一种基于 strongSwan 的 IPSec VPN 网关的实现

蒋 华[1,2] 李康康[2] 胡荣磊[1]

[1](北京电子科技学院 北京 100070)

[2](西安电子科技大学 陕西 西安 710071)

摘 要 IPSec VPN 是一种使用 IPSec 协议来实现的虚拟专用网技术。针对国密算法在网络安全产品上的应用相对较少这一问题,设计一种基于开源 IPSec 项目 strongSwan 的 VPN 网关。该网关使用 SSX0912 加密芯片中的国密算法接口替换了 strongSwan 的国际密码算法接口,完成了 strongSwan 对国密标准的支持。将修改后的 strongSwan 移植到 AM335x 为核心的开发板中,在嵌入式硬件环境中实现了 IPSec VPN 的网关。通过搭建开发环境测试,该网关运行稳定,延时小,使用硬件加密模块,安全性更高,相比于简单的 Linux 系统实现,应用范围也更加广泛。

关键词 虚拟专用网 国密算法 strongSwan 嵌入式

中图分类号 TP393.08 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2017.07.016

AN IMPLEMENTATION OF IPSEC VPN GATEWAY BASED ON STRONGSWAN

Jiang Hua[1,2] Li Kangkang[2] Hu Ronglei[1]

[1](*Beijing Electronics Science and Technology Institute, Beijing 100070, China*)

[2](*Xidian University, Xi'an 710071, Shaanxi, China*)

Abstract IPSec VPN is a technology using IPSec protocol to implement virtual private network. Aiming at the problem that the application of secret algorithm in network security product is relatively small, a VPN gateway based on the open source IPSec project strongSwan is designed. The gateway replaces strongSwan's encryption algorithm interface with the SSX0912 encryption chip, which implements the strongSwan support for the state security standard. The modified strongSwan is transplanted to AM335x as the core of the development board in the embedded hardware environment to achieve the IPSec VPN gateway. By setting up the development environment, it is found that the gateway runs stably and has little delay. Compared with the simple Linux system, the hardware encryption module is more secure and the application range is more extensive.

Keywords Virtual private network State secret algorithm StrongSwan Embedded

0 引 言

随着 IPSec VPN 对性能、安全性的要求越来越高,IKEv1(网络密钥交换)协议[1]的冗余性、缺乏一致性的缺点越来越明显[2]。因此,在 2005 年 10 月,IETF 工作组发布了 IKE 协议的第二版,即 IKEv2[3]。IKEv2 简化了 IKEv1 的复杂功能,增强了安全性,具体体现在抵御中间人攻击、拒绝服务攻击、完美前向保护等几个方面。IPSec 协议是目前 VPN 技术开发中使用最广泛的一种安全协议[4]。Linux 上常用的 IPSec 协议实现有 Frees/wan 项目,Frees/wan 分为 Openswan 和 strongSwan。Openswan 是 Linux 下的开源项目,由于软件结构相对简单,安全性较好,兼容效果好,已经集成于 Cisco 等多家厂商的 VPN 产品中[5]。但是 Openswan 也有局限性,那就是只支持 IKEv1 协议,而 IKEv1 协议的冗余性等缺点对它今后的发展前景带来了很大的限制。IKEv2 同时兼顾高效性、安全性的特点将会越来

收稿日期:2016-07-20。中央高校基本科研业务费项目(328201502)。**蒋华**,教授,主研领域:通信与信息安全。**李康康**,硕士生。**胡荣磊**,副研究员。

越受到人们的青睐。strongSwan 同时支持 IKEv1 和 IKEv2 协议,可以广泛应用于不同的终端,包括 PC、安卓手机、IOS 手机等。

VPN 的数据加解密技术、隧道技术、密钥管理技术、用户身份认证技术已经广泛应用于保障互联网信息安全的方方面面。数据加解密技术中使用的密码算法全部是来自于国外组织或机构制定的标准,常用的有 DES、3DES、AES、SHA-1、ECC 等。为了适应我国自身的安全需求,我国国家密码管理局批准了一系列国密标准的密码算法,例如 SM1、SM2、SM3、SM4、祖冲之算法等。在此基础上也制订了基于国密算法的 VPN 技术规范[5]。因此,基于国密的 IPSec 协议标准,实现自主可控的 IPSec VPN 网关是国家安全和经济发展的需要。

