基于 FPGA 的 IPSec 协议安全算法硬件单元设计*)

刘 航 戴冠中 李晖晖 陈赞锋
(西北工业大学信息安全中心 西安 710072)

摘 要 IPSec 协议中的加解密、消息认证等安全算法的硬化实现可以显著改善关键网络设备的安全处理性能。本文采用现场可编程门阵列(FPGA)设计了一个包括 AES、HMAC-SHA-1 等安全算法及其替换算法的 IPSec 协议安全算法硬件单元。仿真结果表明,本文设计的安全算法硬件单元能显著地提高 IPSec 协议的处理速度。

关键词 网际安全协议,现场可编程门阵列,高级加密标准,消息签名,安全散列算法

Design of the Security Algorithm Hardware Unit Based on FPGA for IPSec

LIU Hang DAI Guan-Zhong LI Hui-Hui CHEN Zan-Feng
(Cyberspace Security Center, Northwestern Polytechnical University, Xi'an 710072)

Abstract The hardware-based implementation of the security algorithms, such as block cipher and message authentication, can effectively improve the security processing performance of the key network equipments. In this paper, a security algorithm hardware unit including Advanced Encryption Standard and HMAC-SHA-1 along with their substitute algorithms for Internet Protocol Security (IPSec) is designed based on Field Programmable Gate Array (FPGA). Simulation results show that the security algorithm hardware unit can greatly promote the processing speed of IPSec.

Keywords Internet protocol security, Field programmable gate array, Advanced encrypt standard, Message authentication, Secure hash algorithm

1 引言

IPSec(Internet Protocol Security)协议通过在网络层(IP)提供数据源的验证、无连接的数据完整性、数据机密性、抗重播和有限业务流机密性等安全服务[1,2],使各种应用程序可以享用 IP 层提供的安全服务和密钥管理,而不必设计和实现自己的安全机制,减少密钥协商的开销,降低产生安全漏洞的可能性,能够有效防止网络传输时的信息泄露、篡改等问题,可以显著提高电子商务、电子政务、国防科研、移动办公、企业信息化等网络平台的安全性,实现安全的网络数据传输。因此,IPSec 协议在电子政务网、电子商务网、国防科研网、企业网以及移动办公等领域有着非常广泛的应用前景。

但是,网络带宽的不断提高和 IPSec 协议中加解密、消息认证等各种计算密集型处理任务的增加,使得 IPSec 协议带来了网络设备负载的明显提高和吞吐量的显著下降,导致服务器、网关、路由器和交换机等关键网络设备的处理性能大大降低。因此,迫切需要用硬件处理芯片完成加解密、消息认证等计算密集型任务,为高速、安全的网络信息传输服务提供支持。

本文在分析和研究 IPSec 协议的基础上,采用现场可编程门阵列(FPGA)将 IPSec 协议中的加解密、消息认证等算法以硬件实现,而将处理量小的数据包封装与解封装等协议处理部分用软件实现,从而达到保证网络敏感数据安全传输、显著提高 IPSec 协议的处理速度、降低网络设备负载、提高安全强度的目的。

2 IPSec 协议安全算法单元的实现基础

IPSec 协议的处理任务可以分为两类:计算复杂型任务(如数据包的封装、安全协商、密钥交换、安全算法替换等)和计算密集型任务(如加解密、消息认证计算等)。若计算复杂型任务以硬件方式实现,将影响 IPSec 协议安全体系结构的灵活性,加大安全处理芯片的开发周期,也难以适应用户不同的安全需求和安全参数的变化;而计算密集型任务以软件方式处理时,将影响 CPU 对关键任务的实时性处理,导致网络设备性能的明显下降。

为了保证敏感信息的安全传输,同时适应用户根据应用需求而调整安全算法和安全参数的要求,我们采用将 IPSec 协议中计算复杂型任务和计算密集型任务分离处理的方法,即 IPSec 协议中分组密码算法和消息摘要算法等计算密集型任务通过专用安全处理芯片加以实现,使网络设备的主处理器尽可能不受影响;而对于数据包封装等计算复杂型任务以软件方式加以实现,确保应用处理的灵活性。我们设计了包含 DES、3DES、AES 等对称加密算法和 SHA-1、MD5 等消息认证算法的 IPSec 协议安全算法硬件单元。

3 IPSec 安全算法单元中典型算法的实现

3.1 加解密模块中 AES 算法的设计

AES(Advanced Encryption Standard)[3]算法是继 DES(Data Encryption Standard)和 3DES 之后又一种被 IPSec 协议推荐使用的分组加密算法,它在 Rijndael 算法可变数据分组长度和可变密钥长度的基础上,将数据分组长度固定为

