

Design and Implementation of a Lightweight AES Encrypted Communication System for Embedded Devices *

LIU Zheng¹, DAI Peipei¹, XING Jianping^{1*}, SUN Qiyu², LIU Yufeng²
(1.School of Microelectronics, Shandong University, Ji'nan Shandong 250101, China;
2.Shuifa Wisdom industry Group Limited, Ji'nan Shandong 250101, China)

Abstract: A lightweight implementation scheme based on software is proposed for embedded devices according to 128-bit AES encryption algorithm. By using T-type lookup table storage scheme, byte substitution, shiftrows and mixcolumns operations in AES encryption algorithm are transformed into lookup table operation, and the memory access mechanism is optimized to enable AES encryption algorithm to run in a limited processor speed, code space and energy use environment. The hardware implementation is carried out on Firefly-RK3399 embedded development board, and a lightweight AES encryption communication system is designed based on TCP/IP socket communication scheme. 32-byte message content and 128-bit time dynamic key are used to perform system experiment. The results show that compared with the traditional AES encryption algorithm, the encryption and decryption speed of the proposed method can be increased by 15.02% without reducing security under the same environmental factors.

Key words: AES encryption; sockets; embedded; communication security

EEACC: 6140 doi: 10.3969/j.issn.1005-9490.2023.01.005

面向嵌入式设备的轻量级 AES 加密通讯系统的 设计与实现 *

刘 政¹, 代培培¹, 邢建平^{1*}, 孙启玉², 刘玉峰²
(1.山东大学微电子学院, 山东 济南 250101; 2.水发智慧产业集团有限公司, 山东 济南 250101)

摘 要: 针对嵌入式设备, 基于 128 位 AES 加密算法提出了一种基于软件的轻量化实现方案, 利用 T 型查找表存储方案, 将 AES 加密算法中字节替代、行移位和列混合操作转化成查表操作, 并优化内存访问机制, 使 AES 加密算法能够在有限的处理器速度、代码空间、能源使用的环境中运行。在 Firefly-RK3399 嵌入式开发板上进行了硬件实现, 基于 TCP/IP 的套接字通讯方案, 设计了轻量级 AES 加密通讯系统。采用 32 字节报文内容、128 位的时间动态密钥, 进行系统实验, 结果表明较传统 AES 加密算法, 在环境因素相同的情况下, 所提方法在不降低安全性的同时, 加解密速度可提升 15.02%。

关键词: AES 加密; 套接字; 嵌入式; 轻量级

中图分类号: TN929.5

文献标识码: A

文章编号: 1005-9490(2023)01-0029-07

高级加密标准 (Advanced Encryption Standard, AES) 在 1998 年由 J. Daemen 等^[1]提出, 于 2001 年被美国国家标准与技术研究所 (NIST) 选中并使用^[2]。AES 是一种包含替换和变换的标准化区块加密算法, 由于其出色的安全性, 被应用于很多领域当中。AES 加密算法在信息安全日益重要的今天显得尤为重要, 密码学界对其软件实现进行了广泛的研究^[3-4]。

随着嵌入式便携设备的信息安全需求增大^[5-6], 传统的 AES 加密算法因为占用较多的硬件

资源, 不能较好地适用于嵌入式设备, 故需求一种基于软件的 AES 加/解密的轻量级实现方案。Dhanda 等^[7]在 2020 年经过调研, 提出 AES 加密算法是最适合在物联网中进行轻量级密码研究的基础算法之一, 轻量级 AES 加密算法被广泛探索。Gomes 等^[8]提出实现由流水线操作进行的组合 S 盒, 增加了一定量地并行计算的进程; Alshammari 等^[9]依托混沌 S 盒和 AES 加密算法提出偏重解决图像加密的物联网端解决方案。

本文设计的嵌入式端轻量级 AES 加密短报文

项目来源: 国家重点研发计划项目 (2021YFB1407001); 山东高速集团省级项目 (2020BZ06-01)

