第34卷 第18期 计 算 机 工 程 2008年9月
Vol.34 No.18 Computer Engineering September 2008
·安全技术· 文章编号：1000—3428(2008)18—0164—03 文献标识码：A 中图分类号：TP309

基于AES的低成本可重构高速加密引擎

梁 伟[1]，徐建波[1,2]，唐明董[1,3]，姜 磊[1]
(1. 湖南科技大学计算机科学与工程学院，湘潭 411201；2. 湖南大学计算机与通信学院，长沙 410082；
3. 中科院计算技术研究所，北京 100080)

摘 要：针对商业加密引擎中硬件资源和电路性能平衡问题，提出一种基于AES的低成本可重构的高速加密引擎的设计方案。该方案在AES加密算法的基础上，根据FPGA内在的结构特点，利用VHDL语言对其加密模块进行描述，改善4级流水线结构，结合密码库的扩展设计，使系统达到实时重构安全策略的目的。通过对高速加密引擎的加密模块的实验仿真结果分析和总体性能评估，证明了该加密引擎不仅具有良好的安全性能，而且在速度和资源性能比方面有优势。

关键词：可重构；VHDL语言；加密引擎

Low-cost Reconstructable High Speed Encryption Engine Based on AES

LIANG Wei[1], XU Jian-bo[1,2], TANG Ming-dong[1,3], JIANG Lei[1]
(1. School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201; 2. School of Computer and Communication, Hunan University, Changsha 410082; 3. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 This paper presents a solution of a high speed, low-cost and reconstructable encryption engine based on the AES encryption aiming to solve the balance problem of hardware resources and circuit performance in commercial encryption engines. According to FPGA inherent structural features, the four-level pipeline structure is improved and the encryption module is described with VHDL. By expanding the password encryption engine, the goal of real-time, reconstructability and security is achieved. Compared with some other encryption engines, this encryption engine has a good safety performance guarantee, speed and resources performance ratio.

【Key words】 reconstructable; VHDL; encryption engine

1 概述

随着信息安全技术的飞速发展，密码技术在电子商务、电子银行和虚拟个人网络系统中得到了广泛的应用。目前许多公司在设计一个适应多种加密算法[1](AES以及针对传统应用的专用算法)的加密引擎，而AES采用的“代替-线性”网络结构是一种基于状态的运算。该算法的优点是设计简单、密钥安装快、需要内存空间少，在所有平台上运行良好，支持并行处理，抗所有已知攻击，所以，特别适合加密引擎中加密算法的选择。

一般加密引擎设计思想为：为了在安全的基础上能充分利用有效的网络带宽，使得加密引擎的运行不仅需要巨大的处理能力，而且还需要很高的实时传输速率。这样将使加密引擎的设计成本增加，开发周期延长，同时也对加密引擎的实时性、灵活性和易实施性提出了很大的挑战。因此，设计一种专用的低成本可重构的加密引擎来快速实现AES加密算法，对开发网络安全加密具有重要的意义。

2 低成本的可重构AES加密引擎的设计

2.1 AES算法的硬件加密原理及过程

AES算法中所有的运算都是完整的字节操作，加解密时把数据分成4行的矩阵，每一列由4 Byte构成，将这种情况称之为状态，所有操作都是在状态之间进行的。

根据加密引擎结构中AES加密算法[2-3]的特点，对整个算法进行了电路的摸块划分，本文主要介绍密钥扩展部分和轮加密操作在FPGA上的实现。密钥扩展实现的基本操作为:求逆，GF(2)上的仿射变换，字节换位，计算圈常数，计算密钥。具体步骤为：

(1)求逆、GF(2)上的仿射变换、字节换位，这与加密过程一致。

(2)计算圈常数，圈常数可计算后再进行造表，按最大圈长度造表。

(3)计算密钥，进行异或运算。

轮加密可以分成4个操作：位变换(ByteSubstitution)行移位(ShiftRow)，列变换(MixColumn)，轮密钥加(AddRoundKey)它们都是对状态(State)进行操作(AES把State看成以字节为单位的4行4列的矩阵)。位变换就是对状态上的每一个字节通过S盒进行变换。行移位是对状态矩阵按字节进行移位操作。因此，对于加密和解密，AES算法一次循环(加密或解密)有4个不同的转换过程：