本文通过替换 strongSwan 中的加密算法,添加对国密 SM3、SM4 算法的支持,并且将 strongSwan 移植到嵌入式平台下,搭建 IPSec VPN 开发环境,最后经过测试,分析了该网关的性能。

1 网关总体结构

网关结构如图 1 所示,包括加密模块和 AM3-35x 为核心的开发板。

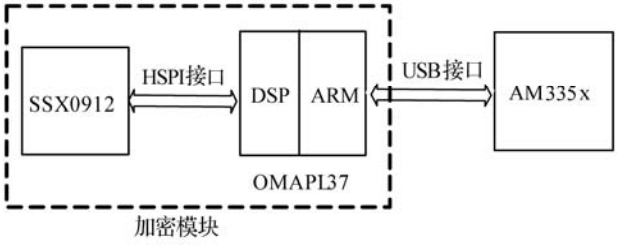

图 1 网关总体结构图

1.1 硬件加密模块设计

硬件加密模块是 VPN 网关实现的基础。传统的软件加密系统具有开发时间短、研发成本低等优点,但是硬件加密在抵御攻击、密钥的安全存储、运算速度等方面有更大的优势。所以加密模块的设计往往决定着 VPN 的性能的好坏。

本文通过美国德州仪器推出的 OMAPL137[6]双核心芯片设计了加密模块,芯片包含 ARM 和 DSP 两部分。其中,ARM 端负责通信机制的控制,而 DSP 端结合 SSX0912 安全加密芯片对密码算法进行处理。ARM 和 DSP 端通信是基于 Ti 公司的 DSPLINK 技术。加密模块在功能上作为一个可接入多个通信终端的设备,并且加密模块的安全服务独立于通信系统,加密模块开发完成以后只向外界提供一些可用的 API 接口,外界通信终端只要调用 API 接口就能实现加解密等安全服务功能。

加密模块的设计采用了 SSX0912 安全加密芯片。SSX0912 安全加密芯片可以实现 SM2、SM3、SM4 等国标密码算法,同时也带有 UART 总线接口、RS232 总线接口、HSPI 总线接口和 USB2.0 接口。用户可以根据应用选择不同的接口。

针对安全应用,SSX0912 安全加密芯片参照了 X.509V3的证书格式,自定义了证书结构,在模块中加入了证书和 CA 公钥存储,方便密钥协商和身份认证。

由于 OMAPL137 要通过 USB 和外界进行通信,要接收外界发过来的数据,所以软件方面驱动的编译是关键。安全模块相比于 AM335x 来说是一个从设备,这就要涉及到 USB gadget 的驱动开发。因为直接编写驱动程序会很困难,所以通过查阅资料,本设计决定修改 Linux 设备驱动程序 gadget_serial.c 来完成驱动程序。

通过对 Linux 设备驱动程序结构的分析,Linux 虚拟串口设备驱动程序的程序框架和其他设备驱动的程序框架是一样的。所不同的是在设备驱动程序里添加一些支持字符处理的接口函数,并且在 USB 虚拟串口中添加关于串口通信的驱动模块。值得一提的是,读写速度有多大,串口不能控制,也就是说 USB 给出多大的速度,出口就能读写多大的速度。所以,虚拟串口的速度不受串口波特率的影响,只受 USB 协议读写速度的影响。

将修改好的程序加载到 Linux 内核,然后交叉编译内核,生成 g_serial.ko。接着通过制作 Ramdisk 根文件系统把 g_serial.ko 文件拷贝进去,烧写文件系统,由超级终端重启开发板 OMAPL137,使用命令 insmodg_serial.ko 加载驱动。通过 cat /proc/devices 可以查看到已加载驱动的设备号,最后用 mknod /dev/ttyusb c 127 0 创建设备节点。为了防止每次启动开发板系统都要重新加载驱动,可以在/etc/init.d/rcS 程序自启动文件中输入以上命令,这样每次重启驱动就自动加载了。驱动加载完成之后就可以通过 USB OTG 接口和外界终端设备通信了。

1.2 AM335x 驱动设计

根据图 1 所示网关结构,strongSwan 开源程序在经过修改后,交叉编译到 AM335x 中,AM335x 和加密模块通过 USB 接口进行通信,此时的 AM335x 相对于 OMAPL137 来说是主设备[7]。因为嵌入式系统内核是裁剪过的,许多设备驱动并没有,所以我们需要为主设备 AM335x 添加 CDC-ACM 驱动,CDC-ACM 驱动允许

