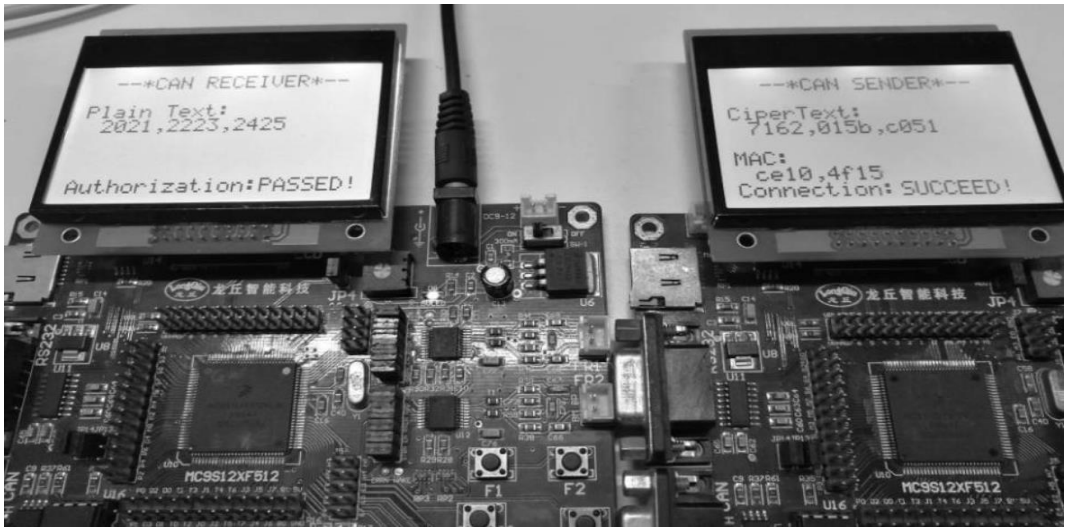

图 3.7 安全 CAN 总线数据帧验证通过示意图

如图 3.4 所示，本节利用 ECU 的定时器模块对安全 CAN 总线协议的性能进行测定，并通过与 PC 机相连的串口显示数据结果。其中 ECU 的配置参数如表 3.4 所示。

表 3.4 ECU 配置参数

ECU 模块	参数
总线时钟频率	16/32/40/128 MHZ
串口波特率	19200 bps
CAN 总线波特率	125K bps
定时器时钟源	总线时钟

为了得到准确的数据，本实验重复进行了 100 次。通过 16 位定时器溢出的次数来计算通讯延迟。分别对普通 CAN 总线、AES-CCM 加密、AES-CCM 解密和安全 CAN 总线在不同时钟频率下进行性能测定，各模块的性能对比结果如图 3.8 所示。当 ECU 的总线频率为 16 MHZ 时，安全 CAN 总线通讯延迟远大于普通 CAN 总线延迟。随着总线频率的增大，安全 CAN 总线的通讯延迟迅速降低，当总线频率为 128 MHZ 时，安全 CAN 总线的通讯可以在 2 毫秒内完成。因此随着车载 ECU 性能的逐步提升，本文设计的安全 CAN 总线协议可以高效的应用于智能汽车。

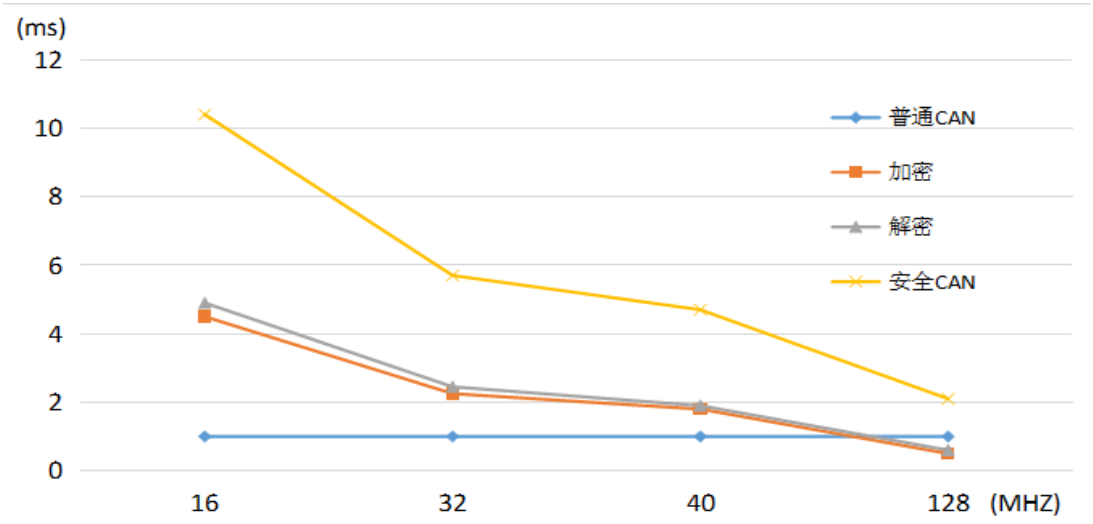

图 3.8 各模块的性能对比结果

3.4.3 工作对比

如表 3.5 所示为章节 3 中方案二与其他工作的对比结果。文献[19][25]所提所提出的安全 CAN 总线不能对数据机密性进行保护，因此攻击者可以对嗅探到的明文进行分析得到具体含义。虽然文献[20]与方案二均采用 AES-128 加密算法，但是 CCM 的加密模式使破解难度更大。其他三个文献中认证算法均采用哈

希算法，MD5 与其他两种加密算法相比有较快的处理速度，但是安全性能较弱，AES-CCM 算法则在安全性和处理速度上处于平衡点。综上所述，方案二有高安全性且性能上也有不俗的表现。

表 3.5 方案二与其他工作的对比结果

功能	文献[19]	文献[25]	文献[20]	方案二
加密算法	不支持	不支持	AES-128	AES-128
加密模式	不支持	不支持	ECB	CCM
认证算法	MD5	SHA-3	MD5	AES-CCM
抗重放攻击	不支持	支持	支持	支持

3.4.4 安全性分析

1、机密性

章节 3 提出基于 AES-CCM 算法的安全 CAN 总线协议，其中 AES-CTR 算法用于对数据帧的机密性进行保护。攻击者在总线上嗅探到的数据帧均为密文，在不知道密钥的情形下很难进行逆向破解。算法的密钥长度为 128 比特位，如果攻击者想通过穷尽攻击来获取密钥，那么需要进行 2^{128} 次尝试。然而，根据 CTR 工作模式的特点，每次加密都会生成新的随机数，再考虑到 ECU 的处理能力和 CAN 总线的传输速度，通过穷尽攻击来获取密钥是不可能实现的。

2、可认证性

本章采用与文献[17]相同的 32 比特位长度的 MAC。虽然 Yasuda 在文献[51]1]中提出如果攻击者在接触到 ECU 的内部固件的情况下，在有限时间内是可以完成对 32 比特位长度的 MAC 进行破解。但通常攻击者很难接触到 ECU 内部固件，因此这种特殊情况不在本文的考虑范围内。攻击者可以通过分析已获取到的 MAC 值结构来获取有效信息，但是在没有密钥的条件下攻击者仍然不能通过这些有效信息来获得正确的 MAC 值。生成 MAC 的密钥值在 ECU 中得到很好的安全存储很难被窃取，因此攻击者只能通过猜测 32 比特位的所有可能 MAC 值来进行攻击。在目前 CAN 总线传输能力下，如果平均每 10 毫秒进行一次密钥猜测尝试，那么攻击者需要 11930 小时来完成整个攻击过程[20]。同时，如果攻攻击者对目标 ECU 进行数据传输的间隔小于 10 毫秒，那么总线会产生 BUS-OFF 错误，这意味着此节点将与 CAN 总线断开连接不能再进行有效的数据通信。

