

面向物联网的轻量级 AES 及其在安全存储中的应用

陈嘉玲, 王 洁, 徐焕章, 孟建熠

(浙江大学 超大规模集成电路设计研究所, 浙江 杭州 310027)

摘 要: 为解决物联网中安全 SOC 芯片资源和性能的平衡问题, 提出了一种软硬件协同工作的可资源复用的轻量级 AES(Advanced Encryption Standard)算法实现方式. 该设计采用同步 FIFO 作为软硬件交互接口以实现软硬件协同工作的 AES 加解密算法, 并改进密钥扩展算法, 提高软件密钥扩展的安全性. AES 模块硬件资源同时可供安全 SOC 芯片复用, 对存储系统数据进行简单加密加扰, 节省硬件资源. 该设计面积约为 2418 个等效门, 并为芯片安全存储系统中存储器读写模块节省约 20% 的面积, 功能全面、安全性高, 适合轻量级安全应用.

关键词: AES 算法; 轻量级; 软硬件协同; 资源复用

中图分类号: TP302.1

文献标识码: A

文章编号: 1000-7180(2018)06-0106-06

Lightweight AES for Internet of Things and its Application in Secure Storage System

CHEN Jia-ling, WANG Jie, XU Huan-zhang, MENG Jian-yi

(Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China)

Abstract: In order to achieve balance between area and performance of secure SOC in the Internet of things(IOT), a lightweight and resource-sharing implementation of Advanced Encryption Standard (AES) algorithm is proposed based on cooperation of software and hardware. This design exploits a synchronous FIFO as the interface between software and hardware to implement AES encryption and decryption based on cooperation of software and hardware. The key-expansion algorithm is optimized so as to improve security. The hardware resources of the AES module can also be reused by the memory system for data encryption and decryption in the secure SOC, thus saving hardware resources. This design consumes about 2418 equivalent gates for hardware, and saves circuit resources by 20% for the memory reading and writing module of secure storage system, which has comprehensive functions and high security with tiny hardware area, and is suitable for lightweight secure SOC.

Key words: AES algorithm; lightweight; cooperation of software and hardware; resource reuse

1 引言

随着物联网的深入发展, 轻量级安全 SOC 芯片得到了广泛应用, 低成本低功耗的轻量级密码算法的实现成为主流需求. 高级加密标准 AES 算法应用广泛^[1], 具有安全性高、易于实现的特点. 针对 AES 的硬件加速实现及优化已有许多研究^[2], 这些方法吞吐率高、运算速度快, 但是耗费资源大, 硬件成本高, 不适合轻量级安全芯片. 纯软件实现 AES 算法不需要额外的硬件资源, 但运算速度慢、效率低, 安

全性也较差. 因此轻量级 AES^[3-5]的实现要取得资源和性能的平衡显得尤为重要.

本文针对资源受限的安全 SOC 芯片设计, 提出了一种软硬件协同工作的可资源复用的轻量级 AES-128 实现方式, 并通过最大程度复用硬件结构和改进密钥扩展算法, 达到加解密功能齐全、硬件资源消耗少、安全性高的设计目标.

2 AES 算法原理

2.1 轮运算

AES 是一种对称加解密算法,主要分为加解密过程和密钥扩展过程.加密和解密算法的输入是一个 128 位分组,在 FIPS PUB 197 中,这个分组被描述为 4x4 的字节方阵,被复制到状态矩阵,并在加密或解密的各个阶段被修改.同样地,128 位密钥也被描述为字节的方阵.

AES-128 的算法流程如图 1.

加解密各十轮运算,加密前九轮每轮运算为:字节代替(SubBytes)、行移位(ShiftRow)、列混合(MixColumns)、轮密钥加(AddRoundKey)四个步骤,第十轮与前九轮的差别在于没有列混合运算.解密是加密的逆过程.输入的密钥被扩展成由 44 个 32 位字所组成的数组 W[0]-W[43],每轮有四个不同的字(128 位)作为该轮的轮密钥.