*)基金项目:国防基础研究项目(项目编号:J1300B005)。刘 航 博士生,讲师,主要研究方向为智能信息处理、计算机网络与网络信息安全;戴冠中 教授,博士生导师,主要研究方向为自动控制、信息安全等。

128 位，并且仅支持 128 位、196 位或 256 位长度的密钥。如图 1 所示，AES 算法由 3 部分构成：密钥扩展模块、加密模块和解密模块。密钥扩展模块实现加解密密钥的扩展，导出用于初始密钥加法的一个轮密钥 ExpandedKey[0] 和 N_r 个用于轮变换的密钥 ExpandedKey[i]。加密模块则由一个初始密钥加法 AddRoundKey、N_r-1 次轮变换 Round 和一个尾轮变换 FinalRound 构成。其中，128 位数据分组构成的 4×4 状态矩阵 State 和轮密钥 ExpandedKey[i]组成初始密钥加法和所有轮变换的输入。为了实现的方便，解密模块一般采用与加密模块相同的步骤次序：一个初始密钥加法、N_r-1 次等价轮变换 EqRound 和一个等价尾轮变换 EqFinalRound，只是在 EqRound 和 EqFinalRound 中的每一步被改成 Round 和 FinalRound 中相应步骤的逆，并相应改变密钥选择次序。

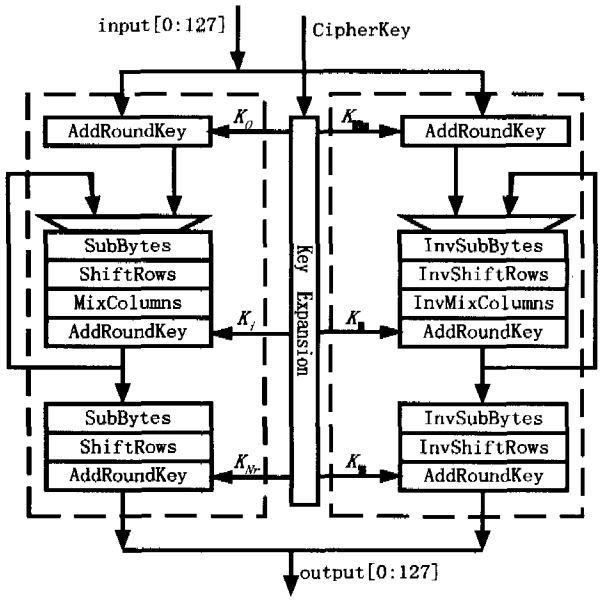

图 1　AES 算法结构图

在 AES 算法的硬件实现中，我们结合算法的特性进行了优化设计：行移位单元仅涉及状态矩阵各字节的移位操作，直接的信号换位既可省去 FPGA 中逻辑单元的使用，又可以缩短电路延迟时间；密钥加法单元仅需要一次逻辑异或运算即可实现。在线实现密钥扩展将使加解密模块经常处于等待轮密钥的状态，降低 AES 算法的实现性能。我们借鉴文[4]的方法，将预先计算好的轮密钥存储于存储器中，从而保证加密和解密模块的处理性能。为了使芯片达到优化的运算处理性能和优化的资源占用率，用 ALTERA 芯片的 ESB(Embeded System Block)分别实现用于加密和解密运算的 S_{RD} 和 S_{RD}^{-1}。在列混合与等价逆列混合的实现中，考虑到 InvMixColumns 变换的系数矩阵由{0x09、0x0B、0x0D、0x0E}组成，硬件上的实现会占用较多的逻辑单元，并造成硬件时延的增加，导致整体性能降低。为了降低系统的资源占用率和电路延迟，我们设计了 1 字节的 MixColumns 和 InvMixColumns 基本复用运算模块，实现状态矩阵每列第 1 个字节的 MixColumns 和 InvMixColumns 运算，在此基础上设计了多个 4 字节的 MixColumns 和 InvMixColumns 单元，完成列混合计算。

优化后的 AES 算法硬件实现结构如图 2 所示。在该实现中，数据加密和解密模块被单独实现，每个加密或解密的轮运算步骤均在 1 个时钟周期内完成。因此，一个 128 位数据分组经过 11 个时钟周期可以完成加密或解密计算。进一步优化还可以使其具有并行处理任意两个数据分组的解密运算和不同数据包中两个数据分组的加密运算的能力，有效提高数据吞吐量。