3、抗重放攻击

重放攻击是指当攻击者嗅探到数据消息和认证消息后，将这些内容重复放入 CAN 总线发送至目标 ECU 而达到攻击的效果。在本章的安全 CAN 总线协议中，数据帧具有可认证性，而 MAC 值是判断该数据帧是否有效的直接因素。在

每个 MAC 的生成过程中都需要发送 ECU 节点和接收 ECU 节点共同管理的随机数作为输入参数，因此不同的安全 CAN 总线协议数据帧 MAC 值均不相同。当攻击者将嗅探到的内容发送到 CAN 总线上后，接收节点会检测到数据异常并进行抛弃处理。

3.5 小结

随着智能汽车在网络结构上发生的改变，传统 CAN 总线协议固有的设计缺陷使车内网络容易遭受来自外界的网络攻击。通过分析，现代密码学在车内网络中的应用可以有效的提升其防御能力。本章提出基于 AES-CCM 算法的安全 CAN 总线协议。该设计不仅考虑到了 ECU 较低的处理能力和 CAN 数据帧有限的数据场，而且在保证安全性的前提下对传统 CAN 总线只进行了轻量级的修改。根据不同的应用场景，本章提出两种方案。方案一主要利用 MAC 可以替代数据帧 CRC 场的性质来实现 CAN 总线的可认证性。而方案二则利用了扩展 ID 场来发送 MAC，将方案一中所需的帧数量缩减一半。两种设计方案均使传统 CAN 总线具有机密性和可认证性，很大程度上增强了对网络攻击的防御能力。本章在由 2 块飞思卡尔 MC9S12XF512 开发板构成的实验平台上进行了基于 AES-CCM 算法的安全 CAN 总线协议的设计。通过对传统 CAN 总线协议 CRC 场的利用，通过两组标准帧使 CAN 总线具有机密性、可认证性以及抗重放攻击的能力。为了进一步提高安全 CAN 总线协议的性能，通过利用扩展帧 ID 场的承载能力实现了单扩展帧构成安全 CAN 总线数据帧，并且实验结果与其他工作相比具有良好的性能表现。

第4章 一种针对 AES 解密的高效 DFA 攻击方法

安全总线协议的应用可以使智能汽车有效的抵御来自外界的多种网络攻击手段。在该应用中处于重要位置的密码系统通常是嵌入式系统，而嵌入式系统易受到 DFA 攻击的威胁。当入侵者可直接接触到汽车密码系统时，DFA 攻击可以有效的攻破此系统。针对 AES 算法的 DFA 攻击，目前绝大多数的工作都集中于对加密过程的影响，而很少有对 AES 解密过程的研究工作。AES 解密作为 AES 算法的重要组成部分，需要得到与加密过程同等的重视。因此本章提出一种针对 AES 解密的高效 DFA 攻击方法。其中基础方法证明了在 AES 解密过程可以进行 DFA 攻击，而优化方法将攻击效率大幅度提升，增强其破坏性。

4.1 故障模型

本章假定注入的故障为随机单字节非零故障，故障将被注入在 AES 解密过程的第三轮输入状态任意位置。为了描述清晰，本章定义 3 个 4×4 的字节数组，无故障明文 P、故障明文 P*和原始密钥 K 可以由公式组（4.1）表示。

$$
P=\left(\begin{array}{cc|cc} P_{00} & P_{04} & P_{08} & P_{12} \\ P_{01} & P_{05} & P_{09} & P_{13} \\ P_{02} & P_{06} & P_{10} & P_{14} \\ P_{03} & P_{07} & P_{11} & P_{15} \end{array}\right)
$$

$$
K=\left(\begin{array}{cc|cc} K_{00} & K_{04} & K_{08} & K_{12} \\ K_{01} & K_{05} & K_{09} & K_{13} \\ K_{02} & K_{06} & K_{10} & K_{14} \\ K_{03} & K_{07} & K_{11} & K_{15} \end{array}\right)
$$

$$
P^{*}=\left(\begin{array}{cc|cc} P_{00}^{*} & P_{04}^{*} & P_{08}^{*} & P_{12}^{*} \\ P_{01}^{*} & P_{05}^{*} & P_{09}^{*} & P_{13}^{*} \\ P_{02}^{*} & P_{06}^{*} & P_{10}^{*} & P_{14}^{*} \\ P_{03}^{*} & P_{07}^{*} & P_{11}^{*} & P_{15}^{*} \end{array}\right) \tag{4.1}
$$

4.2 基础攻击方法

如图 4.1 描述了在 AES 解密过程第三轮输入状态位置注入一个故障的密钥扩展过程。注入非零故障‘f’，经过逆字节替代，故障值被替换为另一个未知数，仍记‘f’。逆行位移不影响状态数组，在逆列混合的结尾，故障扩展到整个列，形成故障‘ef’‘9f’‘df’和‘bf’。轮密钥加操作对故障扩展也没有影响，所以第

二轮输入状态状态数组没有变化。在逆字节替代之后，故障变成‘F1’‘F2’‘F3’和‘F4’，经过逆行位移、逆列混合和轮密钥加，在第二轮的最后，中间态所有字节均受到故障值的影响。在第一轮逆字节替代后，故障值依次是 A0~A15，最终经过逆行位移与轮密钥加操作后得到故障明文 P*，由公式（4.2）表示。

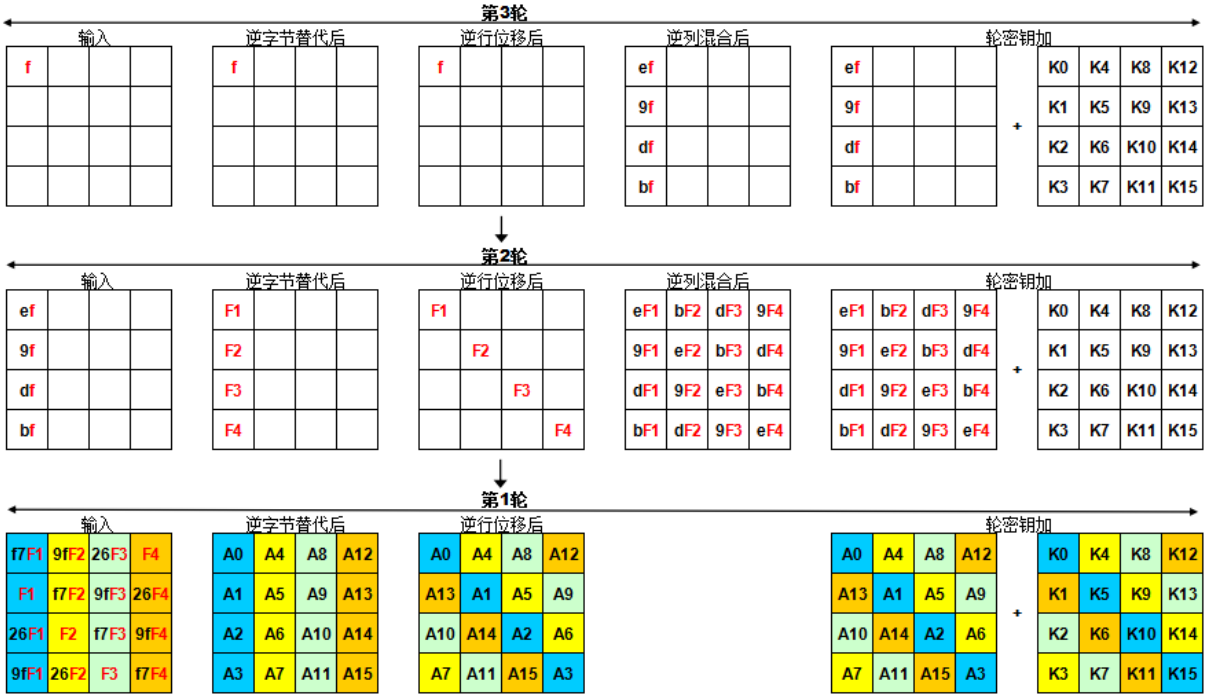

图 4.1 在 AES 解密过程第三轮输入状态位置注入一个故障的密钥扩展过程

$$P^{*}=\left(\begin{array}{cc|cc} P_{00}\oplus A_{00} & P_{04}\oplus A_{04} & P_{08}\oplus A_{08} & P_{12}\oplus A_{12} \\ P_{01}\oplus A_{13} & P_{05}\oplus A_{01} & P_{09}\oplus A_{05} & P_{13}\oplus A_{09} \\ P_{02}\oplus A_{10} & P_{06}\oplus A_{14} & P_{10}\oplus A_{02} & P_{14}\oplus A_{06} \\ P_{03}\oplus A_{07} & P_{07}\oplus A_{11} & P_{11}\oplus A_{15} & P_{15}\oplus A_{03} \end{array}\right) \tag{4.2}$$