(1)通过(S-box)进行(ByteSubstitution)位变换。

基金项目：国家自然科学基金资助项目(60673061)；湖南省自然科学基金资助项目(02JJY5006)；湖南省教育厅资助科研基金资助项目(05c182, 06c303)

作者简介：梁 伟(1978－)，男，助教、硕士研究生，主研方向：计算机网络，嵌入式系统；徐建波，教授；唐明董，讲师、博士研究生；姜 磊，讲师

收稿日期：2007-12-05 **E-mail：**idlink@163.com

(2)通过不同偏移量进行状态矩阵行的移动操作 ShiftRows(state)。

(3)对状态矩阵的列数据进行列操作 MixColumns(state)。最后状态矩阵进行轮密钥加 AddRoundKey。加密原理如图 1 所示，解密原理为其加密原理的逆过程，在此不再描述。

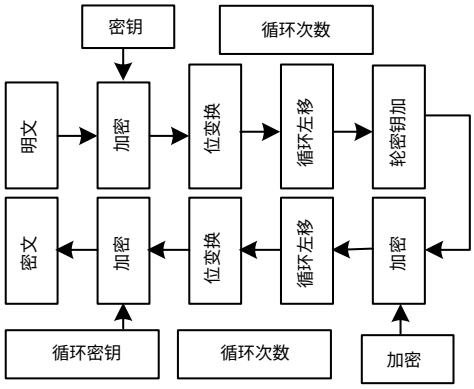

图 1　加密原理

2.2　加密引擎的硬件结构设计

加密引擎的总体结构见图 2。本系统中采用的外部接口为 Logiccore 32 1 bit/66 MHz PCI 总线。

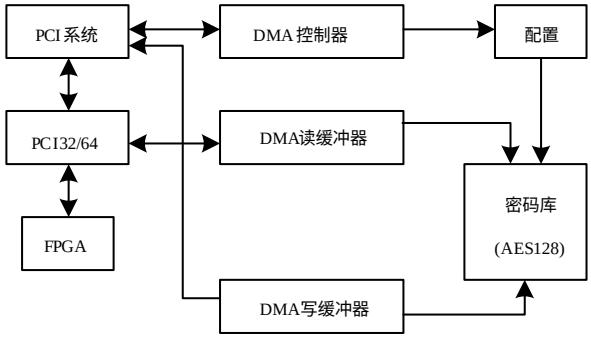

图 2　加密引擎总体结构

加密引擎的基本流程如下：

(1)通过加密引擎中配置控制器，选择调用密码库中系统所适用的 AES 算法，将所有加密密钥的信息和需要加密的明文信息都存储在 PCI 系统的存储器中，并通过 PCI32/64 逻辑核接口将其传输至 FPGA 中。

(2)DMA 读写缓冲利用块 RAM 来实现系统中 PCI 碎发长读写传输，同时由 FPGA 来实现加密数据(密文)的校验，计算和缓冲等操作。

(3)通过加密核心部分来连接 DMA 缓冲器间的主要数据通道。而系统软件主要负责实现相应控制数据结构的创建，这一结构将详细描述分段的位置、数量、大小及加密的类型。所以，可以将写入 PCI 系统存储器中的明文数据流分段成为多个数据包的形式。这样有利于明文数据从 PCI 系统存储器快速传输到 DMA 控制器中的 RAM 单元中。

(4)在加密引擎读取到所有明文片段的同时，配置控制器利用所选择的加密算法(AES 算法)对它们进行加密，然后再在 PCI 系统存储器中重新组装密文片断。在将密文写回 PCI 系统存储器的过程中，FPGA 为每一加密后的片断计算一个校验和，这一数值加到所有片断的总校验和中。这些校验和存储在 FPGA 中的分布式 RAM 区中，在每次传输结束后再由 DMA 控制器读回。