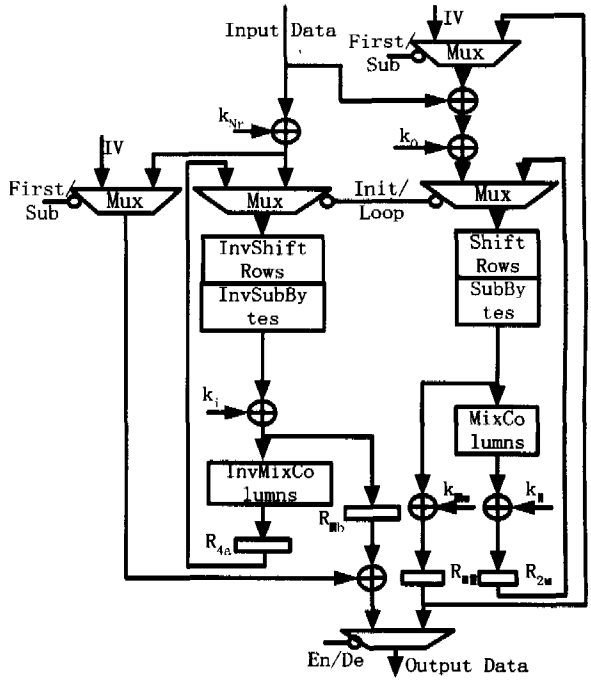

图 2　AES 算法的硬件实现结构图

3.2　消息认证模块中 HMAC-SHA-1 算法的实现

由于 HMAC 包括两轮相同的散列计算，散列算法的逻辑单元占很大比重，因而消息认证模块整体采用图 3 所示的结构，这可以复用散列算法模块，大大减少逻辑单元。图中的 H_SEL 是散列算法的选择信号，状态控制机负责控制两轮散列算法的控制信号输入，首轮输出摘要寄存器用于保存第一轮的摘要输出。

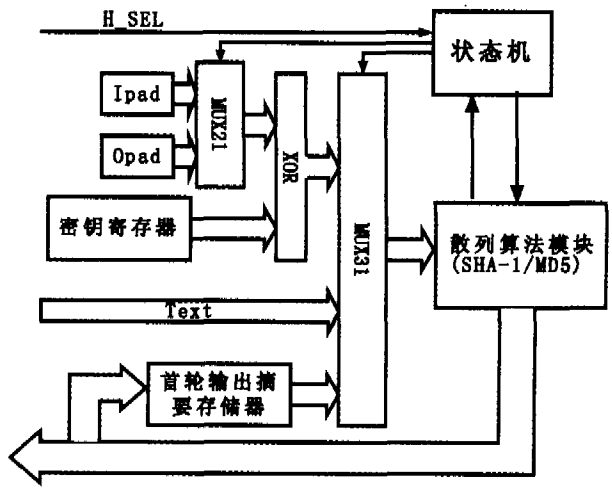

图 3　HMAC-SHA-1/HMAC-MD5 算法实现框架

对于散列算法模块中的 SHA-1 算法，可将其分成输入预处理模块、存储模块、状态机控制模块、核心处理模块等 4 个主要模块。预处理模块完成填充位和填充明文长度的功能；存储模块完成明文的存储；状态机控制模块提供控制信号；核心处理模块根据状态机给出的控制信号，循环完成计算。其中，SHA-1 算法的整体电路图如图 4 所示。存储器模块完成从 M_i 到 W_i 的转化；核心操作模块循环完成 80 步的操作。

对任何一个明文分组，核心操作模块都有 82 个状态(包括一个预处理、80 个循环操作、一个与初始摘要相加)。

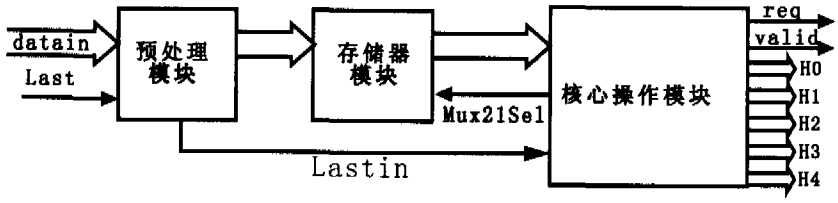

图 4　SHA-1 实现结构图

由于 W_i 是 W_{i-3}，W_{i-8}，W_{i-14}，W_{i-16} 的异或、移位运算，因而加快 M_i 到 W_i 转换是提高速度的关键。为了提高存储器模块的速度，采用图 5 所示的存储器结构，分别从 32×16 的移位寄存器矩阵 $Wmatrix_{j\times i}$ 引出 W_{i-3}，W_{i-8}，W_{i-14}，W_{i-16}，再经过异或移位计算将输出反馈回输入端。当 $0\leqslant i\leqslant 15$ 时，输入为 M_i；当 $16\leqslant i\leqslant 79$ 时，输入为 W_i。