根据图 4.1 中第二轮轮密钥加 P 和 P*的异或值，利用 AES 的性质可以得到公式组(4.3)。

$$\begin{aligned} eF_1 &= SB(P_{00}\oplus K_{00})\oplus SB(P_{00}\oplus K_{00}\oplus A_{00}) \\ 9F_1 &= SB(P_{05}\oplus K_{05})\oplus SB(P_{05}\oplus K_{05}\oplus A_{01}) \\ dF_1 &= SB(P_{10}\oplus K_{10})\oplus SB(P_{10}\oplus K_{10}\oplus A_{02}) \\ bF_1 &= SB(P_{15}\oplus K_{15})\oplus SB(P_{15}\oplus K_{15}\oplus A_{03}) \end{aligned} \tag{4.3}$$

根据公式组(4.3)各等式左侧的系数关系，可以得到公式组（4.4）。

$$\begin{aligned} e[SB(P_{05}\oplus K_{05})\oplus SB(P_{05}\oplus K_{05}\oplus A_{01})] &= 9[SB(P_{00}\oplus K_{00})\oplus SB(P_{00}\oplus K_{00}\oplus A_{00})] \\ d[SB(P_{05}\oplus K_{05})\oplus SB(P_{05}\oplus K_{05}\oplus A_{01})] &= 9[SB(P_{10}\oplus K_{10})\oplus SB(P_{10}\oplus K_{10}\oplus A_{02})] \\ b[SB(P_{05}\oplus K_{05})\oplus SB(P_{05}\oplus K_{05}\oplus A_{01})] &= 9[SB(P_{15}\oplus K_{15})\oplus SB(P_{15}\oplus K_{15}\oplus A_{03})] \end{aligned} \tag{4.4}$$

根据 P 和 P*的关系，公式组(4.4)可以被用于缩减密钥空间（K_{00}，K_{05}，K_{10}，K_{15}）。对 K_{00} 和 K_{05} 进行遍历，筛选掉不符合公式组中第 1 个公式的值，

得到缩减的 K_{05} 空间，进而利用第 2 个公式将 K_{05} 空间进一步缩减。经过最后 1 个公式的筛选后，得到 K_{05} 的最小空间。将 K_{05} 最小空间中的所有值代入公式组(4.4)中，得到 K_{00}、K_{10} 和 K_{15} 的最小空间。因此该组密钥的最小空间为 2^8。

利用公式组（4.5）取得密钥组(K_{02}，K_{07}，K_{08}，K_{13})的最小空间。

$$\begin{aligned}
&e[SB(P_{07}\oplus K_{07})\oplus SB(P_{07}\oplus K_{07}\oplus A_{11})]=9[SB(P_{02}\oplus K_{02})\oplus SB(P_{02}\oplus K_{02}\oplus A_{10})]\\
&e[SB(P_{07}\oplus K_{07})\oplus SB(P_{07}\oplus K_{07}\oplus A_{11})]=9[SB(P_{08}\oplus K_{08})\oplus SB(P_{08}\oplus K_{08}\oplus A_{08})]\\
&e[SB(P_{07}\oplus K_{07})\oplus SB(P_{07}\oplus K_{07}\oplus A_{11})]=9[SB(P_{13}\oplus K_{13})\oplus SB(P_{13}\oplus K_{13}\oplus A_{09})]
\end{aligned}\tag{4.5}$$

利用公式组（4.6）取得密钥组(K_{01}，K_{06}，K_{11}，K_{12})的最小空间。

$$\begin{aligned}
&e[SB(P_{12}\oplus K_{12})\oplus SB(P_{12}\oplus K_{12}\oplus A_{12})]=9[SB(P_{11}\oplus K_{11})\oplus SB(P_{11}\oplus K_{11}\oplus A_{15})]\\
&e[SB(P_{12}\oplus K_{12})\oplus SB(P_{12}\oplus K_{12}\oplus A_{12})]=9[SB(P_{01}\oplus K_{01})\oplus SB(P_{01}\oplus K_{01}\oplus A_{13})]\\
&e[SB(P_{12}\oplus K_{12})\oplus SB(P_{12}\oplus K_{12}\oplus A_{12})]=9[SB(P_{06}\oplus K_{06})\oplus SB(P_{06}\oplus K_{06}\oplus A_{14})]
\end{aligned}\tag{4.6}$$

利用公式组（4.7）取得密钥组(K_{03}，K_{04}，K_{09}，K_{14})的最小空间。

$$\begin{aligned}
&e[SB(P_{14}\oplus K_{14})\oplus SB(P_{14}\oplus K_{14}\oplus A_{06})]=9[SB(P_{09}\oplus K_{09})\oplus SB(P_{09}\oplus K_{09}\oplus A_{05})]\\
&e[SB(P_{14}\oplus K_{14})\oplus SB(P_{14}\oplus K_{14}\oplus A_{06})]=9[SB(P_{03}\oplus K_{03})\oplus SB(P_{03}\oplus K_{03}\oplus A_{07})]\\
&e[SB(P_{14}\oplus K_{14})\oplus SB(P_{14}\oplus K_{14}\oplus A_{06})]=9[SB(P_{04}\oplus K_{04})\oplus SB(P_{04}\oplus K_{04}\oplus A_{04})]
\end{aligned}\tag{4.7}$$

最终利用 1 对 P 和 P*可以将密钥空间缩减至为 2^{32}。如果有 2 对 P 和 P*（故障注入位置一致），能以 99%的几率获取唯一完整密钥[42]。

在上述攻击中，故障注入在第三轮输入状态的第 1 个字节。当故障注入在不同的字节位置，通常需要对应不同的故障分析方法。根据 AES 解密的性质，本章可以得到如下规律。将第三轮输入状态的状态数组分为四条对角线（D0、D1、D2 和 D3），如图 4.2 为四条对角线在 AES 解密第三轮的故障扩展图。在图中可以发现，当故障注入在 D0、D1、D2 和 D3 的任何一条对角线上，经过逆字节替代、逆行位移、逆列混合和轮密钥加后，对应在第三轮结束的位置具有相同的故障数组。这也就意味着当故障注入在这四条对角线中的同一条时可以采用相同的故障分析方法，因此对应 D0、D1、D2 和 D3 四条对角线共有四种故障分析方法。

