商密算法数据包解析工具的设计与实现

蒋 华 樊金坡 张 罡 胡荣磊
(北京电子科技学院 北京 100070)

摘 要 在网络安全产品密码算法国产化的背景下,商用密码算法在电子政务、金融、交通等诸多领域的应用日益增多。但支持商密算法加密的数据包解析的软件相对较少,对设备安全性、加解密功能正确性、商密数据载荷分析等领域还存在空白。针对该问题,设计一种基于 Wireshark 的商密算法数据包解析工具,能够实时地对使用商用密码算法加密的数据包进行解析,解决了商密设备安全性验证不便、网络故障难以排查等问题,对安全设备的测评与私有安全协议开发有一定的借鉴意义。

关键词 协议解析 商密算法 Wireshark IPSec 协议

中图分类号 TP31 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2020.10.050

DESIGN AND IMPLEMENTATION OF PACKET PARSING TOOL FOR CHINA COMMERCIAL CRYPTOGRAPHIC ALGORITHM

Jiang Hua Fan Jinpo Zhang Gang Hu Ronglei
(*Beijing Electronic Science and Technology Institute*, *Beijing 100070*, *China*)

Abstract Under the background of network security product using China cryptographic algorithm, the application of commercial cryptographic algorithm is increasing significantly in many fields such as e-government, finance, transportation, etc. But there are relatively few tools supporting packet analysis of commercial cryptographic algorithm. There are still gaps in the fields of equipment security, encryption and decryption, commercial cryptographic algorithm payload analysis and so on. To solve this problem, a packet parsing tool based on Wireshark for commercial cryptographic algorithm is designed. It can support real-time commercial cryptographic algorithm packet parsing and solve the difficulties of inconvenience verification of commercial cipher device security and troubleshoot network failures. It has certain reference significance in evaluation of safety equipment and private security protocol development.

Keywords Protocol parsing China cryptographic algorithm Wireshark IPSec protocol

0 引 言

随着信息化的发展,商用密码已经成为保障网络安全的重要一环,在包交换网络中,商密数据包承载着协商安全关联、密态数据传输、会话保持和状态查询等任务。网络协议是通信双方为进行数据交换建立的标准体系,通过网络协议解析,可以了解网络数据包在产生和传输过程中的行为[1]。

目前网络协议解析方面的研究多集中在私有协议设计与解析上[2-4],对私有协议的解析通常可以分为插件型和内嵌型两种方式[5]。为了满足保障关键信息基础设施安全的需求,网络安全协议被不断改进,密钥协商、密码算法的使用与标准协议相比产生变化,导致网络流量的解析与网络活动的识别较为不便。

本文介绍了网络协议解析的基本流程与一般方法,研究了网络协议的语法设计与字段处理,对 Wireshark 进行二次开发,在内嵌型开发方式下,完成了对使用商密算法的安全协议全生命周期的解析支持。以使用商密算法的 ISAKMP[6] 和 ESP[7] 协议为例,验证了数据包解析的有效性和正确性。为商密数据包的解析与内容检测提供了借鉴,同时为后续安全测评、用户

收稿日期:2019-06-16。国家自然科学基金项目(61640216)。**蒋华**,教授,主研领域:通信与信息安全。**樊金坡**,硕士生。**张罡**,硕士生。**胡荣磊**,副教授。

行为统计等应用打下了基础。

1 商密数据源的生成

网络安全协议运用密码算法和协议逻辑实现加密和认证功能,用于解决计算机网络面临的安全威胁问题,常见的网络安全协议有 SSL、TLS、IPSec 和 SET 等。根据实际应用需求,文献[8-9]对相关协议的技术规范给出了特定要求。本文使用商密 IPSec VPN 作为数据源。

1.1 商密 IPSec 简介

IPSec 是在 IETF(The Internet Engineering Task Force)的赞助下开发的一组协议,旨在通过 IP 分组交换网络提供安全服务[10],也是目前易于扩展的、完整的网络安全方案。IPSec 有两种工作模式:传输模式和隧道模式。在传输模式下,相互通信的设备 IP 地址必须在其间的网络上可路由,这种模式不具备 NAT 穿越功能。在实际应用中,隧道模式更为常见。下面对隧道模式下商密 IPSec 的两种数据封装方式进行分析,AH 协议与 ESP 协议的数据封装分别如图 1(a)和(b)所示。