收稿日期: 2021-11-08 修改日期: 2022-01-02

播报系统,融合了 128 位轻量级 AES 加密算法、基于 TCP/IP 的 Socket 及时通讯技术,以 Firefly-RK3399 开发板为硬件平台,基于 Linux 操作系统,完成集合时间戳的动态密钥和 32 字节的短报文内容设计。系统用于需要实时播报信息的嵌入式便携设备,以达到安全通信效果。

1 整体方案设计

本系统应用于需实时向服务站播报短报文的移动设备,目的是安全地将短报文发送到服务站,避免明文播报可能导致的信息泄露。系统整体框架如图 1 所示,硬件部分主要为嵌入式客户端和远程监控服务站,软件主要为轻量级 AES 短报文加解密部分和基于 TCP 的 Socket 通信部分。

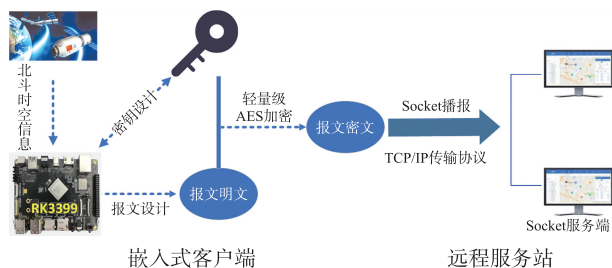


图 1 系统架构图

嵌入式客户端部分由 Firefly-RK3399 开发板、北斗定位模组、轻量级 AES 加密算法、基于 TCP 的 Socket 通信客户端等组成。此部分的主要作用为:获取当前设备的实时信息,进行报文设计,将报文进行轻量级 AES 算法加密,通过 Socket 通信客户端将密文播发至服务端。

远程监控服务站部分由基于 linux 系统的服务器、轻量级 AES 解密算法、基于 TCP 的 Socket 通信服务端等组成。此部分的主要作用为:接收嵌入式客户端播发的密文,通过轻量级 AES 解密算法将密文解密为明文,进行报文分析并存储至服务器便于后续分析。

2 Socket 通信系统

2.1 Socket 通信简介

Socket 最早是应用于 UNIX 系统的一种通信模式^[10],UNIX 系统下的所有操作均是面向文件的,即 Socket 的通信模式也是基于文件操作的。客户端和服务端均对一个文件完成“打开-读/写-关闭”的操作,通过此文件传输信息,完成通信。

在现有的 Socket 通信体系中,依托其使用的 TCP/IP 协议,可以将网络体系结构抽象地分为四个层次:应用层、传输层、网络层和数据链路层。应用

层为与用户交互的层面,Socket 应用以进程的形式存在于用户计算机内,服从调度;传输层在网络体系结构中是非常重要的层次,其主要服务对象是两个主机中的进程,为主机进程提供通信服务,利用 TCP/UDP 等协议进行数据传输;网络层介于传输层和数据链路层之间,其任务是将数据传送到各自的目的地址,面向传输层提供端到端的数据传送服务,直接或间接使用 Socket 端口套接字来访问 TCP/IP 协议网络栈;数据链路层可以分为两个小方面,物理层和数据层,物理层的主要功能是利用传输介质为数据层提供物理连接的保证,实现数据流的透明传输,数据层通过各种通信协议,将有差异的物理信号转换为无差别的、可传输的数据链路信号。

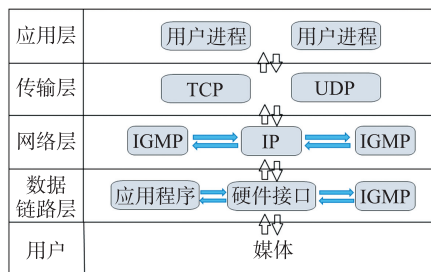


图 2 Socket 通信网络结构

Socket 直译为“插座”,这个“插座”可以看做一个负责用户和服务端之间通话的电话插座,存在于两点通话的线路之间;Socket 依托 TCP/IP 通信协议,客户端所绑定的端口,可以看做“插座”上面的“插口”,每个“插口”仅限一个用户使用,一个端口在同一时间只能被一个用户占用。在服务端的监听行为开始后,一直等待客户的请求命令,用户请求本机的 IP 地址的端口,本机就接通用户。在工程应用中,可以抽象的将“Socket”理解为在应用层和传输层之间的一个抽象层,是一个负责数据文件的修改的“插座”或者“管道”。