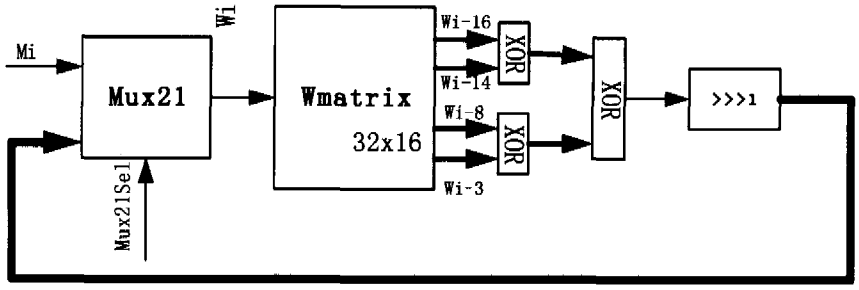

图 5　SHA-1 存储器模块结构图

对核心操作树进行优化，使 T0 周期完成 $f(0,B,C,D)$[4] 和 $SumEk_0=E_0+K_0$，从 T1 到 T80 周期完成($A_i<<<5$)，$f_i(B,C,D)+W_i+SumEK_i$，T81 周期完成与初始摘要的相加。这样，每一步的最大操作延时减少至两个加法，提高了最大时钟频率 f_{max}。

4　仿真实验结果

为了测试本文设计的 IPSec 协议安全算法硬件单元的性能，我们采用 ALTERA EP20KE200EFC484-2x 作为实验载体进行了综合仿真验证。其中，AES 加密模块的最大仿真时钟频率为 $f_{max}=59.75$MHz，占用 749(9%)个逻辑单元、17 个 ESB(32%)，理论处理速度可达 128×59.75÷11=695Mbps；AES 解密模块的最大仿真时钟频率为 $f_{max}=49.53$MHz，占用 1183(14%)个逻辑单元、21 个 ESB(40%)，理论处理速度可达 128×52.6÷11=576Mbps；HMAC-SHA-1 模块的最大仿真时钟频率 f_{max} 达到 $f_{max}=62.49$MHz，占用 1624(19%)个逻辑单元、32 个(61.5%)ESB，理论处理速度可以达到 512×62.49÷82=390Mbps。此外，我们还搭建了一个用于验证各个模块的硬件平台，由 FPGA 内部产生测试激励，分别以接近最大仿真频率(易获得)的频率对本文涉及的 AES、SHA-1 算法和 IPSec 协议标准规定的 DES、3-DES、HMAC-MD 等安全算法硬件单元进行了实物验证，获得了与仿真相一致的结果。

结束语　为了解决引入 IPSec 协议后，网络数据传输中安全与速度之间的矛盾，本文采用现场可编程门阵列(FPGA)设计和实现了 IPSec 协议中加解密、消息认证等计算密集型单元，并在算法分析的基础上，对设计进行了优化。仿真结果表明，本文设计的 IPSec 安全算法硬件单元能显著提高协议处理速度，改善关键网络设备的安全处理性能和实时性，能够满足 100M 网络的传输要求。但是，针对日益提高的网络速度，本文所设计的 IPSec 协议安全算法硬件单元各模块的性能还有待进一步从体系结构、算法、协议、芯片工艺等方面加以优化和改进，使之更为合理、高效。

参 考 文 献

1　Davis C R. IPSec —— VPN 的安全实施. 周永彬，等译. 北京：清华大学出版社，2002

2　IP Security Protocol(ipsec) Charter - Lastest RFCs and Internet Drafts for IPSec. http://ietf.org/html.charters/ipsec-charter.html

3　Daemen J, Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag Berlin Heidelberg, 2002

4　Chodowiec P, Gaj K, Bellows P, et al. Experimental Testing of Gigabit IPSec-Compliant Implementation of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board. In: Proceedings of the 4th International Information Security Conference, 2001. 220～234

5　US Secure Hash Algorithm 1 (SHA-1). www.faqs.org/rfcs/rfc3174.html