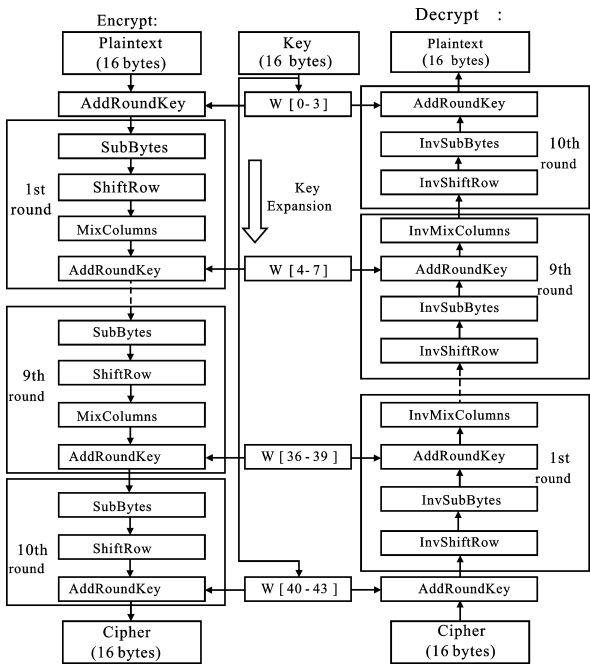


图 1 AES 原理

字节代替即用 S 盒完成分组的字节到字节的代替;行移位是简单置换,状态矩阵的第 1,2,3,4 行分别左移 0,1,2,3 个字节;列混合是有限域 GF(2⁸)上的运算,通过式(1)将一个固定矩阵与状态矩阵相乘;轮密钥加将前三步计算出的状态矩阵与该轮运算相应轮密钥进行异或操作.解密时分别使用这四种运算的逆变换.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_0 & S_4 & S_8 & S_{12} \\ S_1 & S_5 & S_9 & S_{13} \\ S_2 & S_6 & S_{10} & S_{14} \\ S_3 & S_7 & S_{11} & S_{15} \end{bmatrix} = \begin{bmatrix} S_{0'} & S_{4'} & S_{8'} & S_{12'} \\ S_{1'} & S_{5'} & S_{9'} & S_{13'} \\ S_{2'} & S_{6'} & S_{10'} & S_{14'} \\ S_{3'} & S_{7'} & S_{11'} & S_{15'} \end{bmatrix} \quad (1)$$

2.2 CBC 模式

对于很长的消息,AES 最常见的模式——ECB 模式不够安全^[6],而 CBC 模式引入了反馈,可以克服 ECB 模式的弱点.

CBC 模式如图 2,加密算法的输入是当前的明文组和上一个密文组的异或,解密时每个密文分组分别进行解密,再与上一块密文异或即可恢复出明文.

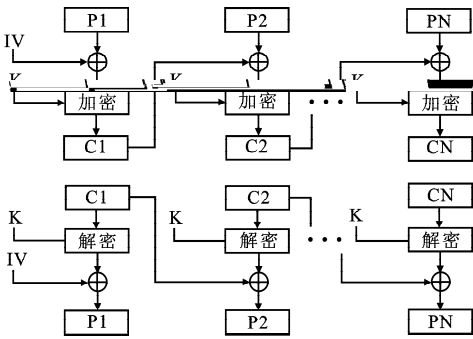


图 2 密文分组链接(CBC)模式

2.3 等价逆算法

图 1 所示的 AES 加密算法和解密算法不同,为能够结构复用,减少硬件复杂度,有一种等价的解密算法实现方式.由式(2)、(3)可知,等价解密算法需要交换逆字节代替和逆行移位的顺序,同理交换逆列混合和轮密钥加的顺序,但是要改进密钥算法,对除第一次和最后一次轮密钥加的密钥进行逆列混合操作.