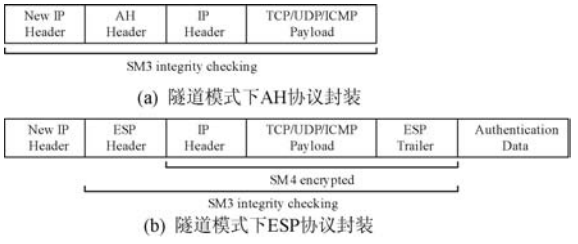

图 1 商密 IPSec 数据封装方式

在隧道模式下,需要产生一个新 IP 头部,IPSec 头(AH 头或 ESP 头)被插入到新 IP 头与原 IP 头之间。在 AH 协议中,IPSec 头包含一个带密钥的 hash 散列,用于提供数据包的完整性保护和抗重放攻击,其密码算法使用 SM3;在 ESP 协议中,ESP 报头包括安全参数索引、序列号信息,ESP 报尾标识了下一个报头与扩展位信息,认证报尾中则写入了数据包的 hash 校验值。与 AH 协议相比,ESP 协议的安全性更高,除了提供完整性保护和数据源认证外,还具有加密功能,其密码算法使用 SM3 与 SM4。

1.2 商密密钥材料的生成

IPSec 协议使用 IKE 协议[11]建立安全联盟 SA 并完成密钥交换的过程。在 IKEv2 的消息交换过程中,除了 IKE_SA_INIT 交换,其余的交换都为密态消息,SK{}中的数据受到加密和完整性保护。密钥材料的生成为后续的商密数据包的机密与完整性校验提供了支持。IKEv2 的消息交换过程如图 2 所示。

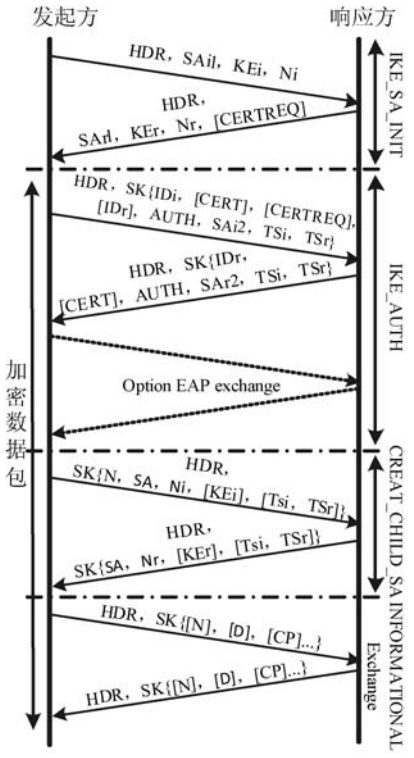

图 2 IKEv2 的消息交换过程图

IKE_SA 的共享密钥以如下方式进行计算:从 IKE_SA_INIT 交换中交换的 nonce 值和 DH 共享密钥计算出称为 SKEYSEED 的数。SKEYSEED 的生成算法如下:SKEYSEED = prf(Ni | Nr, g^ir)。SKETSEED 用于生成七个其他的密钥。各密钥以如下顺序和方式生成:

{SK_d | SK_ai | SK_ar | SK_ei | SK_er | SK_pi | SK_pr} = prf + (SKEYSEED, Ni | Nr | SPIi | SPIr)

在本文选用的商密 VPN 中,prf + 就是 HMAC - SM3,所以:

SK_d = HMAC - SM3(SKEYSEED, Ni | Nr | SPIi | SPIr | 0x01)

SK_ai = HMAC - SM3(SKEYSEED, SK_d | Ni | Nr | SPIi | SPIr | 0x02)

SK_ar = HMAC - SM3(SKEYSEED, SK_ai | Ni | Nr | SPIi | SPIr | 0x03)