2.3　加密引擎核心部件的实现

对于加密引擎核心部件 FPGA 的实现，采用硬件描述语言进行设计，用 VHDL 语言编写了程序，然后在 quartus5.1 平台上完成了调试编译和模拟工作。在功能测试中，按照先子模块测试、后整体测试的步骤进行。整个加密引擎的代码设计主要包括 3 个模块：加密输入控制模块，128 bit 分组加密模块，加密输出控制模块。按自上至下(Top-Down)的方法进行设计，顶层设计 VHDL 源程序的主要代码如下：

```
LIBRARY IEEE;
//初始化部分
luncntout: OUT STD_LOGIC_VECTOR(0 TO 3));
//加密文本输出轮次计数
END;
ARCHITECTURE behave OF aesnew2 IS
SIGNAL cytout_text :STD_LOGIC_VECTOR(0 TO 127);
SIGNAL luncnt:STD_LOGIC_VECTOR(0 TO 3);
……
BEGIN
cyp:PROCESS(clk, reset)
VARIABLE surtext1: STD_LOGIC_VECTOR(0 TO 31);
//分组的文本
……
VARIABLE cytdata1 :STD_LOGIC_VECTOR(0 TO 31);
//轮加密密码
……
VARIABLE cyttext1 :STD_LOGIC_VECTOR(0 TO 31);
//轮加密后文本
……
BEGIN
surtext1:= source_text (0 TO 31);
//文本分组
……
          IF reset='0' THEN //对信号和变量进行初始化
              luncnt<="0000";
              intersw<='0';
              ……
        ELSE
FOR i IN 1 TO 10 LOOP //10 轮加密
cyttext1:= surtext1;
……
 FOR j IN 1 TO 4   LOOP //每轮加密
rotdata(da,db,dc,dd); //位置变换
……
subdata(dsa,dsb,dsc,dsd); //S 盒变换
……
rconz(xi); //行变换
……
mixcoluz(yi); //列变换
            ……
cyttext1:= cyttext1 NOR cytdata1;//与扩展密匙进行异或运算
……
          IF ena='1' THEN
CASE outcnt IS
WHEN 0 => cryptedout<=cytout_text(0 TO 31);
             ……
  END PROCESS;
END behave;
```

3 加密引擎中核心加密模块中的关键技术

3.1 改善后的 4 级流水线设计

AES算法结构[4]简单，只需要逻辑运算和查找表运算。本文通过优化设计轮函数，使得基本迭代方式下的时钟频率远高于PCI接口的时钟频率 33 MHz。本文在满足算法时钟频率的基础上，采用 4 级轮外流水线，将AES算法的 12 轮迭代过程分为 4 个操作段，每个操作段可作为一级流水线，在操作段内部，每轮之间以反馈(FB)方式完成 4 轮基本迭代，前一个操作段结束后，将结果直接送入第 2 个操作段，同时去处理下一个分组数据，4 个操作段互不影响，并行执行。因此，采用图 3 的 4 级流水线结构可以使加密引擎加密过程更加快速和安全。

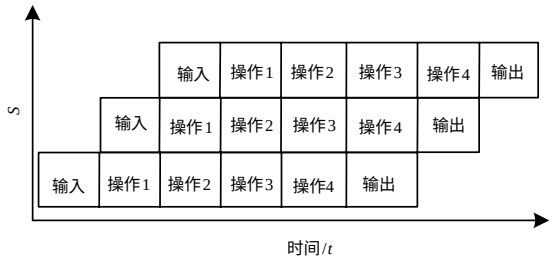

图 3　4 级流水线

3.2 AES 加密引擎的可重构设计

利用 FPGA 内部可编程、可重构的特点，可以将以往分离设计的逻辑电路，利用 FPGA 内部丰富的逻辑资源，以及 VHDL 具有语言门级电路描述强的特点，集成到系统内部。可以根据实际需要选择不同配置信息。由于配置过程时间很短，通常在几百毫秒内，且不需要修改硬件电路，因此可以在系统工作过程中重新对 FPGA 进行配置，实现实时可重构。