$$\begin{aligned} \text{InvShiftRow}[\text{InvSubBytes}(S_i)] &= \\ \text{InvSubBytes}[\text{InvShiftRow}(S_i)] &= \\ \text{InvMixColumns}(S_i \oplus W_j) &= \\ [\text{InvMixColumns}(S_i)] \oplus [\text{InvMixColumns}(W_j)] &= \end{aligned} \quad (2)$$
$$(3)$$

3 算法研究与设计

3.1 软硬件划分

3.1.1 轮密钥加、字节代替和行移位

轮密钥加需要获取每轮密钥,软件实现降低了系统安全性,硬件实现仅需要异或门,因此该运算适

合硬件实现.

S 盒变换的软件实现有查表法和构造法. 查表法消耗大量存储资源, 构造法需要 CPU 进行大量复杂的有限域运算, 运算速度慢. 而硬件实现 S 盒可以大大提高效率, 且硬件开销可以通过适当的方法降低. 行移位不占用硬件资源, 只需改变连线关系即可, 通常与字节代替结合起来实现.

S 盒的硬件实现方式有查表法, 扩展的欧几里德算法及变种^[7], 费马定理算法和域变换法^[8]. 查表法易于实现, 但消耗存储资源; 扩展的欧几里德算法效率较高, 但硬件复杂度高且面积大; 费马定理算法适合流水线技术, 但效率较低, 且流水线结构的 AES 不支持 CBC 模式; 域变换法的 S 盒和逆 S 盒可以复用模逆电路结构, 资源消耗最小. 因此本文采用域变换法实现字节代替.

传统的 AES 在一个周期内完成一个状态矩阵的字节代替和密钥扩展, 所需的几十个 S 盒、逆 S 盒消耗大量组合逻辑资源. 本文对于 S 盒的数据通路进行分时复用, 将其数量减少为一个, 极大地节省了因并行使用多个 S 盒产生的硬件资源, 从而减小面积. 基于运算频率要求不高的前提, 分时复用不会导致时序问题.

3.1.2 列混合

列混合涉及有限域 $GF(2^8)$ 上的运算, 包括异或、移位等, 在仅使用一个 S 盒的情况下, 列混合和逆列混合所占硬件资源比重显得尤为突出, 而这部分的运算功能由 CPU 简单的异或、移位指令即可实现. 软件实现导致的效率降低程度相对较小, 因此将列混合运算划入软件实现.

3.1.3 密钥扩展算法及优化

为节省资源, 将密钥扩展算法用软件实现, 并将算好的每一轮密钥存入芯片公用的 SRAM 中, 进行轮运算时读出使用, 基本不影响运算效率.

如图 3 所示为相邻轮数的密钥关系, g 为一个复杂函数, 可知每轮密钥与上一轮、下一轮密钥有密切关系, 例如若获知 $W[i+4]-W[i+7]$, 则可轻易推导出 $W[i]-W[i+3]$, 软件实现密钥扩展使得安全性降低. 本文提出一种优化方式, 对扩展完成的每一轮密钥进行简单加密加扰并存入 SRAM, 在硬件读取密钥后进行解密解扰, 获得每轮的真实密钥, 再进行加解密运算. S 盒和行移位的机制具有较高安全性, 故对密钥的加密加扰也可采用同样的策略, 即软件对每个字节的密钥进行 S 盒代替以及顺序扰乱, 由硬件使用对应的逆

运算恢复出真实的密钥, 提高软件密钥扩展的安全性.