SK_ei | SK_er = HMAC - SM3(SKEYSEED, SK_ar | Ni | Nr | SPIi | SPIr | 0x04)

SK_pi = HMAC - SM3(SKEYSEED, SK_ei | SK_er | Ni | Nr | SPIi | SPIr | 0x05)

SK_pr = HMAC - SM3(SKEYSEED, SK_pi | Ni | Nr | SPIi | SPIr | 0x06)

其中:SK_d 用于为 IKE_SA 的 CHILD_SA 生成新的密钥;SK_ai 和 SK_ar 为完整性保护算法的密钥;SK_ei 和 SK_e 为加密所有后续交换的密钥;SK_pi 和 SK_pr 用于生成 AUTH 载荷。

1.3　商密安全隧道的建立

在安全联网终端与服务器之间通过 Gmswan 建立安全隧道。Gmswan 基于开源项目 strongSwan 设计,对 strongSwan 中的加密算法进行了替换,其中对称算法通过修改 aes_crypter. c 文件替换为商密 SM4 算法,摘要算法通过修改 mac_signer. c 文件替换为商密 SM3 算法,非对称算法通过修改 openssl_ec_private_key. c 文件替换为 SM2 算法。此外将密钥协商算法 DH 替换为 SM2,将随机数生成算法 prf + 替换为 HMAC - SM3。Gmswan 通过 ipsec. conf 来配置安全管理的协商参数,在完成商密安全隧道的建立之后,使用 Wireshark 对安全联网终端与服务器之间的数据包进行捕获。

2　协议解析流程

流经网卡的数据被捕获引擎获取后,会将数据包存储到本地,供解析器分析。在 Wireshark 中,其数据包捕获引擎使用 Dumpcap,获取的数据包文件存在 Wiretap 中。捕获的内容会按层级被解析为帧、段和消息载荷数据。Wireshark 采用模块化设计结构,在其六大功能[5]模块中,Epan 负责协议的具体解析。商密协议解析的一般流程如图 3 所示。

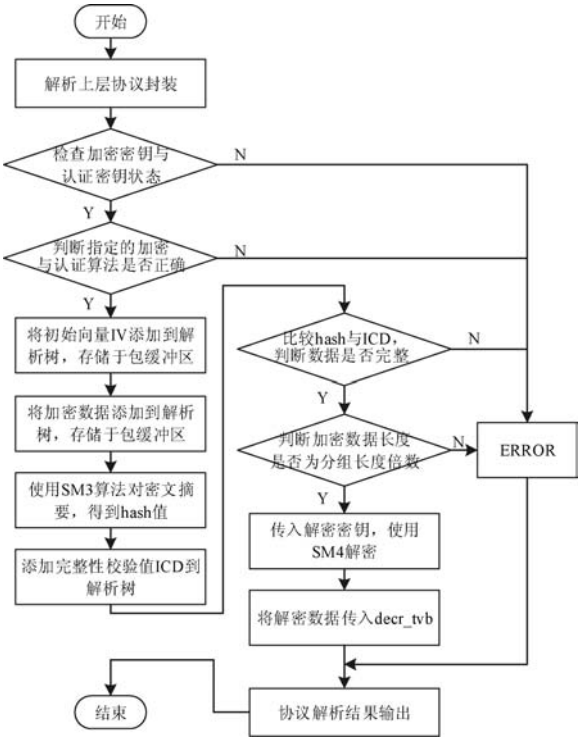

图 3　商密数据包解析流程图

协议解析一般按照 OSI 七层模型进行,从链路层的帧开始,进行层层剥离。在加载协议解析器时,通过通信协议端口号或包头获取特征值的方式,进行过协议解析器的选择匹配。当本层协议报头解析完成,数据部分会被送往上层协议解析器解析,直至整个消息被解析完全。在显示方式上,Wireshark 使用偏移量进行其树形结构的维护。

3　Wireshark 的二次开发

本文使用内嵌型开发方式完成 Wireshark 对商密算法的支持。与编写插件的方式相比,内嵌型方式对源码进行静态编译,无论是程序的执行速度还是调用功能代码方面都更具优势。在内嵌型开发方式下,对 Wireshark 协议解析功能的扩展主要是对 Epan\Dissector 中的文件进行修改,与 IPSec 有关的源码主要是 isakmp. c 和 ipsec. c,分别对应 IKE 的过程和 ESP 与 AH 的封包。