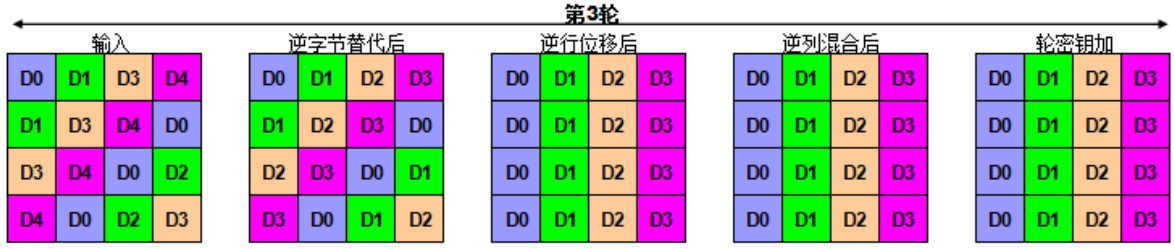

图 4.2 四条对角线在 AES 解密第三轮的故障扩展图

图 4.3~图 4.6 分别为故障注入在 D0~D3 对角线对应的第 2 轮及第 1 轮故障扩展图。通过观察这 4 张故障扩展图可以发现，在第 1 轮输入状态位置每张故障

扩展图中都有着不同的字节内部关系，并且在特定的某一条对角线情景中，这些内部关系不受故障值的影响。每张故障扩展图对应的字节内部关系都可以被用来进行故障分析，根据 AES 性质可以得到如公式组(4.4)~公式组(4.7)的筛选公式，利用 1 对正确明文和故障明文最终将完整密钥空间缩减至 2^{32}。