任何通信设备去提供一个串口通信接口。具体步骤:

1) 在 AM335x 的 Linux 内核中找到 CDC-ACM 支持的选项,如图 2 所示。

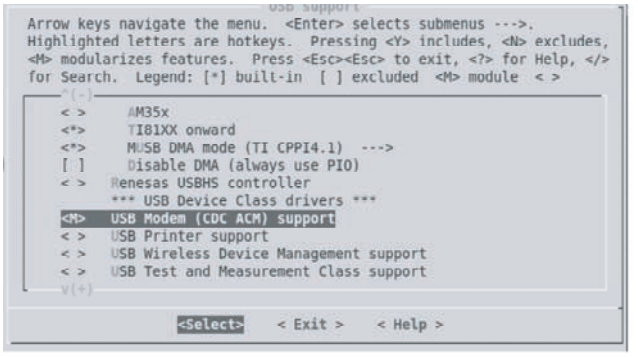

图 2　CDC-ACM 内核选项

2) 选中 module 之后,使用 AM335x 的交叉编译工具编译内核,在./drivers/usb/class 目录下可以找到 cdc-acm.ko。

3) 将 cdc-acm.ko 移到 AM335x 中,insmod cdc-acm.ko,在插上 USB 数据线时会动态产生驱动 ttyACM0。

2　网关软件设计

2.1　软件平台

VPN 网关的操作系统是经过裁剪的实时 Linux 系统,通过对内核的裁剪和修改,去除了许多不必要的模块,保留了一些必要的驱动支持。操作系统由通用的 Linux 系统变为实时的操作系统,节省了空间,提升了便利性,使系统性能得到了很大的提高[8]。

2.2　strongSwan 简介

strongSwan 是一个完整的 IPSec 实现,支持的 Linux 内核为 2.6、3.x 和 4.x。strongSwan 的重点是其强大的身份认证机制,它还有很多优点,例如支持证书撤消列表和在线证书状态协议(OCSP),完全支持 IPv6 IPSec 隧道和传输连接,支持 ESP(封装安全载荷)[9] 单独使用时的 NAT 穿越,完美的 PFS(前向保护性)等。和 Openswan 仅支持 IKEv1 协议相比,strongSwan 同时支持 IKEv1 和 IKEv2 协议。

strongSwan 中的 IKEv2 协议的消息协商对 IKEv1 进行了很大的改进。IKEv1 协商过程非常复杂,分为两个阶段,四种模式。而 IKEv2 取消了模式的概念,消息交换包含三个基本交换类型:初始交换(Initial 交换)、协商子 SA 交换(CREATE_ CHI-LD_SA 交换)、信息交换(INFORMATION Exchang-e)[2-3]。

strongSwan 使用 daemon(守护进程)来控制 IKE 过程,如图 3 所示。不同的 IKE 版本 daemon 不同,IKEv1 协议使用 pluto,而 IKEv2 协议使用 charon。

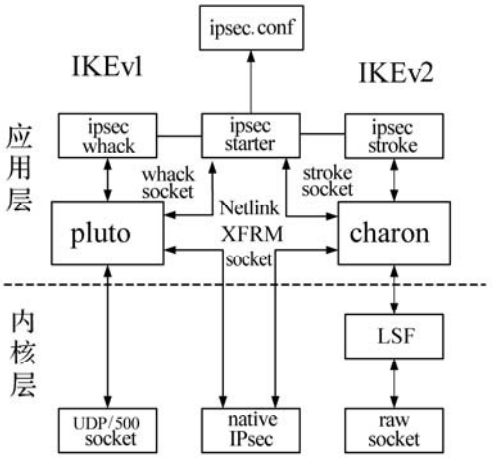

图 3　strongSwan IKE 守护进程

2.3　strongSwan 密码算法的替换

SSX0912 加密芯片提供了 SM3、SM4 等国密算法的接口,外界通信终端只要调用这些接口就能实现加解密等安全服务功能。

通过分析 strongSwan 源码的结构,其中有许多插件(plugin),加载密码算法的插件在/src/libstrongswan/目录中,libstrongswan 结构如图 4。在所加载的插件中,对称加密算法插件有 aes、des 等,完整性验证所用到的摘要算法有 sha1、sha2、md4、md5 等,除了这些密码算法外,还有 openssl 库、sqlite 库等。其中一些不常用的算法会使用 openssl 中的函数来实现,strongSwan 数字签名算法用的就是 openssl 函数库中的 ECDSA,即椭圆曲线数字签名算法。