3.1　开发环境配置

进行二次开发之前,需要下载 Wireshark 的源码,安装配置相应的库。下面介绍在 Windows 平台上搭建 Wireshark 开发环境的过程[5]。

1)安装 Chocolate:用于 Windows 平台的包管理。

2)安装 Microsoft C compiler and SDK:本文采用的是 Visual Studio 2015。

3)安装 QT:Wireshark 的主要应用程序使用 QT 的窗口工具。

4)安装 Cygwin:Cygwin 包含了大量 GNU 和开源工具,在安装过程中,除了默认选项外,还需要将 Devel/bison、Devel/flex、Devel/git、Devel/patch、Interpreters/perl 和 Text/docbook - xml45 这些选项选中。

5)安装 Python:为 Cygwin 的包在 Win 32 下顺利执行提供支持。

6)安装 Git:用于下载及管理 wireshark 源码。

7)安装 CMake:用于生成系统的构建文件。

8)安装 Asciidoctor, Xsltproc, DocBook:用于生成文档和用户指南。

完成上述工具的安装后,使用 git clone 下载 Wireshark 源码,即可对代码进行重构。

3.2　SM3 算法替换

Wireshark 中默认支持 ecdsa-sha256 算法,该算法的摘要长度为 256 位。SM3 算法的摘要长度同样是 256 位,在设计商密 VPN 时,数据的完整性使用 SM3 保护,因此在进行商密数据包解析时,需要将受完整性保护的字段找到,使用 SM3 算法对其进行摘要,再与完整性校验和数据 ICD(Integrity Checksum Data)进行比对。Wireshark 使用 libcrypt 库进行密码算法的调

用,其摘要函数接口为 gcry_md_read( ),将其替换为 SM3 算法的函数接口即可。在算法替换之后,原有的国际算法(如 SHA1、SHA2)无法调用,但这与设计商密算法数据包解析的初衷并不冲突,软件解析功能也不会因此受到影响。

3.3 SM4 算法替换

在商密 VPN 设计时,使用 SM4 对分组算法 AES 进行了替换。AES 与 SM4 的分组长度均为 128 位,其密钥长度也相同,均为 16 字节。这简化了算法替换的过程。对 AES 的替换主要是两处函数接口,分别为解密函数 gcry_cipher_decrypt( ) 和传入密钥函数 gcry_cipher_setkey( )。由于不同协议对消息机密性保护的字段不同,向 SM4 解密函数接口中传入的数据长度有所区别。如 IKEv2 中消息解密函数 sm4_crypt( ) 传入的数据长度为 encr_data_len,而 ESP 中消息解密函数 sm4_crypt( ) 传入的数据长度为 decrypted_len_alloc + esp_iv_len,因此需要在设计时,对原有解析规则进行深入分析。

在完成载荷的解密后,使用 dissect_payloads( ) 函数对有效载荷进行解析,通过 proto_item_append_text( ) 将解析字段发送至 GUI,完成解析数据的显示。

3.4 软件编译

完成对源码的扩展后,在编译器的命令行终端 VS2015 x86 Native Tools Command Prompt 中设置环境变量以满足编译过程中对库的调用,创建构建目录并生成构建文件,使用 msbuild /m /p:Configuration = Rel - WithDebInfo Wireshark. sln 对 Wireshark 进行构建。在编译完成后,命令行终端输出如图 4 所示。

图 4 编译 Wireshark 的输出

4 测试与验证

4.1 测试环境搭建

测试环境拓朴图如图 5 所示,在局域网环境下部署了 IPSec VPN 系统,测试环境遵循服务器-客户端模式。PC1 与 Web 服务器通过 VPN 建立的加密隧道进行数据流的交互。PC2 的网卡设置为混杂模式[12],允许网卡获取所有流经网络线路的数据包,运行在其上的 Wireshark,基于伯克利数据包过滤器语法[13]对经过服务器网卡的网络数据包进行抓取,并进行协议解析。Wireshark 测试环境网络配置如表 1 所示。

