doi:10.11959/j.issn.1000-436x.2016121

基于最大似然概率的协议关键词长度确定方法

罗建桢[1]，余顺争[2]，蔡君[1]
(1. 广东技术师范学院电子与信息学院，广东 广州 510665；2. 中山大学电子与信息工程系，广东 广州 510006)

摘 要：提出非齐次左—右型级联隐马尔可夫模型，用于应用层网络协议报文建模，描述状态之间的转移规律和各状态的内部相位变化规律，刻画报文的字段跳转规律和字段内的马尔可夫性质，基于最大似然概率准则确定协议关键词的长度，推断协议关键词，自动重构协议的报文格式。实验结果表明，所提出方法能有效地识别出协议关键词和重构协议报文格式。

关键词：隐马尔可夫模型；协议逆向工程；网络安全；报文格式
中图分类号：TP393 **文献标识码**：A

Method for determining the lengths of protocol keywords based on maximum likelihood probability

LUO Jian-zhen[1], YU Shun-zheng[2], CAI Jun[1]
(1. School of Electronic and Information, Guangdong Polytechnic Normal University, Guangzhou 510665, China;
2. School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, China)

Abstract: A left-to-right inhomogeneous cascaded hidden Markov modelwas proposed and applied to model application protocol messages. The proposed modeldescribed the transition probabilities between states and the evolution rule of phases inside the states,revealed the transition feature ofmessage fields and the left-to-right Markov characteristicsinside the fields. The protocol keywords were inferred by selecting lengths with maximum likelihood probability, and then the message format was recovered. The experimental results demonstrated that the proposed method perform well in protocol keyword extraction and message format recovery.

Key words: hidden Markov model, protocol reverse engineering, network security, message format

1 引言

在网络管理和网络安全领域中，网络协议规范显得尤其重要[1,2]。如网络管理软件需要整合各类协议的规范，以便能够快速高效地识别和解析网络中的各类应用和协议；入侵检测系统(IDS) 和网络防

收稿日期：2015-02-10；**修回日期**：2016-05-10
通信作者：余顺争，syu@mail.sysu.edu.cn
基金项目：国家自然科学基金资助项目（No.61571141, No.61272381）；广东省自然科学基金资助项目（No.2014A030313637, No.2016A030311013）；广东省教育厅特色创新项目（自然科学）基金资助项目（No.2014KTSCX149）；广东省高校优秀青年教师基金资助资助项目（YQ2015105）；广东省应用型科技研发专项基金资助项目（No.2015B010131017）；广东省科技计划基金资助项目（No.2014A010101156）；广东省教育厅省级重大基金资助项目（No.2014KZDXM060）；广东省普通高校国际合作重大基金资助项目（No.2015KGJHZ021）；广东省公益研究与能力建设专项基金资助项目（No.2014A010103032）
Foundation Items: The National Natural Science Foundation of China (No.61571141, No.61272381), The Natural Science Foundation of Guangdong Province(No.2014A030313637, No.2016A030311013), Guangdong Provincial Department of Education Innovation Project (No.2014KTSCX149), The Excellent Young Teachers in Universities in Guangdong Province (No.YQ2015105) Guangdong Provincial Application-Oriented Technical Research and Development Special(No.2015B010131017), Science and Technology Planning Project of Guangdong Province(No.2014A010101156), Science and Technology Major Project of Education Department of Guangdong Province (No.2014KZDXM060), International Scientific and Technological Cooperation Projects of Education Department of Guangdong Province (No.2015KGJHZ021), Science and Technology Project of Guangdong Province (No.2014A010103032)

火墙等网络安全设备都需要获取网络应用的协议规范并配置相应的安全规则和安全策略；只有深入了解命令控制协议(C&C)才能有效检测并防御僵尸网络[3]。除此以外，要实现基于不同通信协议的多个系统之间的互操作，则必须清楚各协议的规范，才能设计和开发兼容多系统的平台[4~6]。协议规范还可以与自动化模糊测试结合，以快速高效地发现软件系统中的漏洞[7,8]。