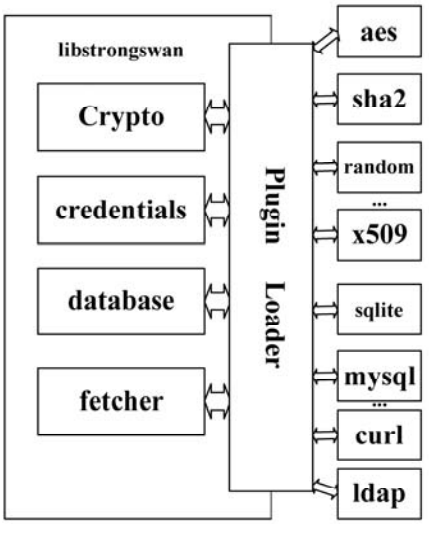

图 4　基于 plugin 的 libstrongswan 模块结构

SM3、SM4 算法的替换只是在原有算法的基础上更改了函数的接口,并且修改了算法的命名机制,将原有的“aes-128”和“sha-256”分别替换为“SM4”和“SM3”。算法替换完成后需要证明是否替换正确。

2.3.1 SM4 算法的替换

strongSwan 启动时默认加载的是 aes-128 算法,即密钥长度和数据长度均为 16 字节,这和 SM4 算法相对应,所以只需修改 aes_crypter. c 程序即可。通过研读 aes_crypter. c,主要函数有:encrypt( ) 和 decrypt( )。Encypt( ) 函数是加密函数,decrypt( ) 是解密函数,它们都包含 encrypt_block( ) 函数,就是用来加密一块数据的函数。SM4 程序中对应的函数为 sm4_crypt_ecb( ) 和 sm4_crypt_cbc( ),这两个函数均为 SSX0912 中的算法接口,可以使用 ioctl 的方式调用这两个接口。因为 SM4 算法解密是加密的逆过程,所以只需将加密标志 SM4_ENRYPT 换为解密标志 SM4_DECRYPT 即可,具体代码不再赘述。

2.3.2 SM3 算法的替换

strongSwan 默认加载的杂凑算法为 sha-1,sha-1 输出的摘要值长度为96 位或者 160 位,而 SM3 算法输出摘要值为 256 位。又因为 sha-256 算法的输出摘要值长度为 256 位,这样就能够和 SM3 算法相对应起来了。由于 strongSwan 结构的特殊性,提供 sha-256 函数源码的插件有两个:一个是 plugin 文件夹中的 sha2 插件,这里面包含 sha-256、sha-384、sha-512 三个杂凑函数;另一个是 openssl 插件,它是调用 openssl 库中的开源函数来实现杂凑函数的。但是在 IKE 和 ESP 过程中,默认加载的并不是 sha2 插件中的 sha-256 函数,而是 openssl 函数库中的 sha-256 函数。通过屏蔽 openssl 中调用 sha-256 函数的代码,strongSwan 就开始调用 sha2 插件中的 sha-256 函数了,这时候只要修改 sha2 插件中的 sha2_hasher. c 程序即可。

在 sha2_hasher. c 程序中,有两个函数:get_hash-256( )、allocate_hash256( ),主要功能分别是获取摘要值和分配哈希函数空间。它们都包含两个主要的接口:sha256_write( ) 和 sha256_final( ),将它们替换成 SM3 算法的接口,即 sm3_update( )、sm3_finish( ),它们均为 SSX0912 加密芯片的接口。同理,al-locate_hash256( ) 函数替换方法与之相同。

2.4 移植 strongSwan

在一种计算机环境中运行的编译程序,能编译出在另外一种环境下运行的代码,我们就称这种编译器支持交叉编译。简单地说,就是在一个平台上生成另一个平台上的可执行代码,而这种工具就是交叉编译器。

strongSwan 要想在 AM335x 上成功运行,这就需要进行交叉编译。移植之前,已经在 Ubuntu Linux 中安装好了 AM335x 的交叉编译工具。