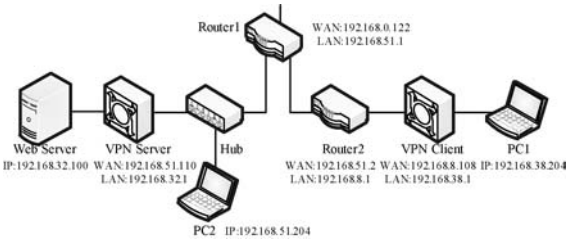

图 5 测试环境拓扑图

表 1 Wireshark 测试环境网络配置

序号	设备	IP	LAN
1	Router1	192.168.0.135	192.168.51.1
2	Router2	192.168.51.2	192.168.8.1
3	VPN Server	192.168.51.110	192.168.32.1
4	VPN Client	192.168.8.108	192.168.38.1
5	PC1	192.168.38.204	—
6	PC2	192.168. 51.204	—
7	Web Server	192.168. 32.100	—

4.2 解析正确性验证

通过抓取数据包并添加过滤规则(isakmp or esp),可以看到 IKE 过程的 4 条信息与 ESP 消息。Wireshark 发行版在输入交换密钥并选择加解密算法与完整性校验算法后能够对 IPSec 协议簇中的密态消息进行解析,但由于 Wireshark 发行版中不含商密系列算法,因此无法正确解析商密数据包,对数据包完整性校验与密态载荷解析均异常,Expert Info 中判定出现严重错误。

Wireshark 发行版解析时的完整性校验警告如下:

[Expert Info(Warning/Checksum):IKEv2 Integrity Checksum Data is incorrect]

[IKEv2 Intergrity Checksum Data is incorrect]

[Severity level:Warning]

[Group:Checksum]

Wireshark 发行版解析时判定数据包格式错误:

[Expert Info(Error/Malformed):MalformedPacket(Exception occurred)]

[Malformed Packet (Exception occurred)]

[Severity level:Error]

[Group: Malformed]

使用基于 Wireshark 的商密数据包解析工具解析

商密数据包,在菜单栏的编辑/首选项/Protocols 中,完成密钥材料的添加。图 6 为解密后商密数据包解析工具中的明文数据,图 7 为商密 VPN 系统日志 server. log 中的解密消息。经过与 Wireshark 中 Decrypted Data 对比,可以看出,本文解析工具对商密数据的解密完全正确。

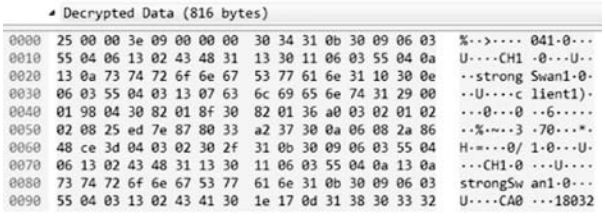

图 6　Wireshark 中的解密数据

```
plain => 816 bytes @ 0xaea00638
   0: 25 00 00 3E 09 00 00 00 30 34 31 0B 30 09 06 03  %..>....041.0...
  16: 55 04 06 13 02 43 48 31 13 30 11 06 03 55 04 0A  U....CH1.0...U..
  32: 13 0A 73 74 72 6F 6E 67 53 77 61 6E 31 10 30 0E  ..strongSwan1.0.
  48: 06 03 55 04 03 13 07 63 6C 69 65 6E 74 31 29 00  ..U....client1).
  64: 01 98 04 30 82 01 8F 30 82 01 36 A0 03 02 01 02  ...0...0..6.....
  80: 02 08 25 ED 7E 87 80 33 A2 37 30 0A 06 08 2A 86  ..%.~..3.70...*.
  96: 48 CE 3D 04 03 02 30 2F 31 0B 30 09 06 03 55 04  H.=...0/1.0...U.
 112: 06 13 02 43 48 31 13 30 11 06 03 55 04 0A 13 0A  ...CH1.0...U....
 128: 73 74 72 6F 6E 67 53 77 61 6E 31 0B 30 09 06 03  strongSwan1.0...
 144: 55 04 03 13 02 43 41 30 1E 17 0D 31 38 30 33 32  U....CA0...18032
```