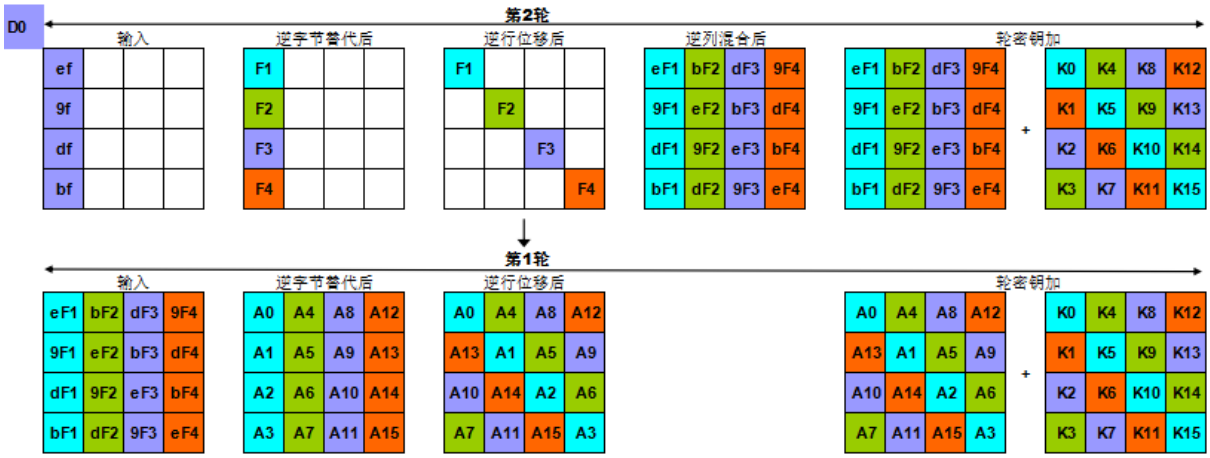

图 4.3 故障注入在 D0 对角线对应的第 2 轮及第 1 轮故障扩展图

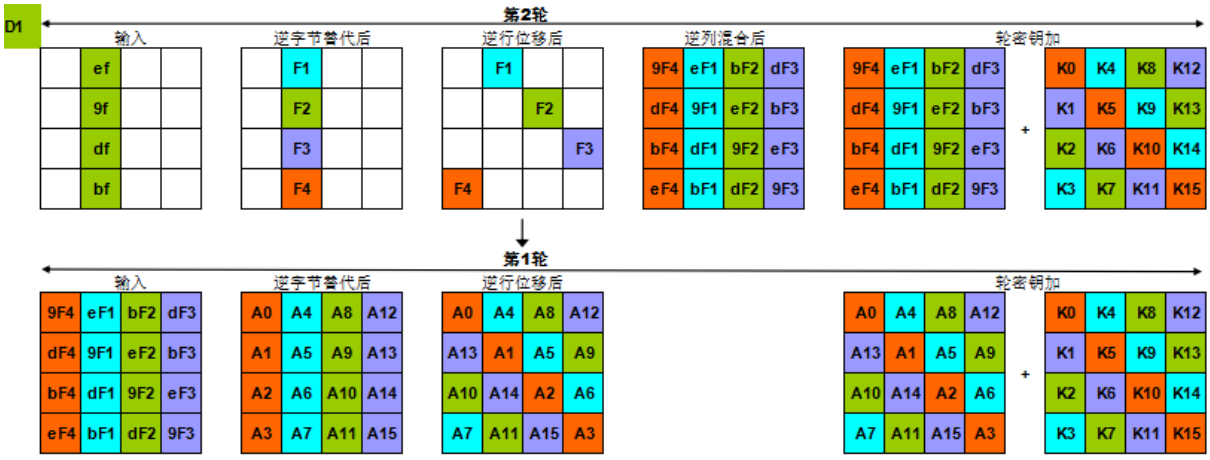

图 4.4 故障注入在 D1 对角线对应的第 2 轮及第 1 轮故障扩展图

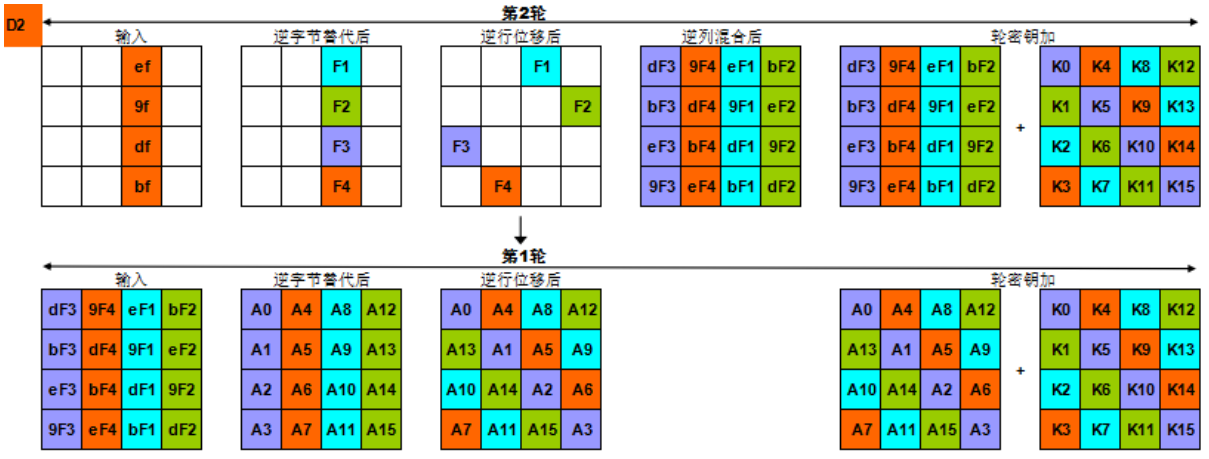

图 4.5 故障注入在 D2 对角线对应的第 2 轮及第 1 轮故障扩展图

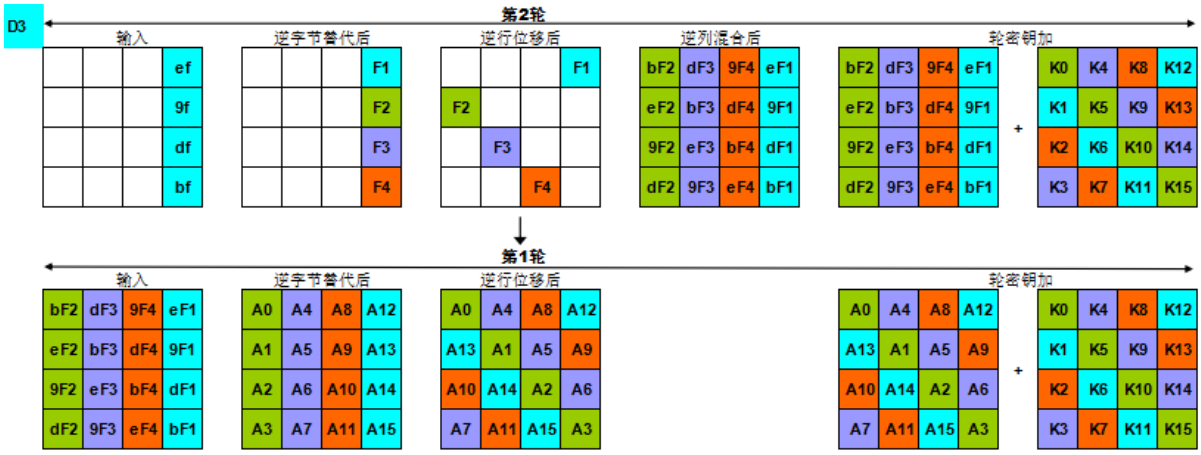

图 4.6 故障注入在 D3 对角线对应的第 2 轮及第 1 轮故障扩展图

4.3 优化攻击方法

DFA 攻击中的密钥恢复部分通常是由普通计算机通过软件来实现，与嵌入式设备相比普通计算机有着近无限大的存储空间。因此利用 AES 的性质将一部分 DFA 密钥恢复的计算内容以 S 盒分布表的形式存储在普通计算机中，可以大幅度提高破解密钥的速度。本节介绍了如何构建 S 盒分布表，然后分析介绍了如何利用 S 盒分布表提出本节的优化攻击方法。

4.3.1 构建 S 盒分布表

S 盒分布表包含两个输入（$InputA_i$，$InputB_i$）和一个输出集合 δ_i（含 δ_1 和 δ_2）构成。$InputA_i$ 为正确明文 P 和故障密文 P*第 i 个字节的异或值。$InputB_i$ 为图 4.1 中第二轮轮密钥加的状态数组第 i 个字节故障值。$InputA_i$ 和 $InputB_i$ 用公式组（4.8）表示。

$$\begin{aligned} InputA_i &= (P_i \oplus K_i) \oplus (P_i \oplus K_i \oplus A_i) \\ InputB_i &= SB(P_i \oplus K_i) \oplus SB(P_i \oplus K_i \oplus A_i) \end{aligned} \tag{4.8}$$

取 $\delta_1 = P_i \oplus K_i$，$\delta_2 = P_i \oplus A_i \oplus K_i$，根据公式组(4.8）对 $InputA_i$、$InputB_i$、δ_1 和 δ_2 进行遍历筛选出满足等式的 δ_1 和 δ_2。将所有的 δ_1 和 δ_2 以及对应的 $InputA_i$ 和 $InputB_i$ 存储于表格，最终得到完整的 S 盒分布表 $DT(InputA\ InputB)$，其中算法 4.1 为构建 S 盒分布表算法。

根据公式(4.9)，输入 $InputA_i$ 和 $InputB_i$ 可得到 δ，结果 δ 与明文 P 异或可返回一个正确的密钥值。

$$\delta = DT（InputA_i \ \ InputB_i） \tag{4.9}$$

算法 4.1 构建 S 盒分布表

输入：$InputA_i$、$InputB_i$、δ_1和δ_2

输出：DT（$InputA_i$，$InputB_i$）

```
1.    for InputA_i from 1 to 255
2.      for InputB_i from 1 to 255
3.        for δ1 from 0 to 255
4.          for δ2 from 0 to 255
5.            if   InputA_i = δ1 ⊕ δ2   and   InputB_i = SB(δ1) ⊕ SB(δ2)
6.                 then   保存 InputA_i 和 InputB_i 对应的 δ1 和 δ2
7.            end if
8.          end for
9.        end for
10.     end for
11.   end for
```

4.3.2 攻击过程

当获取 1 对正确明文 P 和故障明文 P*，根据公式（4.9）可得到 $InputA_i$。如果对 $InputB_i$的所有值进行猜测，可以返回约 128 个有效δ（根据第 5 章实验得出）。因为δ包含δ_1和δ_2两个值，则密钥 K_i 的可能值有 $2\times128=2^8$ 个。根据图 4.1 中第二轮轮密钥加的第 1 列关系可以得到公式组（4.10）。

$$\begin{aligned} eInputB_{05} &= 9InputB_{00} \\ eInputB_{05} &= 9InputB_{10} \\ bInputB_{05} &= 9InputB_{15} \end{aligned} \tag{4.10}$$

根据公式组（4.10）可以得到公式组（4.11）。

$$\begin{aligned} InputB_{00} &= f7InputB_{05} \\ InputB_{10} &= 26InputB_{05} \\ InputB_{15} &= 9fInputB_{05} \end{aligned} \tag{4.11}$$

根据公式组（4.11）以及公式（4.9）推导出公式组（4.12）。

$$\begin{aligned} \delta_{00} &= DT(InputA_{00}\ \ f7InputB_{05}) \\ \delta_{05} &= DT(InputA_{05}\ \ 01InputB_{05}) \\ \delta_{10} &= DT(InputA_{10}\ \ 26InputB_{05}) \\ \delta_{15} &= DT(InputA_{00}\ \ 9fInputB_{05}) \end{aligned} \tag{4.12}$$

公式组（4.12）中 $InputA_i$ 可根据公式组（4.8）的第 1 个公式获得，对 $InputB_{05}$ 进行遍历可得到 δ_{00}，δ_{05}，δ_{10}，δ_{15}。因为本章中注入的故障为非 0 值，故需祛除值为 0 的 δ 值。最终密钥 K_{05} 的空间约为 32，本组（K_{00}，K_{05}，K_{10}，K_{15}）的密钥空间为 $32\times32=2^8$。其完整密钥 K 的搜索空间为 $(2^8)^4=2^{32}$。

4.4 实验及分析

由于实验器材的限制，后者无法实现章节 4 中的故障注入部分，因此本章在 VS 2010 开发平台上通过仿真的方式完成 DFA 攻击实验，并将实验结果与前人工作进行了详细的比较。

4.4.1 实验平台

DFA 攻击包括故障注入和故障分析两个部分，而故障注入通常需要昂贵、复杂的专业工具才可以完成，因此由于实验器材的限制本章只进行故障分析部分的实验。同时，由于篇幅所限本章只进行第 4 章优化 DFA 攻击方法的实验。其中实验平台为普通 PC(dual Intel(R) Pentium(R) E6700 core (3.20 GHz))，编译器版本为 VC++ 10.0。

4.4.2 选取实验参数

本章实验中，需要 2 对正确明文和故障明文而不需要密钥及密文。但是为了验证攻击的有效性，本章随机选取 1 组密钥 K 和 1 组密文 CT，如表 4.1 所示。

表 4.1 密钥 K 和密文 CT 的值

参数名称	参数值(16 进制)
K	2B 7E 15 16 28 AE D2 A6 AB F7 15 88 09 CF 4F 3C
CT	39 25 84 1D 02 DC 09 FB DC 11 85 97 19 6A 0B 32

经过 AES 的解密算法，得到正确明文 PT，用表 4.2 表示明文 PT 的结果。

表 4.2 正确明文 PT 的值

参数名称	参数值(16 进制)
PT	32 43 F6 A8 88 5A 30 8D 31 31 98 A2 E0 37 07 34

根据故障模型，本章假设在 AES 解密的第三轮输入状态第 0 个字节注入随机非零故障，经过连续相同的两次操作分别得到故障明文 PT_1^* 和 PT_2^*，如表 4.3 所示。