常见的协议规范可以从协议开发者或 IETF[9]发布的公开文档中获得，但是一些私有的网络应用或协议的开发者，往往会出于商业机密或者其他原因而拒绝提供有关的协议规范文档。网络攻击或恶意软件的制造者也不愿意公开相应的协议规范。在这种情况下，网络管理员或安全专家就必须依赖于协议逆向工程技术来重构协议的规范。传统的协议逆向工程主要依靠专家对网络流量的人工分析，手工推断协议的规范。人工分析方法的准确性严重依赖于网络专家的知识水平，而且非常耗时和容易出错。

随着 Internet 尤其是移动互联网的飞速发展，新兴的网络应用（如微博、微信以及各种 App 等）和新型网络攻击不断呈现，越来越多的网络应用采用了私有的协议，以致大约 40% 的网络流量无法识别[10]。同时，网络用户数量持续攀升，网络流量呈现多样化的同时又具有数据海量化的特征，基于人工分析的协议逆向工程方法严重影响了网络管理的运作效率和妨碍了网络安全应用的发展。因此，在大数据背景下，研究能适应目前网络形势和满足当今网络安全需求的自动协议逆向分析方法成为研究热点。

本文的主要贡献是提出一种非齐次的左—右型级联隐马尔可夫模型，用于对应用层网络协议报文建模，并基于最大似然概率准则确定协议关键词的长度，最终自动重构协议的报文格式。特别说明，本文只研究基于明文的协议报文。加解密过程需要增加时间和存储的额外开销，在不涉及敏感信息的情况下，网络应用选择基于明文的通信协议，更有利于降低处理时间，提升用户体验[11~14]。因此，研究基于明文的协议报文依然具有必要性。

2 相关工作

网络协议逆向工程[15]的目的是，在不需要协议规范先验知识的条件下，通过分析协议的流量或协议的可执行代码，还原协议的报文格式，重构协议的交互状态机。协议逆向工程技术大致可分为 3 种：人工分析、网络流量分析和动态二进制分析。

1) 人工分析

Samba、Pidgin 和 Rdesktop 等开源项目都是人工逆向分析的典型例子，其准确性依赖于安全专家的知识水平，而且周期长、易出错、效率低。

2) 网络流量分析

该方法仅分析协议的网络数据分组，一直以来备受国内外众多研究者所关注。PI 项目[16]最早提出借助生物信息学的序列比对算法识别网络报文的字段。Cui 等[17]基于递归聚类算法提取报文的基本单元，还原协议的报文格式。Wang 等[18]根据报文内部的 *N*-gram 特性识别报文关键字，再运用序列比对算法分析协议的报文格式。Zhang 等[19,20]采用基于 Trie 数据结构的专家投票算法提取协议的特征字。He 等[21]逆向分析 TLV 结构的网络数据分组，重构其格式。Li 等[22]和 Tao 等[23]基于多序列比对算法，提取二进制协议的报文格式。Meng 等[24,25]研究未知二进制协议状态机的推断方法。Gascon 等[26]通过逆向分析协议的交互状态机，实现私有协议的有状态的黑盒子模糊测试。在国内，李伟明等[8]提出基于报文长度的报文格式提取方法，并将其应用于自动化模糊测试；肖明明等[27]提出基于差错纠正的文法推断方法从应用层协议交互过程中的报文序列反推协议状态机；游翔等[28]提出一种将端口与正则表达式相结合的飞信协议识别方法，基于飞信通信序列关系从大量混杂的数据分组中快速定位飞信业务报文，获取飞信的交互状态机。

近年来，关于未知协议栈的帧切分研究也受到学术界的关注。例如，岳旸等[29]提出一种基于聚类的未知协议二进制数据帧分离方法；琚玉建等[30]运用位置差关联规则推断未知协议数据帧的帧头位置及帧长，实现协议帧的切分；Li 等[31]提出基于频繁项挖掘的无线协议帧分离算法。