3.3 密码库的设计

本文采用 FPGA 与算法的目标体系设计密码库，密码库结构见图 4。

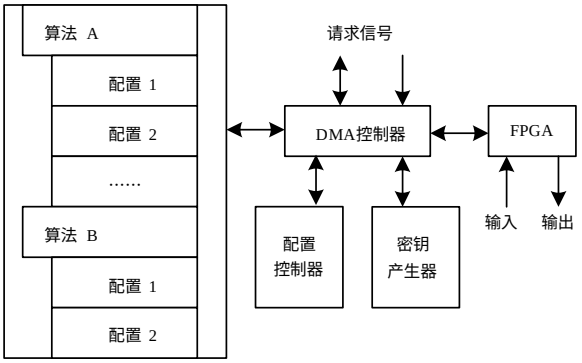

图 4　密码库结构

在加密引擎密码库的设计中，其密码库的数据地址总线直接和 DMA 控制器相联，DMA 控制器为整个协处理器的核心，配置器可以通过外部设置各种参数来达到对加密算法的修订，密钥产生器在 DMA 控制器的配合下，给加密引擎产生密钥。加密引擎的核心是计算部件，其中，集成了 AES,DES,RSA 等最常用的加密算法(本文采用的是 AES 算法)，还有 RND(随机数)、DIG(消息摘要)等基本运算。它们是实现以上对称和非对称加密体系的基础，也是用户级和系统级数据安全的重要保证。通过 FPGA 加密技术，可以获得更好的加密效率。

4 实验仿真与性能比较

本次实验选择 Altera 公司的 APEX2 系列的 EP2A15 B724C7 芯片作为基于 AES 加密引擎的处理器，使用 ISE5.1 做芯片综合和布局布线，ModelSim5.7 做时序仿真，在一块芯片上完成 128 bit 分组的加密和解密。仿真可以满足的较低全局时钟频率为 72.61 MHz，整个系统设计采用 33 MHz 时钟，实验测试结果表明，吞吐量已达到 887 Mb/s。如果提高全局时钟频率，则吞吐量会达到 1.2 Gb/s。

设主密钥为

0102030405060708090a0b0c0d0e0f00(128 bit)

加密明文为

11223344556677889AABBCCDD EEFF00

根据以上数据可得出加密后的密文结果为

56A3D734F68E2531D5BDE56130C4D69E

为了与国外同类产品在主要性能参数进行比较，表 1 中的参数为在时钟频率为 33 MHz下FPGA芯片的处理结果。笔者将本系统所执行结果与基于其他加密候选算法[5-7]的加密性能进行比较。该加密引擎效率测试的参考运行平台为 133 MHz, 256 MB RAM的Pentium处理器。

表 1　高速设计方案性能参数对比

方案	器件	资源/slice	数据流量/(Mb·s^{-1})	速度/资源比
Gai	1P2A15B724C7	774	673	0.66
DES	1P2A15B724C7	863	398	0.55
RSA	1P2A15B724C7	979	280	0.46
本文	1P2A15B724C7	765	710	0.73

从表 1 中可以看出，本文的 AES 加密引擎在成本和速度方面具有较好的优越性。

5 结束语

本文结合嵌入式系统设计的相关经验，设计了一种低成本可重构的 AES 加密引擎，本引擎加快了解密速度，能够支持千兆位数据速率，改进加密引擎的体系结构，缩短加密算法的开发周期，保持较低成本，具有可重构配置灵活性。实验证明了该方法的有效性。

参考文献

[1] Gladman B. A Specification for Rijndael, AES Algorthm[Z]. (2002-10-08). http://www.comms.scitech.susx.ac.uk/fft/crypto/aesspec.pdf.

[2] 王　欣, 马自堂, 徐金甫. 一种 AES 算法的快速硬件实现[J]. 计算机工程, 2005, 31(2): 154-156.

[3] 冯登国. 国内外密码学研究现状及发展趋势[J]. 通信学报, 2002, 23(5): 18-26.

[4] 彭艮鹏, 刘常澍, 李志华. 基于 CPLD/FPGA 的 AES 算法混合流水实现[J]. 电子与信息学报, 2005: 1(2): 155-156.

[5] Dandalis A, Prasanna V K, Rolim J D P. A Comparative Study of Performance of AES Final Candidates Using FPGAs[C]//Proc. of CHES'00. Massachusetts, USA: [s. n.], 2000: 125-140.

[6] Chodowiec P, Gaj K. Very Compact FPGA Implementation of the AES Algorithm[M]. [S. l.]: Springer-Verlag, 2003: 319-333.

[7] Mcloone M, McCanny J V. High Performance Single-chip FPGA Rijndael Algorithm Implementations[C]//Proc. of CHES'01. [S. l.]: Springer-Verlag, 2001: 65-76.