2.2 Socket 通信模式

Socket 通信采用的是较为常见的客户端-服务端通信模式。客户端-服务端模型如今在分布式计算、多端通信体系以及计算机网络中是一个常见的概念,在实际应用中有许多客户端-服务端程序案例。在客户端-服务端模型中,网络上每个计算机终端或进程可以是客户端或服务端,客户端是允许用户访问和查看其数据的程序或终端,每个客户端均可与服务端进行通信和连接。

Socket 的通信模式被广泛应用于各类终端之间的数据交互,其保证了服务器与不同客户端之间的稳定通信。在一般情况下,客户端作为发起通信请求的一端,服务器作为等待响应客户端请求的一端。

服务器与客户端在通信前都会建立一个 Socket 对象,完成建立后,服务器会进入循环监听模式,等待客户端的连接请求。在服务器得到客户端的请求后,服务器对请求进行判别,是否同意连接,同意后,服务器所建立的 Socket 对象与客户端所建立的 Socket 对象进行绑定,客户端对 Socket 进行更改,服务器会同步收到此更改,以达到数据交互的目的,并依据规则进行同类型回复。

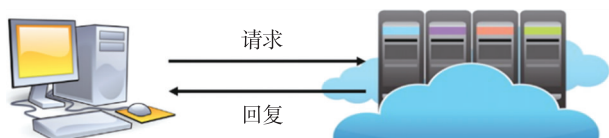


图 3 客户端-服务器体系示意

3 轻量化 AES 加密算法

3.1 AES 加密算法简介

AES 是一种经 FIPS 批准的加密算法,可用于保护电子数据。该算法的原名为 Rijndael,于 1998 年由两位比利时密码学家发明发表^[1],并于 2001 年在 FIPS PUB 197 中对计算机安全标准进行规范。AES 算法是一种标准的对称块状密码,能够使用 128、192 和 256 比特的密码密钥。

AES 加密算法的加密过程和解密过程是完全对称的,加密和解密过程使用相同的密钥,是最常见的对称加密算法之一,其流程如图 4 所示。

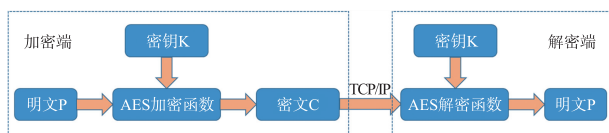


图 4 AES 加密过程示意图

3.2 AES 加密过程

AES 加密算法是一种分组加密算法,分组加密将明文分成多个区块,每区块占用的位数相同,每个块长度为 128 位,支持 128、192 和 256 位的密钥长度。明文首先被复制到一个四行四列的方阵,称其为状态矩阵。在添加初始轮密钥后,状态矩阵通过执行轮函数 10、12 或 14 次(分别针对 128 位、192 位和 256 位密钥)进行变换,最终状态为密文。根据密钥的长度不同,加密轮数也不同,对应轮次见表 1。

表 1 AES 加密算法分类

AES	密钥长度	分组长度	加密轮次
AES-128	128	128	10
AES-192	192	128	12
AES-256	256	128	14

AES 的每轮加密过程包含四个变换:字节替代、行移位、列混合和轮密钥,加密过程的最后一轮不包括列混合操作。字节替代操作是一种非线性字节替换,它使用替换表对状态矩阵的每个字节进行独立操作。在行移位操作中,每个组块的状态矩阵的第 1~4 行分别左移 0~3 个字节。