2.4.1 strongSwan 驱动编译

strongSwan 运行时需要许多必要的驱动,这些驱动在 AM335x 的内核中并没有被加载,需要手动添加内核选项并且重新编译内核,根据 strongSwan 官网提供的内核编译选项进行添加,如图 5 所示。

Required Kernel Modules

Include the following modules:

```
Networking  --->
 Networking options  --->
   Transformation user configuration interface [CONFIG_XFRM_USER]
   PF_KEY sockets [CONFIG_NET_KEY]
   TCP/IP networking [CONFIG_INET]
     IP: advanced router [CONFIG_IP_ADVANCED_ROUTER]
     IP: policy routing [CONFIG_IP_MULTIPLE_TABLES]
     IP: AH transformation [CONFIG_INET_AH]
     IP: ESP transformation [CONFIG_INET_ESP]
     IP: IPComp transformation [CONFIG_INET_IPCOMP]
     IP: IPsec transport mode [CONFIG_INET_XFRM_MODE_TRANSPORT]
     IP: IPsec tunnel mode [CONFIG_INET_XFRM_MODE_TUNNEL]
     IP: IPsec BEET mode [CONFIG_INET_XFRM_MODE_BEET]
```

图 5 strongSwan 需要的内核驱动选项

按照图 5 提供的内核选项,可以看出 strongSwan 需要加载 AH 传输、ESP 传输、IPSec 传输和隧道模式等必要的驱动。将这些驱动直接加载进内核中,交叉编译 AM335x 开发板内核,在./arm/boot 文件夹中生成了新的内核 zImage,根据开发板提供的烧写方法,重新烧写内核即可。

2.4.2 交叉编译 strongSwan

通过阅读 strongSwan 在 Ubuntu Linux 上成功运行的日志,可以看到 strongSwan 需要加载一些插件。而有些插件例如 openssl 库、gmp 库、sqilte 库在嵌入式环境下是没有的,这时候就需要将这些必要的库移植进开发板。

交叉编译过程中需要注意的几点:

1) 从源码安装开始,也就是使用命令./configure,后面需要许多选项,包括不同的插件,交叉编译工具的路径,生成文件的路径,必须要链接的一些库。

2) 使用 make 命令编译安装:make&&make install。

3) 编译生成的 strongSwan 文件夹如果直接通过 tftp 协议传到开发板中,这样会比较麻烦,重新制作 AM335x 的文件系统再进行烧写会比较简单。

3 网关性能测试与分析

3.1 测试环境的搭建

在将 strongSwan 移植进 AM335x 开发板之后,搭建如图 6 所示站点到站点的 VPN。

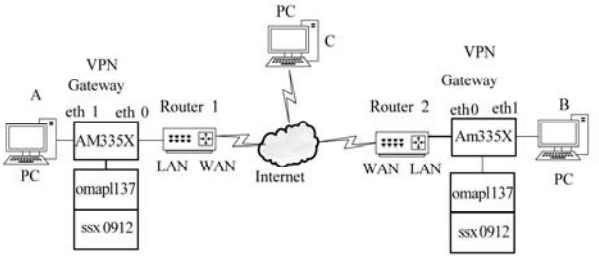

图 6　测试环境

VPN 网关是网关型安全设备,它可以访问公网,而不改变原配置局域网和路由器之间的网络结构[10]。AM335x 开发板有两个网口,同时支持 4G 和 WiFi,测试环境就选在 WiFi 中实现。如图 6 所示,终端 PC A 和 PC B 分别连接 AM335x 的 eth1 网口,两个 eth0 网口分别通过一个 Router 连接公网 Internet。

Router1 WAN 口 IP 地址:192.168.227.121,Router2 WAN 口 IP 地址:192.168.227.122,PC C 的 IP 地址:192.168.227.123。设置其他网卡的 IP 地址,其中 PC A 的 IP 地址:192.168.30.101,PC B 的 IP 地址:192.168.31.101,与 A 连接的 eth1 网口 IP 地址:192.168.30.121,eth0 网口 IP 地址:192.168.1.2,与 B 连接的 eth1 网口 IP 地址:192.168.31.121,eth0 网口 IP 地址:192.168.2.2。

3.2　IKE 协商过程测试

strongSwan 成功移植进 AM335x 开发板后,使用 IPSec pki 命令生成证书来验证发起方和响应方的身份,经过对 IPSec.conf、IPSec.secrets 文件的配置,紧接着使用命令 IPSec start-nofork 分别启动两端的 strongSwan,成功地产生了隧道“192.168. 30.121/24 === 192.168.31.121/24”,此时网关后面的主机 A 和 B 可以互相访问。