表 4.3 故障明文 PT_1*和 PT_2*的值

参数名称	参数值(16 进制)
PT_1^*	D0 C3 5B 15 24 E1 A7 C6 0A 05 E0 6F 73 55 1A F0
PT_2^*	6E 80 1D 53 21 F0 21 C3 A9 1C D2 9F F9 6B E5 D7

4.4.3 密钥恢复结果

根据上节的计算过程，对 16 字节密钥的恢复需要分成四组，由于篇幅所限，本节仅列出其中一组（K_{00}、K_{05}、K_{10}、K_{15}）的实验结果。

将正确明文 PT 和故障明文 PT_1^*代入 4.4.2 章节的攻击过程，得到 PT_1^*对应（K_{00}、K_{05}、K_{10}、K_{15}）组所有候选值的实验结果，如表 4.4 所示。

表 4.4 PT_1*对应（K_{00}、K_{05}、K_{10}、K_{15}）组的所有候选值

序号	K_{00}	K_{05}	K_{10}	K_{15}
01	E3 01	A7 1C	66 1E	A1 65
02	D7 35	C6 7D	5E 26	FC 38
03	A6 44	E2 59	EB 93	F2 36
04	B7 55	CF 74	E7 9F	D2 16
05	C9 2B	AE 15	6D 15	F8 3C
06	F8 1A	C2 79	64 1C	CC 08
07	E0 02	CB 70	4F 37	AE 6A
08	FD 1F	B9 02	DB A3	C8 0C
09	A5 47	99 22	E6 9E	A9 6D
10	E4 06	EC 57	C0 B8	C0 04
11	90 72	F9 42	F6 8E	E9 2D
12	83 61	B1 0A	6F 17	E5 21
13	DC 3E	F4 4F	FD 85	C7 03
14	87 65	AF 14	72 0A	BD 79
15	AF 4D	8C 37	E9 91	A6 62
16	8B 69	A4 1F	C2 BA	8D 49
17	9D 7F	90 2B	E8 90	C6 02
18	CF 2D	BD 06	EC 94	A7 63
19	C2 20	C8 73	D8 A0	87 43
20	85 67	87 3C	C5 BD	E2 26

同理，将正确明文 PT 和故障明文 PT_2^*代入上节的攻击过程，得到 PT_2^*对应（K_{00}、K_{05}、K_{10}、K_{15}）组所有候选值的实验结果，如表 4.5 所示。

表 4.5 PT_2*对应（K_{00}、K_{05}、K_{10}、K_{15}）组的所有候选值

序号	K_{00}	K_{05}	K_{10}	K_{15}
01	EC B0	88 22	C4 8E	C6 25
02	67 3B	9F 35	CF 85	FB 18
03	0C 59	A7 0D	77 3D	81 62
04	49 15	F4 5E	F3 B9	91 72
05	E2 BE	F1 5B	F6 BC	85 66
06	EF B3	93 39	68 22	B2 51
07	F3 AF	DC 76	E7 AD	8F 6C
08	D5 89	A2 08	E5 AF	FA 19
09	D7 8B	EA 40	65 2F	FC 1F
10	CC 90	D5 7F	F5 BF	A1 42
11	4C 10	F7 5D	E4 AE	DA 39
12	78 24	E4 4E	7C 36	ED 0E
13	D9 85	98 32	D7 9D	88 6B
14	5E 02	A0 0A	D4 9E	E7 04
15	69 35	DF 75	62 28	B5 56
16	E5 B9	BC 16	4D 07	98 7B
17	6C 30	F2 58	7B 31	97 74
18	40 1C	C7 6D	FF B5	9D 7E
19	E6 BA	B8 12	C6 8C	D9 30
20	4A 16	F0 5A	DA 90	FF 1C
21	41 1D	99 33	76 3C	F9 1A
22	5C 00	FE 54	D0 9A	A8 4B
23	77 2B	AE 04	5F 15	DF 3C
24	6B 37	91 3B	46 0C	A3 40

结合 PT_1^*对应的第 05 项与 PT_2^*对应的第 22 项可以确认（K_{00}、K_{05}、K_{10}、K_{15}）组的正确密钥，如表 4.6 所示。

表 4.6 PT_1*，PT_2*（K_{00}、K_{05}、K_{10}、K_{15}）组的正确密钥

序号	K_{00}	K_{05}	K_{10}	K_{15}
05	C9 2B	AE 15	6D 15	F8 3C
23	77 2B	AE 04	5F 15	DF 3C
-	2B	AE	15	3C

同理获得另外三组的密钥，最终得出完整的 16 字节原始密钥的值 K_0。攻击结果和攻击耗时如图 4.7 所示。

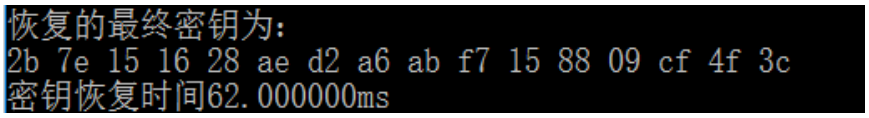

图 4.7 攻击结果和攻击耗时

将 K_0 与 K 进行对比，密钥值一致，即证明了攻击的有效性。本章实验共随机采取 1000 条测试样例，密钥恢复成功率高于 99%，所需要的平均时间为 70 毫秒，其攻击方法具有准确、高效和所需故障密文少的优势。

4.4.4 工作结果对比

表 4.7 显示本章 DFA 攻击实验和其他工作结果的对比。由于没有针对 AES 解密过程的 DFA 攻击工作，因此本节将实验结果与针对加密过程的 DFA 攻击工作进行对比。表中的故障明文（密文）个数均为密钥空间为 1 条件下的数目。与文献[32]和文献[38-41]的工作（所需密文个数从 40 到 128 不等）进行对比，本节实验结果仅需 2 对正确密文和故障密文。与文献[41]的工作进行对比，在同等故障明文/密文数量下将破解时间从 342 毫秒缩减至 70 毫秒。章节 4.3 提出的基础攻击方法虽然可以成功施展，但是计算时间却达到了 853 毫秒，因此基于 S 盒分布表的优化攻击方法更具有优势。

表 4.7 与其他工作结果对比

参考文献	故障模型	故障注入位置	故障明文/密文(个)	时间(毫秒)	目标
[38]	单比特位	固定比特位	128	-	加密
[32]	单比特位	固定字节的任意比特位	大约 50	-	加密
[39]	单字节	第 9 轮行位移之后	大约 40	-	加密
[40]	单字节	第 8 轮与第 9 轮之间	2	-	加密
[41]	单字节	第 8 轮开始的位置	2	342	加密
基础攻击	单字节	第 3 轮开始的位置	2	853	解密
优化攻击	单字节	第 3 轮开始的位置	2	70	解密

4.5 小结

本章针对 AES 的解密过程提出一种高效的 DFA 攻击方法。本章在第三轮输入状态注入单字节非 0 故障，根据 AES 性质得到故障扩展图。利用故障关系列出 4 组等式，将密钥 K 的空间缩减至 2^{32}。同时，利用普通计算机的强大存储能力保存 S 盒分布表，提出一种基于 S 盒分布表的高效攻击方法。两种方法的完

整密钥搜索空间均为 2^{32}，但基于 S 盒分布表的 DFA 攻击方案拥有更快的攻击速度。当存在 2 对正确明文和故障明文，两种方法均可以 99%的几率获取唯一完整密钥。同时，本章在 VS 2010 开发平台开展了一种针对 AES 解密的 DFA 攻击方法实验。针对 AES 解密过程的特性，首先提出基础攻击方法验证 DFA 攻击对 AES 解密过程的有效性。进而通过构建 S 盒分布表提出优化的攻击方法，攻击效率得到显著提升，实验结果显示攻击者在 2 对正确密文和故障密文的条件下可以于 70 毫秒内完成 AES 算法密钥的恢复。