图 7　商密 VPN 服务器端日志信息

Wireshark 显示了数据包信息并为各字段添加标签,按照协议标准将各消息载荷解析为带有标签的、可理解的文本之后,就完成了对商密数据包的完全解析。图 8 为将商密 ESP 数据包解密后,Wireshark 中显示实际使用的通信协议。图 9 为在有效维护 Wireshark 独特的树形结构的基础上,解析后商密数据包中内容,其中各字段按照协议标准规定的明确格式进行排列。

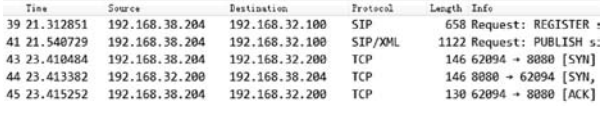

图 8　使用二次开发的 Wireshark 解析商密数据包

图 9　解析后商密数据包中内容

5　结　语

随着移动通信和下一代网络等技术的发展,保障网络信息传输的安全性与可靠性成为研究热点。在政务、金融、教育、医疗等诸多领域,越来越多的安全产品及应用不断涌现。对产品安全性与功能正确性的验证需要一个良好的数据解析工具,而 Wireshark 以其开源和易扩展的特点满足了这一需求。本文对 Wireshark 的体系结构及开发方式进行了分析与研究,给出了一种可扩展的商密数据解析方法与流程,并以一款符合商密《IPSec VPN 技术规范》的 VPN 网关作为数据源,进行了实例的解析与验证。结果表明,基于 Wireshark 的商密数据包解析工具能够对商密数据进行正确有效的解析,为后续设备安全性检测与数据内容匹配提供了借鉴。

参　考　文　献

[1] 罗青林,徐克付,臧文羽,等. Wireshark 环境下的网络协议解析与验证方法[J]. 计算机工程与设计,2011,32(3):770 - 773.

[2] 刘玉瑾. ICA 协议解析关键技术研究[D]. 西安:西安电子科技大学,2017.

[3] 彭勇,王婷,熊琦,等. 针对私有协议的模糊测试技术研究[J]. 北京交通大学学报,2013,37(5):8 - 12.

[4] 唐辉. 基于 Wireshark 二次开发的地铁信号系统应用协议解析插件[J]. 交通与运输,2016(z1):84 - 88.

[5] Wireshark Developer's Guide[EB/OL]. [2019 - 06 - 16]. https://www. wireshark. org/docs/wsdg _ html _ chunked/ChWorksOverview. html.

[6] Maughan D,Schertler M,Schneider M,et al. Internet security association and key management protocol[EB/OL]. (1998 - 11)[2019 - 06 - 16]. https://tools. ietf. org/html/rfc2408.

[7] Kent S. IP encapsulating security payload[EB/OL]. (2005 - 12)[2019 - 06 - 16]. https://tools. ietf. org/html/rfc4303.

[8] 全国信息安全标准化技术委员会. 信息安全技术 IPSec VPN 技术规范:GB/T 36968—2018[S]. 北京:中国标准出版社,2018.

[9] 全国信息安全标准化技术委员会. SSL 协议应用测试规范:GB/T 28457—2012[S]. 北京:中国标准出版社,2012.

[10] Bollapragada V, Khalid M, Wainner S. IPSec VPN 设计[M]. 袁国忠,译. 北京:人民邮电出版社,2012.

[11] Kaufman C,Hoffman P,Nir Y,et al. Internet key exchange protocol version 2[EB/OL]. (2010 - 09)[2019 - 06 - 16]. https://tools. ietf. org/html/rfc5996.

[12] Banerjee U,Vashishtha A,Saxena M. Evaluation of the capabilities of WireShark as a tool for intrusion detection[J]. International Journal of Computer Applications,2010,6(7):1 - 5.

[13] Orzach Y. Wireshark 网络分析实战[M]. 古红霞,孙余强,译. 北京:人民邮电出版社,2015.