通过 wireshark 在 PC C 使用混杂模式进行抓包,可以捕捉到 IKE 三种消息交换的数据包。

IKEv2 的初始交换分为了两对消息[2],其中第一对消息的 IKE_SA_INIT 过程协商了加密与杂凑算法,交换了 nonce 值,完成了 Diffie-Hellman 密钥协商。最后计算出了后面阶段需要的各种密钥值。在输出日志上可以看到协商出的加密算法为 SM4,杂凑算法为 SM3,这就是之前替换过的算法。

3.3　ESP 过程测试

根据图 6 搭建的测试环境,在 PC A 上 cmd 命令行 ping 192.168.31.101,也就是 PC B 的 IP 地址,再通过 wireshark 在 PC C 上抓包,使用混杂模式,就可以捕捉到了 ESP 包。

因为 wireshark 上抓到的 ESP 包是加密后的包,无法进行解密,通过设置 strongSwan 的日志等级为 4 级。查看一些敏感信息,就可以获得加密和 HMAC 所需的密钥,然后就可以进行 SM4 和 SM3 算法的验证了。

通过阅读日志得到的数据包和加密密钥,使用国密 SM4 算法软件进行加密,和 wireshark 抓到的加密包进行比较,结果完全相同,证明算法已经替换为了 SM4 国密算法。SM3 算法经过验证,也是正确的。

经过对结果分析,SM3 和 SM4 算法替换成功,已经成功在 strongSwan 中添加了国密标准。

3.4　网关性能分析

首先测试和比较了在开发环境下,有 VPN 隧道和没有 VPN 隧道时 TCP 和 UDP 传输的性能。首要的性能指数是 Round-Trip Time(往返时延)[11],单位是 ms。TCP 一个特殊的性能指数是 TCP 吞吐量。UDP 三个特殊的性能指标有 UDP 吞吐量、网络抖动时间、丢包率[11]。吞吐量单位是 Mbit/s,抖动时间单位是 ms。使用命令 ping 来测试往返时延,发送 100 次,结果表明,没有 VPN 隧道和有 VPN 隧道时平均往返时间分别为 1 和 2 ms。

在 VPN 网关上使用命令 iperf 测试 30M 带宽下,TCP 在有 VPN 隧道和没有 VPN 隧道时的吞吐量。分别设置发送数据大小为 8、128、256、512 KB。结果如表 1 所示。

表 1　不同数据下 TCP 吞吐量　　Mbit · S^{-1}

数据 / 吞吐量	8 KB	128 KB	256 KB	512 KB
无 VPN 隧道	21.71	27.58	27.88	27.76
有 VPN 隧道	16.65	19.23	19.40	19.25

由表 1 可知,我们可知在没有 VPN 隧道的情况下,使用命令 iperf 读取 256 KB 数据时,TCP 吞吐量最大,接近为 28 Mbit/s,有 VPN 隧道时,TCP 吞吐量为 19.4 Mbit/s,减少了接近 31%。

分别设置测试环境中带宽为 1、10、30、50 M,使用命令 iperf 来测试 UDP 传输的性能。其中 UDP 吞吐量如表 2 所示。

表 2　不同带宽下 UDP 吞吐量　　Mbit · S^{-1}

带宽 / 吞吐量	1 M	10 M	30 M	50 M
无 VPN 隧道	1.00	9.80	29.90	50.00
有 VPN 隧道	0.99	8.50	28.40	42.20

结果表示,当带宽设置为 50 M,没有 VPN 隧道产生时,UDP 吞吐量为 29.8 Mbit/s,有 VPN 隧道时,UDP

吞吐量为 20.4 Mbit/s,减少量超过 31%,而在小于 30 M带宽时,UDP 吞吐量变化相对较小。

影响 VPN 隧道产生时 TCP 和 UDP 吞吐量的主要因素可能有以下几点:

1) 加密模块加密程序是基于多线程设计的,每次使用都需要打开和关闭加密模块一次,这个步骤会使实际速度不能达到理论的峰值。

2) 根据表 2 结果分析,吞吐量的变化量和带宽有关系,当带宽为 30 M 时,吞吐量变化最小。

3) strongSwan 程序在运行时也会消耗一定的时间,这样也会影响速率。

此外,用命令 iperf 测出的 UDP 抖动时间和 UDP 丢包率分别如表 3、表 4 所示。

表 3 不同带宽下 UDP 抖动时间 ms

带宽 / 吞吐量	1 M	10 M	30 M	50 M
无 VPN 隧道	0.019	0.013	0.008	0.001
有 VPN 隧道	0.013	0.012	0.007	0.001

表 4 不同带宽下 UDP 丢包率 %

带宽 / 吞吐量	1 M	10 M	30 M	50 M
无 VPN 隧道	0.00	0.00	0.01	0.01
有 VPN 隧道	0.00	0.17	0.15	0.12

分析表 3、表 4 结果可知,在 VPN 隧道产生的情况下,UDP 抖动时间在减少,虽然丢包率相比无 VPN 隧道时有了增加,总体来说,网关运行很稳定。

4 结 语

本文在嵌入式平台下实现 VPN 网关,对系统内核进行了裁剪,去除了许多不必要的模块,实时性相比于简单的 Linux 上实现有较大的提高。使用开源软件 strongSwan 实现 IPSec VPN,替换了密码算法,支持了国密标准,可以满足国家安全与经济发展的需求。srongSwan 精简了 Openswan 中 IKE 协商 SA 的过程,增强了安全性,提高了使用的便利性。采用硬件加密模块,网关数据加解密、完整性验证过程都由加密模块来完成,实现了端到端的加密,相比于通信链路加密,进一步增强了安全性。根据文献[11]给出的方法测试,网关运行稳定,延时小,基本符合应用需求。在 strongSwan 算法库和 Linux 内核中注册国密算法[12]而不是替换国密算法成为下一步工作的重点。

参 考 文 献

[1] Harkins D,Carrel D. RFC2409:The Internet Key Exch-ange (IKE) [S]. 1998.

[2] 刘骥宇. IKEv2 协议在 Linux 环境下的实现[D]. 河南大学,2007.

[3] Kaufman C. RFC4306:Internet Key Exchange (IKEv2) Protocol [S]. 2005.

[4] 王凤领. 基于 IPSec 的 VPN 技术的应用研究[J]. 计算机技术与发展,2012,22(9):250 - 253.

[5] 郑艺斌. 基于国密标准的 IPSec VPN 服务器设计与实现[D]. 西安电子科技大学,2014.

[6] TI. OMAP-L137 C6000 DSP + ARM Processor Technical Reference Manual[Z]. TI;2013.

[7] 李成龙. 基于 USB 通信的嵌入式主从机系统设计与应用[D]. 中南大学,2013.

[8] 杨黎斌,慕德俊. 基于硬件加密的嵌入式 VPN 网关实现[J]. 计算机工程与应用,2007,43(4):122 - 124.

[9] KentS. RFC4303:IP Encapsulating Security Payload(ESP) [S]. IETF. 2005.

[10] Fei C, Wu K, Wei C, et al. The Research and Implementation of the VPN Gateway Based on SSL[C]// Fifth International Conference on Computational and Information Sciences. IEEE, 2013:1376 - 1379.

[11] Du Meng. Implementation of a Host-to-Host VPN based on UDP tunnel and Open VPN Tap Interface in Java and its performance Analysis[C]. International Conference on Computer Science & Education. 2013:940 - 943.

[12] 邓旻昊,汪海航. 为 IPSec 添加新对称加密算法[J]. 计算机安全,2008,28(1):25 - 27.

(上接第 73 页)

[13] Ioannidis Y E, Wong E. Query optimization by simulated annealing[C]// Association for Computing Machinery Special Interest Group on Management of Data Conference. ACM, 1987:9 - 22.

[14] 曹阳, 方强, 王国仁, 等. 基于遗传算法的多连接表达式并行查询优化[J]. 软件学报, 2002, 13(2):250 - 257.

[15] 林桂亚. 基于粒子群算法的数据库查询优化[J]. 计算机应用研究, 2012, 29(3): 974 - 975.

[16] Hammoud M, Rabbou D A, Nouri R, et al. DREAM: Distributed RDF engine with adaptive query planner and minimal communication[J]. Proceedings of the VLDB Endowment, 2015, 8(6): 654 - 665.

[17] 徐秉堃. 解多目标优化问题的改进加权求和算法[D]. 西安电子科技大学, 2010.