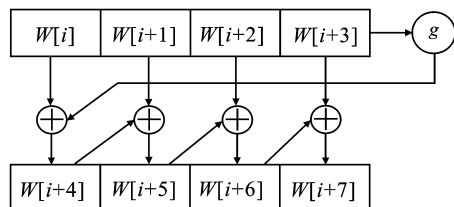


图 3 密钥扩展

3.1.4 软硬件交互

划分完软硬件实现功能, 即轮密钥加、字节代替和行移位由硬件完成、列混合和密钥扩展由软件完成之后, 需要提供交互的接口. 本文采用一块 16×8 的同步 FIFO, 通过 FIFO 控制逻辑进行操作, AES 硬件和 CPU 通过读写寄存器均可操作 FIFO. 轮运算时, 硬件部分完成轮密钥加、字节代替、行移位后, 将状态矩阵的数据依次写入 FIFO, CPU 读出 FIFO 的数据, 进行列混合运算, 运算完成后将状态矩阵的数据依次写入 FIFO, 硬件读取后继续下一轮运算.

本文设计的软硬件协同工作实现的 AES 算法流程如图 4 所示. scramble 和 descramble 分别指对密钥的加扰和解扰, 图中虚线框为软件实现部分.

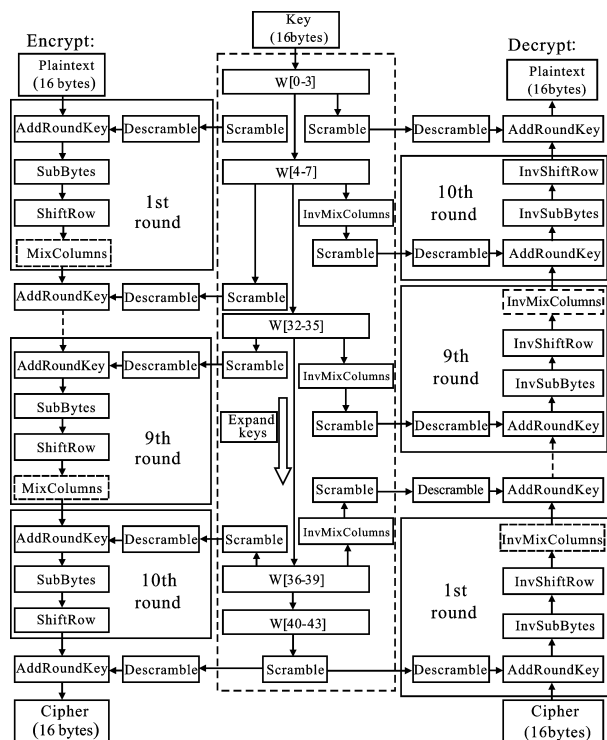


图 4 改进后软硬件协同工作的 AES 算法流程

3.1.5 AES 模块复用

AES 模块(aes_top)在安全芯片中的应用, 不局

状态被清除. 对于最终轮的运算, 硬件算完并压栈之后马上弹栈, 完成最后一次轮密钥加运算, 得出加密结果.

因加解密状态机结构复用, 故解密时通过控制信号 dec 将 S 盒设为逆 S 盒, 同时改变连线顺序, 将行移位改为逆行移位, 软件中进行逆列混合运算, 软件控制流程与加密完全相同.

4.1.3 S 盒

基于域变换的 S 盒运算利用 $GF(2^8)$ 与 $GF(2^4)^2$ 同构性, 将 $GF(2^8)$ 上的求逆转换为 $GF(2^4)^2$ 上的求逆运算, 简化求逆运算. 原理如图 8 所示, S 盒与逆 S 盒的模逆电路 Inversion 一致, dec 为解密信号, Affine 为仿射变换, Affine-1 为逆仿射变换. dec 为 0 时先求逆再作仿射变换, 用作 S 盒; 为 1 时先作逆仿射变换再求逆, 用作逆 S 盒. S 盒的设计达到结构复用的目的, 节省硬件资源.