以上工作与本文的最大区别在于确定协议关键词长度的方法。PI 项目是基于 LCS 准则来确定关键词长度的，这种方法具有较明显的经验性，缺乏严密的理论基础。Wang 等[18]基于 *N*-gram 的方法将报文分割为相等长度的片段，其准确率受 *N* 的取值影响，也难以捕获报文内部的隐藏结构。Discoverer 提取的协议关键词的长度决定于分隔符的选取，也是不够严密的。而本文提出基于最大似

然概率的方法来确定协议关键词长度，具有严密的数学基础，分析结果也更加合理。另外，Zhang 等[19,20]为了减少内存占用的空间，在构造 Trie 结构时、删剪了频率较小的分支，因而可能导致丢失部分特征字；Zhang 等也没有重构协议的报文格式。

3) 动态二进制分析

动态二进制分析方法的核心思想是将协议的可执行程序置于一个可控制的环境下运行，跟踪观察程序处理报文的运行时信息（包括指令序列、堆栈和寄存器使用信息等），据此反推协议的报文格式，如 Polyglot[32]、Dispatcher[33,34]、Autoformat[35]以及 Lin 等[36]和 Cui 等[37]的工作。另外，动态二进制分析方法可用于分析加密的安全协议和恶意程序[38~42]。

然而，未知协议或网络攻击的可执行代码难以获取，某些加入防逆向技术的程序不能在可控环境下正确运行，诸如此类的限制条件导致动态二进制分析方法只能局限在特定的应用场景中。相比而言，基于数据分组分析的方法只需要捕获待分析的网络应用的流量，其实现和部署都要比基于二进制分析的方法容易。因此，本文只关注基于数据分组分析的协议逆向分析方法。

3　模型描述

3.1　应用层网络协议规范

应用层协议的会话（session）是 Internet 上 2 台主机的进程之间互相通信的基本形式。每个会话由它的五元组唯一确定，并由一对方向相反的流（flow）组成。流定义为 2 个进程之间通信时传输的字节流，也可以认为是 2 个进程之间通信时在同一方向上传递的报文序列。图 1 描述了应用层协议会话的简单实例，其中，m_i 表示会话中的第 i 个报文，报文序列 $m_1, m_3, m_5, \cdots$ 和 $m_2, m_4, \cdots$ 分别表示一个会话中 2 个方向传输的 2 个报文序列。报文是应用层协议进行数据交换的基本单元。

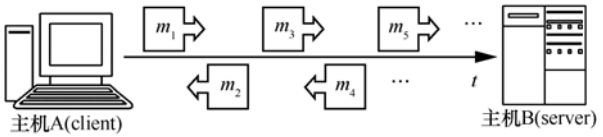

图 1　应用层协议的会话过程

网络协议规范包括协议的报文格式和协议状态机。报文格式刻画了协议所使用的各种报文的组成结构和各组成部分的语义信息，而协议状态机则描述了会话过程中不同类型的报文之间交互的次序。报文格式可以采用不同的表现形式。在本文中，报文格式定义为字段序列。图 2 给出了应用层协议的报文格式，其中，关键词字段的内容（如“GET”、“HTTP/1.1”）为协议关键词，数据字段的长度和内容都是可变的字段。一个关键词字段后面往往紧跟一个数据字段，此时数据字段的内容通常是其紧跟的协议关键词的数据值或参数值。不同类型的报文具有不同的协议关键词和字段序列。因此，在提取报文格式时，只要挖掘出报文中的协议关键词就可以将报文划分为由一系列字段组成的序列。协议关键词定义为协议采用的字符常量、协议的状态码或分隔符等。如在 HTTP 协议中，“GET”、“200”、“OK”等都是协议关键词。