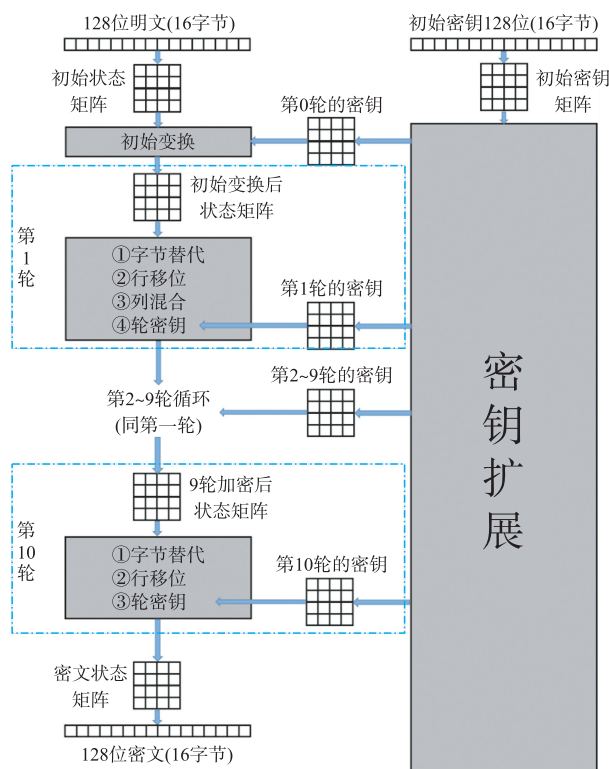


图 5 AES 算法加密过程

列混合操作是对状态矩阵逐列进行操作,将列视为多项式并乘以一个常数 4×4 上的矩阵,计算过程如式(1)所示,其中第 $j(0 \leq j \leq 3)$ 列的计算过程如式(2)所示,列混合计算过程均在 $GF(2^8)^{[1]}$ 域上完成。

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} \quad (1)$$

$$\begin{cases} S'_{0,j} = (2 \times S_{0,j}) + (3 \times S_{1,j}) + S_{2,j} + S_{3,j} \\ S'_{1,j} = S_{0,j} + (2 \times S_{1,j}) + (2 \times S_{2,j}) + S_{3,j} \\ S'_{2,j} = S_{0,j} + S_{1,j} + (2 \times S_{2,j}) + (3 \times S_{3,j}) \\ S'_{3,j} = (3 \times S_{0,j}) + S_{1,j} + S_{2,j} + (2 \times S_{3,j}) \end{cases} \quad (2)$$

在轮密钥转换中,通过简单的按位异或(XOR)

操作将轮密钥添加到状态。解密过程为进行相应的逆运算。密钥被扩展成由 44 个 32 位字所组成的数组,每轮有四个不同的区块作为该轮的轮密钥,每个区块的长度为 128 位。

针对嵌入式设备轻量化计算量的要求,本文设计采用 AES-128 位加密算法。加密和解密过程各经过十轮运算,加密过程的前九轮运算每轮运算包含四个过程,字节代替、行移位、列混合和轮密钥,第十轮与前九轮的相比缺少了列混合的过程。解密过程是加密的逆过程。

3.3 AES 轻量化

3.3.1 查找表

在具有 32 位或更多位数的系统中,例如在 Firefly-RK3399 的嵌入式设备中,可以通过将字节替代和行移位步骤与列混合步骤相结合,将它们转化为一连串的表格查询,从而加快该加密过程的执行速度。这样可以在 32 位以上的平台上非常有效地缩短加密时间,以额外的较小存储为代价,存储查找表。Li 等^[12]曾提出利用掩码查找表的方式简化 AES 加密计算量,但未在嵌入式端进行实验。