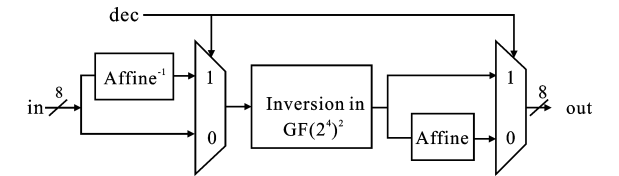


图 8 域变换法结构复用的 S 盒

4.2 软硬件协同工作

软件部分负责工作流程控制、密钥扩展和列混合运算. 密钥扩展在 AES 加解密前算好每一轮的密钥并作加扰, 存至 SRAM, 不影响整体加解密效率. 如图 9 所示的软硬件协同工作的流程为: 先输入明文, 并将该轮密钥从 SRAM 取出后输入, 再配置控制寄存器 (ECB 或 CBC 模式、数据大小端等) 使 AES 指示硬件开始运算, 同时软件开始检测状态位. 随后由状态机进行运算, 按照状态矩阵的顺序, 每一个时钟周期算出状态矩阵的一个字节值并存入 FIFO. 由式(1)可知, 计算 $S_0 \sim S_3$ 时只需 $S_0 \sim S_3$, 而不用等后面的 12 个字节全部算完, 且列混合这段函数可循环使用, 故可并行操作软硬件. 当运行完 S_3 状态即 FIFO 中有 4 个字节数据 ($S_0 \sim S_3$) 时, 硬件将状态位置 1, 软件检测到后马上从 FIFO 读取这 4 个字节并开始列混合运算, 计算 $S_0' \sim S_3'$ 并存入 FIFO, 此时硬件继续运算 $S_4 \sim S_{15}$, 软硬件并行协同工作, 极大地提高了运算效率. 由于软件运算速度比硬件慢, 故可直接循环进行剩下 12 字节的列混合运算, 运算完成后将状态矩阵的字节值依次写入 FIFO, 随后输入新一轮密钥值并使能 AES, 进入下

一轮运算, 如此重复十轮即可得出结果. 本文软件部分的运算基于 CK-810 CPU, 工作频率 100 MHz.

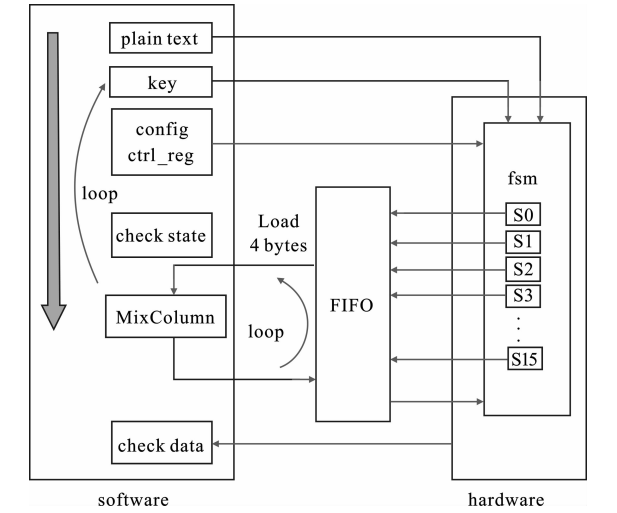


图 9 软硬件协同工作数据流程图

5 实验结果比较与分析

本文针对 128 位数据进行加解密, 仿真时 CPU 频率为 100 MHz, 加解密分别需要 2 209/4 142 个时钟周期, 支持 ECB 和 CBC 两种模式. 在 smic 55 nm 的 CMOS 工艺下综合实现, 占用硬件资源为 2 418 个等效门.

对于一些已有的轻量级 AES 的设计方案, 本文的设计从功能、实现方式、占用硬件资源、时钟周期、功耗等方面与之进行比较, 如表 1 所示.