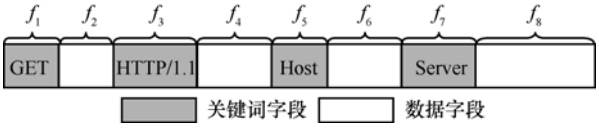

图 2　应用层协议的报文格式

3.2　非齐次左—右型级联隐马尔可夫模型

应用层网络协议的报文可表示为一个字节序列：$\boldsymbol{o} = o_1 o_2 \mathrm{L}\ o_T$，其中，$T$ 为报文的长度。如图 2 所示，报文的结构则可看作是一个字段序列，即 $\boldsymbol{o} = \boldsymbol{o}^{(1)}\boldsymbol{o}^{(2)}\mathrm{L}\ \boldsymbol{o}^{(R)}$，其中，$\boldsymbol{o}^{(r)} = o_1^{(r)} o_1^{(r)} \mathrm{L}\ o_{T_r}^{(r)}$ 表示报文的第 r 个字段，r=1,2,⋯,R。

随着时间的推移，报文中的字段依次出现。假定协议的报文可以用一个随机过程来描述，报文从一个字段向另一个字段转移时，对应的随机过程也从一个隐状态向另一个隐状态转移(隐状态的状态空间为 $\boldsymbol{S}$={1,2,⋯,M})。字段之间的转移概率决定于随机过程中隐状态之间的转移概率，即 a_{ij}，其中，$i,j \in \boldsymbol{S}$。a_{ij} 表示给定状态 i 的条件下，随机过程从状态 i 向状态 j 的转移概率，即

$$a_{ij} = P[s_{t+1} = j \mid s_t = i] \tag{1}$$

状态间的转移概率还满足

$$a_{ii} = 0, i \in \boldsymbol{S} \tag{2}$$

$$\sum_{j \in \boldsymbol{S}} a_{ij} = 1, i \in \boldsymbol{S} \tag{3}$$

字段内部也会随着时间的推移而在相位上发生某些改变，例如关键词字段从左向右逐个出现关键词的各个字符，直到一个关键词的所有字符都按

位置先后逐一出现，这时该字段的相位也进化完毕; 数据字段的相位推进则体现在字段长度从小至大逐一增长。因此，这种相位的改变可以用一个具有相位的从左到右型的马尔可夫(left-to-right HMM)过程[43,44]表示。在从左到右型的马尔可夫过程中，随着时间从左向右推移，字段内部的相位也从低向高推进。

假定字段的最大长度为 K，那么对每个给定状态 i 定义 $\boldsymbol{K}$ 个相位：$\boldsymbol{K}$={1,2,⋯,K}，用(i，k) 表示随机过程处于状态 i 的相位 k，相位 k 代表一个字段的进化程度，或者代表字段的马尔可夫过程历经的程度。在一个状态 i 中，随着时间的推移，状态 i 的相位 k 只能从相位 1 开始，并逐一向右转移，即由 k 转变到 k+1，再从 k+1 转变到 k+2，或者从某一相位直接向 K(代表消亡相位)相位转移，因此，只有$(i,k)\to(i,k+1)$和$(i,k)\to(i,K)$的转移概率不等于 0，而其他相位之间的转移概率定义为 0。在给定(i，k) 的情况下，观测到字符 c 的概率为

$$b_{i,k}(c)=P[o_t=c\,|\,s_t=i,p_t=k] \tag{4}$$

其中，c 是当前观测值(报文中的一个字节)，观测值的集合为 V={0,1,2,⋯,255}，即一个字节的所有可能取值。

当从某个状态(不等于 i)转移到状态 i 时，首先进入状态 i 的相位 1，在相位 1 时，以 $b_{i,1}(c)$ 的概率观察到观测值 c，接着以相位转移概率 $p_i(1)$ 转移到相位 2，或者转移概率 $1-p_i(1)$ 结束当前相位，并以状态转移概率 $a_{ii'}$ 转移到下一个状态 i'; 在相位 k 时，以 $b_{i,k}(c)$ 的概率观察到观测值 c，接着以相位转移概率 $p_j(k)$ 转移到相位 k+1，或者以转移概率 $1-p_j(k)$ 结束当前相位，然后以状态转移概率 $a_{jj'}$ 转移到下一个状态 j'，依此类推。

在这里，$p_i(k)$ 表示在给定状态 i 时，由相位 k 向相位 k+1 转移的概率分布，其定义为

$$p_i(k)=P[s_{t+1}=i,p_{t+1}=k+1\,|\,s_t=i,p_t=k],i\in\boldsymbol{S},k\in\boldsymbol{K} \tag{5}$$