通过使用 T 查找表,可以实现一轮加密中的字节替代、行移位、列混合三个操作^[13]。定义四个 T 查找表,字节替代的输入状态矩阵用 a 表示,列混合的输入状态矩阵用 d 表示,字节替代操作用 S_d 表示。

$$\begin{aligned} T_0[a] &= \begin{bmatrix} 02 \cdot S_d[a] \\ 01 \cdot S_d[a] \\ 01 \cdot S_d[a] \\ 03 \cdot S_d[a] \end{bmatrix} & T_1[a] &= \begin{bmatrix} 03 \cdot S_d[a] \\ 02 \cdot S_d[a] \\ 01 \cdot S_d[a] \\ 01 \cdot S_d[a] \end{bmatrix} \\ T_2[a] &= \begin{bmatrix} 01 \cdot S_d[a] \\ 03 \cdot S_d[a] \\ 02 \cdot S_d[a] \\ 01 \cdot S_d[a] \end{bmatrix} & T_3[a] &= \begin{bmatrix} 01 \cdot S_d[a] \\ 01 \cdot S_d[a] \\ 03 \cdot S_d[a] \\ 02 \cdot S_d[a] \end{bmatrix} \end{aligned} \quad (3)$$

每个 T 查找表有 256 个 4 字节的数据,需要 4 KB 的存储空间。轮密钥之前的操作如下式:

$$\begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} = T_0[a_{0,j+c0}] \oplus T_1[a_{1,j+c1}] \oplus T_2[a_{2,j+c2}] \oplus T_3[a_{3,j+c3}] \quad (4)$$

如式(3),03-S 表可以被计算出来,这可以通过预计算消除算法中的多步骤操作。

$$03 \cdot S = (01 \oplus 02) \cdot S = (01 \cdot S) \oplus (02 \cdot S) \quad (5)$$

这些 T 查找表可以存储在嵌入式设备的 ROM

结构中,可以快速实现字节替代、行移位和列混合步骤,并获得更高的操作速度、更高的运行速度,进一步节省逻辑资源。

3.3.2 加速内存访问

通过减少对 CPU 的访问以及优化软件流程,可以有效地减少计算时间。为了减少内存访问次数,以及缩短内存访问时间,提出了关于内存访问有关的优化方法,可以减少在 RK3399 开发板上运行程序的时钟周期数。使用 RK3399 开发板的闪存来存储指令和表格, RK3399 的等待状态为零,在 3.3 V 的电源电压下, CPU 始终在 120 MHz 的状态下运行, RK3399 存储芯片和时钟芯片的时钟频率依赖性导致在访问闪存时会引入一些等待状态的 RAM。 RK3399 开发板有三个不同区域的 RAM 可用, SRAM0、SRAM1 和 CPU 均引入等待状态,可以将三个 RAM 均保持在等待状态,可以排除惩罚周期的影响,减轻流水线负载。

4 系统实验

4.1 硬件选用

为了对本文提出的面向轻量级 AES 加密通信系统进行实验测试,采用 Firefly-RK3399 开发板作为客户端实验平台,搭载 Cortex-A72+Cortex-A53 大小核架构,内置 ARM Mali-T860 MP4 四核 GPU^[14],可对数据信息进行高效处理,平台采用 Linux 操作系统作为开发环境,实物如图 6 所示。

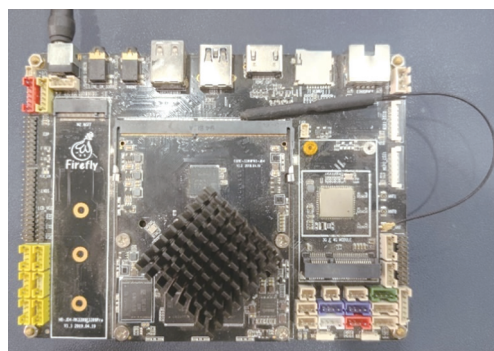


图 6 RK3399 开发板实物图

4.2 软件设计流程

4.2.1 报文设计

本文设计采用 128 位 AES 加密算法,单次加密输入矩阵位数为 128,本设计报文选用 2 个 16 位的 16 进制编码,单次播报传输 32 个 16 进制编码。

实验采用报文为 32 位 16 进制编码,其中前 19 位描述经纬度,选定 1×10^{-6} 度为最小距离单位;后 12 位描述时间,选定 1 ms 为最小时间单位;最后一位为信息标志位,0 表示信息无异常,1 表示信息异常。