结论和展望

随着经济发展和科技进步，互联网规模快速增长，人们的生活进入全面信息化时代。互联网发展的大潮给汽车行业带来了前所未有的冲击。近些年车联网技术、智能交通系统、高级辅助驾驶系统、无人驾驶汽车和云服务等信息化应用如雨后春笋般地涌现出来，使汽车从简单的交通工具逐渐演变成一个庞大而复杂的移动信息平台，极大丰富了汽车的功能与应用。

汽车智能化和网联化已经是汽车行业未来发展的必然方向，网联汽车使汽车不再是信息孤岛，作为一个网络上的移动信息平台，在享受互联网带来便利的同时，也必须面对来自互联网的潜在安全威胁。随着新兴汽车生态系统功能的发展和完善，汽车电子系统越来越依赖于信息共享和车载网络通信，近年来针对汽车的信息安全事件逐渐增加，智能汽车信息安全问题是汽车行业信息化发展必须面对的严峻考验，车载网络系统需要有效的安全解决方案。

现代密码学中的非对称加密算法对汽车内部网络的保护具有得天独厚的优势，不但可以保证数据机密性并且具有可认证性，但其加密速度慢的特性不适合应用于对实时性要求较高的车载电子系统。相反，分组加密算法在加密速度方面非常具有优势，但是不能保证车载电子系统数据的可认证性。由 NIST(美国国家标准与技术研究院)提出的 AES-CCM 算法，结合了对称密码加密速度快和非对称加密算法的数据可认证功能，非常适合应用于车内网络数据的保护工作。

密码的应用通常需要密码系统来实现，对于汽车来说密码系统的本质是嵌入式系统，而以 DFA 攻击为例的侧信道攻击对嵌入式密码系统具有很强的破坏性。因此对 DFA 攻击与防御进行深入研究有利于提升密码系统对侧信道攻击的抗攻击能力。

本文围绕智能汽车的网络安全问题展开了深入的研究，主要研究工作和成果有如下几个方面：

首先，针对普通 CAN 总线的设计缺陷，提出基于 AES-CCM 算法的安全 CAN 总线协议。通过对 CAN 标准帧 CRC 场的利用，使 CAN 总线具有机密性、可认证性和抗重放攻击的能力。通过对扩展帧 ID 场的合理利用提出了另一种方案，即算法产生的密文由数据场发送，而 4 字节的 MAC 则分别由扩展帧的扩展 ID 和 CRC 场共同完成发送，该方案具有设计简单、容错率高和通信延迟小的特点。

其次，提出了一种针对 AES 解密的 DFA 攻击方法。该方法将故障注入在 AES 解密过程的第 3 轮，故障类型为随机字节错误。根据 AES 在解密的性质，

该方法验证了 DFA 攻击可以在 AES 的解密过程展开，并指出利用 1 对正确明文和故障明文可以将密钥空间缩减至 2^{32}，2 对正确明文和故障明文的条件下攻击者能以 99%的概率确定唯一密钥。同时，本文通过构建 S 盒分布表提出一种优化的攻击方法，在保证密钥搜索空间不变的前提下使攻击效率大幅度提升。

最后，本文利用两块飞思卡尔 MC9S12XF512 开发板作为实验平台实现了所设计的安全 CAN 总线协议，并对其可行性和通信延迟进行了分析与评测。实验结果显示本文提出的安全 CAN 总线具有机密性、可认证性和抗重放攻击能力，且所带来的通信延迟可以被当前车载电子系统所接受。同时，本文在 VS2010 开发平台上对所提出的优化 DFA 攻击方法进行了仿真实验，通过具体示例验证了攻击方法的有效性并且指出在有 2 对正确明文和故障明文的情况下可以在 70 毫秒内完成对针对 AES 解密过程的攻击，与前人的工作相比本方法的攻击所需时间减少了 79.5%。

由于作者自身能力和实验条件的限制，本文仍然存在很多需要解决和完善的问题，进一步的研究工作应该重点关注如下几个方面：

1、对于车内网络安全的研究，本文只针对 CAN 总线进行了安全性设计。然而，车内网络除了 CAN 总线还存在 LIN、Flexray 和 MOST 等其他总线。所有总线都是车内网络的一部分，如果任何一种总线缺失安全性，那么攻击者仍可以对车内网络进行攻击。因此下一步工作可以围绕其他类型总线的安全性进行展开。

2、针对 AES 解密的 DFA 攻击方法中，虽然提出的优化方法可以将攻击效率大幅度提升，但是在前人研究工作中本方法的密钥空间不是最优。下一步工作可以将 S 盒分布表应用在最优密钥空间的方法上，使攻击效率进一步提升。

参考文献

[1] Broy M, Kruger I H, Pretschner A, et al. Engineering automotive software. Proceedings of the IEEE, 2007, 95(2): 356-373.

[2] 谢勇. 新一代汽车电子系统的网络体系结构若干关键技术研究: [湖南大学博士学位论文]. 长沙: 湖南大学, 2013, 3-8.

[3] 吴忠泽. 智能汽车发展的现状与挑战. 时代汽车, 2015(7):42-45.

[4] 李骏, 邱少波, 李红建,等. 智慧城市的智能汽车. 中国科学:信息科学, 2016(5).

[5] Koscher K, Czeskis A, Roesner F, et al. Experimental security analysis of a modern automobile. In: IEEE Symposium on Security and Privacy. IEEE, 2010, 447-462.

[6] Miller C, Valasek C. Remote exploitation of an unaltered passenger vehicle. Black Hat USA, 2015, 2-10.

[7] Kleberger P, Tomas O, Erland J. Security aspects of the in-vehicle network in the connected car. In: Intelligent Vehicles Symposium. IEEE, 2011, 530-532.

[8] FlexRay Protocol Specification. Flexray Consortium Std. V. 3.0.1, 2010.

[9] LIN Specification Package. LIN Consortium Std. Revision 2.2 A, 2010.

[10] MOST Specification. MOST Cooperation Std. Revision 3.0 E2, 2010.

[11] Rijmen V, Daemen J. Advanced encryption standard. Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology, 2001:19-22.

[12] Dworkin M. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2004, 1-3

[13] 邵梁强. 车身 CAN 总线系统及上位机检测软件的开发: [南京理工大学硕士学位论文]. 南京: 南京理工大学, 2007, 20-23.

[14] 吴任飞. 新一代车用网络的网关设计及其性能分析:[湖南大学硕士学位论文]. 长沙：湖南大学, 2014.

[15] Wolf M, Weimerskirch A, Wollinger T. State of the art: Embedding security in vehicles. EURASIP Journal on Embedded Systems, 2007, 2007(1): 074706.

[16] Brooks R R, Sander S, Deng J, et al. Automobile security concerns. IEEE Vehicular Technology Magazine, 2009, 4(2):20-25.

[17] Schweppe H, Roudier Y, Weyl B, et al. Car2x communication: securing the last meter-a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography. In: IEEE Vehicular Technology Conference, IEEE, 2011, 1-5.

[18] Schweppe H, Weyl B, Roudier Y, et al. Securing car2X applications with effective hardware software codesign for vehicular on-board networks. VDI Automotive Security, 2011, 27.

[19] Groza B, Murvay S. Efficient protocols for secure broadcast in controller area networks. IEEE Transactions on Industrial Informatics, 2013, 9(4): 2034-2042.

[20] Woo S, Jo H J, Lee D H. A practical wireless attack on the connected car and security protocol for in-vehicle CAN. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(2): 993-1006.

[21] 王亚猛. 基于 OPENSSL-车载网关认证系统的设计与实现:[吉林大学硕士学位论文]. 长春: 吉林大学, 2010.