表 1 AES 算法不同实现方式的比较

采用方案	功能	支持模式	实现方式	占用硬件资源	加/解密时钟周期
文献[8]	加密	ECB	纯硬件	3 100 等效门	160
文献[4]	加/解密	ECB	纯硬件	3 840 等效门	226/880
传统软件方案	加/解密	ECB	纯软件	/	131 480/ 192 961
文献[9]	加密	ECB	软硬件结合	3 100 字节 ROM	256 000
文献[10]	加密	ECB	软硬件结合	4 500 等效门	96 000
本文	加/解密	ECB/CBC	软硬件结合	2 418 等效门	2 209/ 4 142

由表 1 看出, 纯硬件实现 AES 的文献[5]芯片面积较小, 但是只有加密功能, 无法满足解密需求; 文献[4]用最少硬件资源完成了加解密设计, 但是在资源极度受限时, 硬件成本相对仍较大, 且不支持 CBC 模式; 传统软件方案和软硬件结合的文献[9]、[10], 虽然占用较少硬件资源, 但是运算速率过于缓

慢,性能较差.本文相比文献[4],虽然需要较多时钟周期,但是减少了三分之一多的硬件资源并支持 CBC 模式,相比纯软件实现和文献[9]、[10]又提升了几十倍的速度,且该速度完全能够满足物联网中轻量级芯片的加解密应用需求,同时还为安全 SOC 芯片中安全存储系统的存储器读写模块节省了 20% 的资源,硬件利用率高.

6 结束语

本文设计了一种软硬件协同工作、可资源复用的轻量级 AES 实现方案,该方案通过牺牲一定的吞吐率来达到硬件资源最小化的目的,并把硬件资源复用给芯片中的安全存储系统,所使用的密钥加扰和存储数据加密加扰策略具有扩展性而不局限于本文的方法,对于物联网中资源受限而对运算效率要求不高的安全芯片来说,具有广泛的适用性.

参考文献:

- [1] The National Institute of Standards and Technology (NIST). Federal Information Processing Publication 197. 2001.
- [2] 李冬冬,杨军. 并行 AES 算法加密解密电路的高效实现[J]. 微电子学与计算机, 2015, 32(3): 100-103.
- [3] 杨骅,王东辉,张铁军,等. 一种适用于 RFID 标签芯片的 AES 算法结构设计[J]. 微电子学与计算机, 2009, 26(1): 89-92.
- [4] 彭浩,刘恺,张亮,等. 轻量级 AES 加解密芯片设计与实现[J]. 微电子学与计算机, 2014, 31(8): 94-98.
- [5] Hamalainen P, Alho T, Hannikainen M, et al. Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core[C]// Digital System Design: Architectures, Methods and Tools, 2006.

Dsd 2006. Euromicro Conference on. IEEE, 2006: 577-583.

- [6] William Stallings. 密码编码学与网络安全—原理与实践[M]. 王张宜,杨敏,杜瑞颖等,译. 第五版,北京:电子工业出版社,2012.
- [7] Song Haolong, Liang Huaguo, Shan Guohua. Improvement of Inverses in $GF(2^m)$ for Public-key Cryptography System on FPGA[J]. Journal of Chinese Computer Systems, 2010, 31(2): 259-263.
- [8] Yang Hongzhi, Han Wenbao, Zhao Long. Construction method of S-box suitable for hardware implementation: Construction method of S-box suitable for hardware implementation[J]. Journal of Computer Applications, 2010, 30(3): 674-676.
- [9] Prouff E, Giraud C, Aum? nier S. Provably Secure S-Box Implementation Based on Fourier Transform [M]// Cryptographic Hardware and Embedded Systems - CHES 2006. Springer Berlin Heidelberg, 2006: 216-230.
- [10] 窦强,童元满,窦文华. 智能卡中抗高阶功耗攻击 AES 算法实现技术[J]. 计算机工程与科学, 2009, 31(12): 16-19.

作者简介:

陈嘉玲 女, (1992-), 硕士研究生. 研究方向为 SOC 设计与验证. E-mail: chalin819@163.com.

王洁 男, (1987-), 硕士研究生. 研究方向为低功耗 SOC 设计.

徐焕章 男, (1992-), 硕士研究生. 研究方向为生理信号数字信号处理.

孟建熠 男, (1982-), 博士后, 教授. 研究方向为嵌入式高性能低功耗 CPU 设计、低功耗 SOC 设计等.