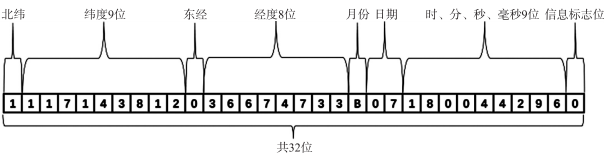


图 7 报文结构图

4.2.2 密钥设计

实验采用时间动态密钥,16 位密钥包含 8 位时间信息和 8 位客户端编号密码,时间信息每小时更换一次,客户端密码由服务器设计分发,与客户端绑定。

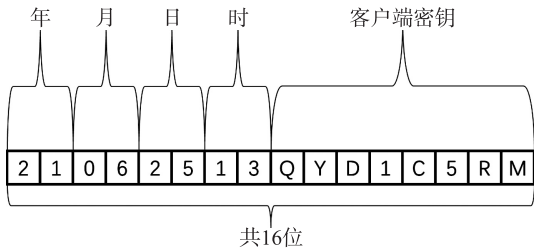


图 8 密钥结构图

4.2.3 客户端-服务端软件设计

一个完整的 Socket 通信过程分为客户端过程和服务端过程。客户端过程比较简单,创建一个 Socket 进程,并尝试连接服务器,将 Socket 与远程主机连接。值得注意的是,只有在基于 TCP/IP 通信协议的 Socket 通信过程中,才有连接的概念。基于 UDP、ICMP 的 Socket 通信过程没有连接的概念,客户端只发送数据。本文设计的基于 TCP/IP 的 Socket 客户端负责发送数据,读取服务器反馈数据,直到通信完毕,关闭连接,结束 TCP 对话。

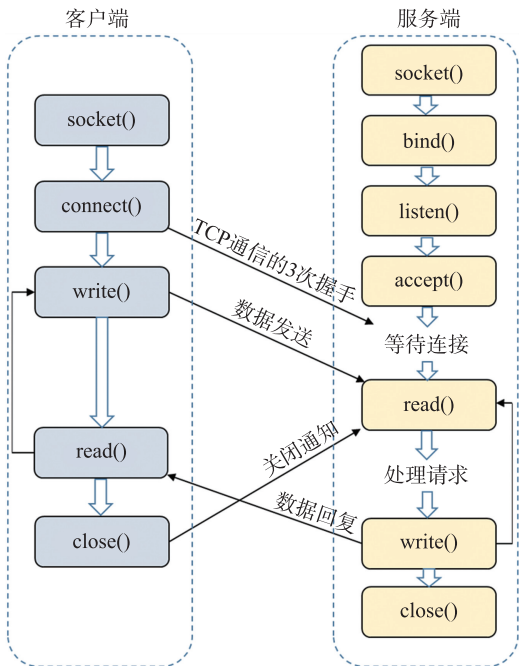


图 9 客户端与服务端通信过程

相对于客户端过程来说,服务端的流程复杂一些,服务端需要先建立一个流式 Socket,并对 Socket 进行初始化,与本机 IP 地址和端口进行绑定,使用 TCP 等待客户端的连接。在等待连接的过程中,客户端一直保持等待连接即监听状态,如果 TCP 的三次握手通信认证通过后,即进入连接状态,接受客户端的请求,并返回需求数据,直到数据交互完毕。最后关闭连接,交互结束。

5 实验结果

5.1 密文安全性

AES 加密算法已是较为完善的加密方式,使用无穷尽法破解 128 位及以上的 AES 加密方式是不现实的,现有个人计算机的算法,使用无穷尽法破解 AES-128 位加密,需要花费几十亿年。对于加密终端,常用的破解方式之一是差分功耗分析攻击。差分功耗分析攻的原理是利用处理器进行不同密钥加密时产生的功率不同,对功率进行记录分析,利用功率的差分性分析密钥组成。在密文安全性分析方面,使用差分功耗分析攻击对传统 AES-128 加密算法和本文设计的轻量级 AES-128 位加密算法进行破解攻击,使用 DPA 仿真测试平台,具体参数见表 2。