然而，现有的模型不能完整地对应用层协议报文结构进行建模。隐马尔可夫模型只刻画了隐状态之间的状态转移规律，但并没有刻画状态内部的微观特性。即使是隐半马尔可夫模型也只是笼统地描述了隐状态的持续时间长度，而没有真正揭示状态内部的变化规律。因此，本文提出非齐次左—右型级联隐马尔可夫模型(LRIHMM, left-to-right inhomogeneous cascaded HMM)，用于对应用层协议的报文结构进行建模，刻画报文的字段间转移规律和字段内部的左右型马尔可夫性质。LRIHMM 的模型参数记为 $\lambda=\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{P},\pi\}$，其中，$\boldsymbol{A}$ 为模型的状态转移概率矩阵，$\boldsymbol{B}$ 为观测概率矩阵，$\boldsymbol{P}$ 为字段的相位转移概率矩阵，π 为初始状态的概率分布。

状态转移概率矩阵定义为

$$\boldsymbol{A}=\{a_{ij}\},i,j\in\boldsymbol{S} \tag{6}$$

观测概率矩阵定义为

$$\boldsymbol{B}=\{b_{i,k}(c)\},i\in\boldsymbol{S},k\in\boldsymbol{K},c\in\boldsymbol{V} \tag{7}$$

字段的相位转移概率矩阵定义为

$$\boldsymbol{P}=\{p_i(k)\},i\in\boldsymbol{S},k\in\boldsymbol{K} \tag{8}$$

初始状态的概率分布定义为

$$\pi=\{\pi_i\},i\in\boldsymbol{S} \tag{9}$$

其中，$\pi_i=P[s_1=i],i\in\boldsymbol{S}$，且满足 $\sum_{i=1}^{M}\pi_i=1$。

4 基于 LRIHMM 的报文模型

4.1 报文模型参数估计

如果一个字符串 x 是另一个字符串 x'的子串，则记为：$x\subset x'$。设 F 为频繁项集合，那么 F 的最长频繁项集合 F_L 定义为: 任意给定 $x\in F_L$，不存在 $x'\in F_L$ 且 $x'\in F$，使 $x\subset x'$。

本文在训练 LRIHMM 时，首先基于已有工作提取报文集里的频繁字符串集[45]，再找出频繁字符串集里的最长频繁项，组成最长频繁项集 F_L。令 F_L 中的每个字符串都与一个状态对应，如果 $x\in F_L$ 是状态 i 对应的一个字符串，则记 x 为 x_i，且 x_i 的所有子字符串 $a\in x_i$ 都可能是状态 i 的观测值。因此，LRIHMM 的关键词状态数目为 $N=|F_L|$。另外定义若干个新的状态，代表数据状态，它的观测值是观测序列集中所有可能的字符。关键词状态数目与数据状态数目的总和为 M。

LRIHMM 参数的初始化过程如下。

相位数为 $K=\max_{x\in F_L}|x|$。

初始状态服从等概率分布的初始化：$\pi_i=\dfrac{1}{M}$。

状态转换概率矩阵服从等概率分布：$a_{ij}=\dfrac{1}{M-1}$，并满足 $\sum_{j\in\boldsymbol{S}}a_{ij}=1,a_{ii}=0$。

观测概率的初始化为

$$b_{i,k}(c)=\begin{cases}\exp(|x_i|-K), & i\leqslant N, k\leqslant|x_i|\\ 0, & i\leqslant N, k>|x_i|\\ \exp(-k), & i>N\end{cases} \tag{10}$$

相位转移概率的初始化为

$$p_i(k)=\begin{cases}random(), & i\leqslant N, k\leqslant|x_i|\\ 0, & i\leqslant N, k>|x_i|\\ random(), & i>N\end{cases} \tag{11}$$