[22] Lin C W, Sangiovanni-Vincentelli A. Cyber-security for the Controller Area Network (CAN) communication protocol. In: IEEE International Conference on Cyber Security. IEEE, 2012, 1-7.

[23] Lin C W, Zhu Q, Phung C, et al. Security-aware mapping for CAN-based real-time distributed automotive systems. In: IEEE Proceedings of the International Conference on Computer-Aided Design, IEEE, 2013, 115-121.

[24] 王喜文. 汽车信息安全问题不容忽视. 汽车工业研究, 2013 (11): 34-39.

[25] Wang Q, Sawhney S. VeCure: A practical security framework to protect the CAN bus of vehicles, In: 2014 International Conference on the Internet of Things. IEEE, 2014, 13-18.

[26] 温凤桐. 分组密码工作模式的研究: [北京邮电大学博士学位论文]. 北京: 北京邮电大学, 2006, 28-29.

[27] FIPS PUB. 81, DES modes of operation. Issued December. 1980, 2: 63.

[28] Kocher P C. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other system. In: Annual International Cryptology Conference. Springer, 1996, 104-113.

[29] 李升. 若干轻量级分组密码故障攻击研究: [上海交通大学硕士学位论文]. 上海: 上海交通大学, 2014, 10-20

[30] Biham E, Shamir A. Differential fault analysis of secret key cryptosystems. In: Annual International Cryptology Conference. Springer, 1997, 513-525.

[31] Giraud C. DFA on aes. In: International Conference on Advanced Encryption Standard. Springer, 2005, 27-41.

[32] Boneh D, DeMillo R A, Lipton R J. On the importance of checking cryptographic protocols for faults. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1997, 37-51.

[33] Wang Y, Ha Y J. FPGA-based 40.9-Gbits/s masked AES with area optimization for storage area network. IEEE Transactions on Circuit and System II, IEEE, 2013, 60(1):36-40.

[34] Wang Y, Ha Y. A performance and area efficient ASIP for higher-order DPA-resistant AES. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2014, 4(2):190-202

[35] Yuan Z, Wang Y, et al. FPGA based Optimization for Masked AES Implementation. In: IEEE International Midwest Symposium on Circuit and Systems, IEEE, 2011, 1-4.

[36] Yen S M, Chen J Z. A DFA on Rijndael. In: Information Security Conference 2002. 2002,35-37.

[37] Chen C N, Yen S M. Differential fault analysis on AES key schedule and some countermeasures. In: Australasian Conference on Information Security and Privacy. Springer, 2003, 118-129.

[38] Blomer J, Seifert J. Fault based cryptanalysis of the advanced encryption standard. In: International Conference on Financial Cryptography. Springer, 2003, 162-181.

[39] Dusart P, Letourneux G, Vivolo O. Differential fault analysis on AES. In: Applied Cryptography and Network Security. Springer, 2003, 293-306.

[40] Piret G, Quisquater J. A differential fault attack technique against SPN structures, with application to the AES and KHAZAD. In: Cryptographic Hardware and Embedded Systems. Springer, 2003, 77-78.

[41] Saha D, Mukhopadhyay D, Chowdhury D R. A Diagonal Fault Attack on the Advanced Encryption Standard. IACR Cryptology ePrint Archive, 2009(581): 8-10.

[42] Debdeep M. An improved fault based attack of the advanced encryption standard. In: International Conference on Cryptology in Africa. Springer, 2009, 421-434

[43] 叶世芬. 安全芯片物理防护研究: [浙江大学硕士学位论文]. 杭州:浙江大学, 2005, 37-40.

[44] 杜育松, 王大星, 沈 静. 一种对 AES-128 的差分错误分析原理. 计算机工程, 2006, 32(23): 174-176.

[45] 刘祥忠. 分组密码 AES-128 的差分故障攻击. 计算机技术与发展, 2012, 22(9): 221-224.

[46] 孙维东, 俞军, 沈磊. 对称加密算法 AES 和 DES 的差分错误分析. 复旦学报(自然科学版), 2013, 03(2013): 297-302.

[47] Somani U, Lakhani K, Mundra M. Implementing digital signature with RSA encryption algorithm to enhance the Data Security of cloud in Cloud Computing. In: 1st International Conference on Parallel Distributed and Grid Computing. IEEE, 2010, 211-216.

[48] Luk M, Mezzour G, Perrig A, et al. MiniSec: a secure sensor network communication architecture. In: Proceedings of the 6th international conference on Information processing in sensor networks. ACM, 2007, 479-488.

[49] Karlof C, Sastry N, Wagner D. TinySec: a link layer security architecture for wireless sensor networks. In: Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 2004, 162-175.

[50] Handschuh H, Preneel B. Minding your MAC algorithms. Information Security Bulletin, 2004, 9(6): 213-221.

[51] Yasuda K. Multilane. HMAC-Security beyond the birthday limit. In: International Conference on Cryptology in India. Springer, 2007, 18-32.

致　谢

时光荏苒，岁月如梭，转眼又到了毕业的季节。这是我在千年学府、百年名校——湖南大学的第三个年头了，在此对曾经帮助和鼓励我的老师、父母、同窗和朋友们表示由衷的感谢，有了你们的陪伴才使我的学术生涯不再孤单且充满意义，同时也有太多的不舍与眷恋。

首先我要感谢我的导师李仁发教授。三年来李老师用渊博的知识和严谨务实的学者风范深深的影响着我，使我不断的激励自己向前奋进。从论文选题到学位论文的撰写的每个环节中，李老师都提供了非常专业的指导建议，且在开拓研究视野和学术研究设备支持上都提供了可靠的保障。能够成为李老师的学生，我深感荣幸。

特别感谢王奕老师，王老师是我在密码学领域的领路人。在科研道路上她为我指明了方向，让我在正确的道路上去学习、研究和实现。在生活上，王老师宽容待人，像挚友一样对我进行无私的开导与帮助。王老师一丝不苟的专研精神和平易近人的为人态度将是我终身学习的榜样。

衷心感谢杨科华老师和付彬老师。谢谢杨科华老师在开题和中期检查对我的指导，让我少走了很多弯路。很幸运能够遇到付彬老师，让我在生活和学业得到很大的帮助。

感谢课题组的博士师兄和师姐，吴武飞、黄晶和白洋。吴博士在汽车电子领域有着深厚的积淀，在与他的探讨中得到了很多的启迪。黄博士在我的论文写作过程中不断进行鼓舞，使我灰心丧气的时候可以勇敢面对困难。白博士无论在生活上还是学术研究上都给我很多关照，让我更加感受到实验室的温暖。衷心祝愿各位师兄、师姐可以顺利毕业，找到理想的工作。

感谢李坤明、马啸啸、卢兴沄、朱婷婷、杨竞、秦凯强和刘四平等各位硕士同学，正是有你们的陪伴和帮助，我才能从孤独与自卑中重建阳光积极迎接新生活。

特别感谢我的父母、哥哥和女朋友，谢谢你们对我无私的爱，在背后永远支持我使我不断向前，没有你们的扶持与奉献，我无法取得今天的成绩。我会带着你们的期许继续拼搏努力，展现更好的自己。

最后，谨向审阅本论文及答辩组的老师们致以诚挚的谢意。由于本人水平有限，文中难免有不足之处，敬请各位老师批评、指正。

朱立民

2017 年 1 月 1 日于长沙

附录 A 攻读硕士学位期间发表的学术论文

[1] Limin Zhu, Yi Wang, Renfa Li. Efficient differential fault analysis attacks to AES decryption for low cost sensors in IoTs. In: Circuits and Systems (ISCAS), 2016 IEEE International Symposium on. IEEE, 2016: 554-557. (EI Index)

附录 B 攻读硕士学位期间所参与的项目

[1] 国家自然科学基金[61672217]：新一代汽车嵌入式系统功能安全的建模与算法研究.