表 2 DPA 仿真测试平台设置攻击参数

测试平台攻击加密方式	电路时钟频率/MHz	功耗数据记录周期/ns	时钟周期记录点数
传统 AES	2.5	1	400
轻量级 AES	2.5	1	400

本文选择对 AES-128 算法的前两轮加密过程进行攻击,得到前两轮瞬时差分功耗数据,如图 10 所示。由实验结果可知,轻量化前后的 AES 加密算法当输出状态矩阵位数高于 80 后,电路的功耗差分泄露都趋近于 0,这表示当输出状态矩阵为 128 位时,此加密算法很难被攻击,即轻量化前后密文安全性一致。

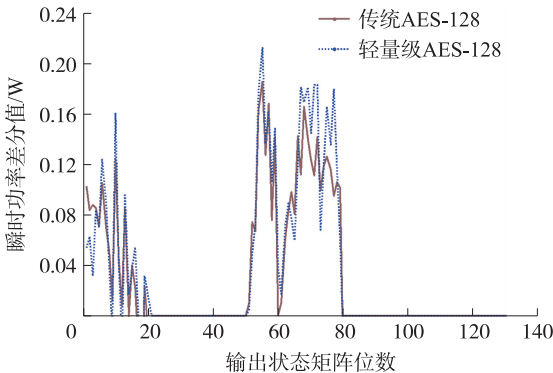


图 10 128 位 AES 加密轻量化前后的差分功耗

5.2 播报可靠性与延迟

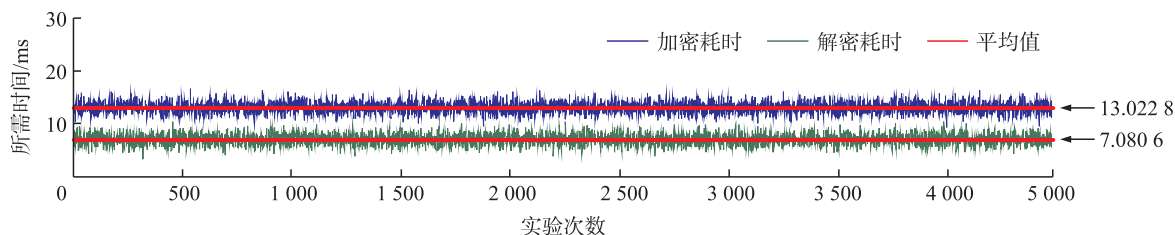
为了测试此加密通讯播报系统的可靠性与稳定性,在实验环境下进行 4 475 次的连续播报,实验中加解密占用时间情况见图 11 所示,对信息进行分析后得到结果。

结果显示在嵌入式客户端进行的加密部分,轻量化前后所占用的时间分别为 13.022 8 ms 和 10.921 6 ms,提升 16.1%;在服务器端进行了解密部分,

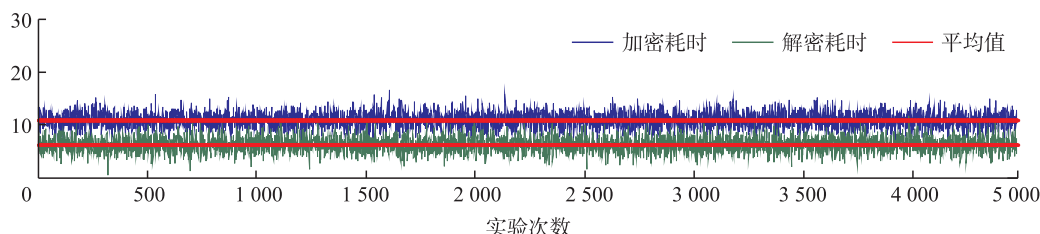
轻量化前后所占用的时间为 7.080 6 ms 和 6.164 0 ms,提升 14.3%。经过系统实验,在加解密过程应用查找表存储和加速内存访问的轻量化应用后,加解密合计时间共提升 15.02%,达到预期效果。

表 3 轻量化前后实验占用时间对比 单位:ms

加密方式	加密耗时	解密耗时	合计	优化
传统	13.02	7.08	20.10	
轻量化	10.92	6.16	17.08	15.02%



(a) 传统AES128位加密算法耗时统计



(b) 轻量级AES12 位加密算法耗时统计

图 11 轻量化前后加/解密耗时对比图

6 结论

本文在嵌入式端寻求一个基于 AES 加密算法的轻量级应用,以加密通信播报系统作为实验切入点,采用字节替代和行移位步骤与列混合步骤相结合存储为查找表并优化内存访问机制,进行轻量化的应用。通过在 RK3399 开发板进行试验,表明进行轻量化后的 AES 加密算法可将加解密时间缩短 15.02%。

参考文献:

- [1] Daemen J, Rijmen V. AES Proposal: Rijndael[EB/OL]. 1998. <https://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf>.
- [2] Daemen J, Rijmen V. The Advanced Encryption Standard [A]//In: Daemen J, rijmen V. The design of Rijndael: AES—The Advanced Encryption Standard (Information Security and Cryptography. New York: Springer, 2002: 1–8.
- [3] Ali L, Aris I, Hossain F S, et al. Design of an Ultra High Speed AES Processor for Next Generation IT Security[J]. Computers & Electrical Engineering, 2011, 37(6): 1160–1170.
- [4] Dutta I K, Ghosh B, Bayoumi M. Lightweight Cryptography

- for Internet of Insecure Things: A Survey[C]//2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019: 475–481.
- [5] 曾小波, 易志中, 焦歆. 基于 51 核的 AES 算法高速硬件设计与实现[J]. 电子科技, 2016, 29(1): 36–39.
- [6] 贺依盟, 周亚军. AXI 总线加密模块的设计与验证[J]. 杭州电子科技大学学报, 2016, 36(1): 57–62.
- [7] Dhanda S S, Singh B, Jindal P. Lightweight Cryptography: A Solution to Secure IoT[J]. Wireless Personal Communications, 2020, 112(3): 1947–1980.
- [8] Gomes O S M, Moreno R L, Pimenta T C, et al. A Fast Cryptography Pipelined Hardware Developed in FPGA with VHDL [C]//2011 3rd International Congress on Ultra Modern Telecommunications & Control Systems & Workshops, Budapest, Hungary, IEEE, 2011: 1–6.
- [9] Alshammari B M, Guesmi R, Guesmi T, et al. Implementing a Symmetric Lightweight Cryptosystem in Highly Constrained IoT Devices by Using a Chaotic S-Box[J]. Symmetry, 2021, 13(1): 129–148.
- [10] Coffield D, Shepherd D. Tutorial Guide to Unix Sockets for Network Communications[J]. Computer Communications, 1987, 10(1): 21–29.
- [11] Murphy S, Robshaw M J B. Essential Algebraic Structure within the AES[C]//22nd Annual International Cryptology

- Conference, Santa Barbara, CA, USA, 2002:1-16.
- [12] Li M, Gan J, Cheng S, et al. Design of Lightweight AES Algorithm Based on Masked Lookup Table [C]//2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT 2019), Dalian, China, 2019:437-441.
- [13] Zhang X, Li M, Hu J. Optimization and Implementation of AES Algorithm Based on FPGA[C]//2018 IEEE 4th International Conference on Computer and Communications (ICCC), Chengdu, China, 2018:2704-2709.
- [14] 马永杰,宋晓凤. 基于 YOLO 和嵌入式系统的车流量检测[J]. 液晶与显示, 2019, 34(6):613-618.



刘 政(1997—),男,河北保定人,硕士研究生,研究方向为加密通信、嵌入式开发,liuzheng2020sdu@163.com;



邢建平(1969—),男,山东烟台人,博士,教授,研究方向为嵌入式与物联网应用、北斗系统应用,sdxingjianping@163.com。