本文提出基于前向后向算法思想[46~48]的参数更新算法用于训练 LRIHMM。

首先，定义前向变量

$$\alpha_t(i,k)\equiv P[o_1^t, s_t=i, p_t=k\,|\,\lambda], i\in\boldsymbol{S}, k\in\boldsymbol{K} \tag{12}$$

$$\alpha_t(i)\equiv P[o_1^t, s_t]=i\,|\,\lambda], i\in\boldsymbol{S} \tag{13}$$

其中，p_t 表示 t 时刻的相位，o_1^t 表示 $o_1o_2\mathrm{L}\ o_t$，$s_{t]}=i$ 表示状态 i 终止于时刻 t，即 $s_t=i$，但 $s_{t+1}\neq i$。同理，本文中出现的 $s_{[t]}=i$ 表示状态 i 开始于时刻 t，即 $s_t=i$，但 $s_{t-1}\neq i$。

前向变量的初始化条件

$$\alpha_1(i,1)=\pi_i b_{i,1}(0_1), i\in\boldsymbol{S} \tag{14}$$

$$\alpha_1(i,k)=0, i\in\boldsymbol{S}, k>1 \tag{15}$$

$$\alpha_1(i)=\pi_i b_{i,1}(0_1)\big(1-p_i(1)\big), i\in\boldsymbol{S} \tag{16}$$

$\alpha_t(j,1)$ 表示在时刻 t 时为状态 j，且处于第 1 个相位，根据 LRIHMM 的定义，t–1 时刻的状态必定不等于 j，因此，$\alpha_t(j,1)$ 是由 t–1 时刻的状态 j 以外的所有可能状态转移而来的。而 $\alpha_t(j,k), 1<k\leqslant K$ 则肯定是由 t–1 时刻的 $\alpha_{t-1}(j,k-1)$ 通过相位进化而得到的。因此，可得

$$\alpha_t(j,1)=\sum_{i\in\boldsymbol{S}}\alpha_{t-1}(i)a_{ij}b_{j,1}(o_t), j\in\boldsymbol{S}, k=1 \tag{17}$$

$$\alpha_t(j,k)=\alpha_{t-1}(j,k-1)p_j(k-1)b_{j,k}(o_t), j\in\boldsymbol{S}, k>1 \tag{18}$$

进而可得

$$\alpha_t(j)=\sum_{k\in\boldsymbol{K}}\alpha_t(j,k)\big(1-pj(k)\big), j\in\boldsymbol{S} \tag{19}$$

计算 $\alpha_t(j)$ 时需要乘以 $1-p_j(k)$，表示状态 j 的相位进化到消亡态，并准备进入下一个状态的相位 1。

再定义后向变量

$$\beta_t(i,k)\equiv P[o_{t+1}^T\,|\,s_t=i, p_t=k, \lambda], i\in\boldsymbol{S}, k\in\boldsymbol{K} \tag{20}$$

$$\beta_t(i)\equiv P[o_{t+1}^T\,|\,s_{t]}=i, \lambda], i\in\boldsymbol{S} \tag{21}$$

由于

$$\beta_t(i,k)=P[o_{t+1}^T, s_{t+1}=i, p_{t+1}=k+1\,|\,s_t=i, p_t=k, \lambda]+$$
$$P[o_{t+1}^T, s_{t+1}=j, p_{t+1}=1\,|\,s_t=i, p_t=k, \lambda]$$

其中，$j\neq i$，因此可得

$$\beta_t(i,k)=p_i(k)b_{i,k+1}(o_{t+1})\beta_{t+1}(i,k+1)+\big(1-p_i(k)\big)\beta_t(i) \tag{22}$$

其中，

$$\beta_t(i)=\sum_{j=1}^{M}a_{ij}b_{j,1}(o_{t+1})\beta_{t+1}(j,1) \tag{23}$$

后向变量初始化条件为

$$\beta_T(i)=1, i\in\boldsymbol{S} \tag{24}$$

$$\beta_T(i,k)=1, i\in\boldsymbol{S}, k\in\boldsymbol{K} \tag{25}$$

为了更新模型的状态转移概率矩阵，定义以下中间变量

$$\begin{aligned}\xi_t(i,j)&\equiv P[o_1^t, s_t=i, s_{t+1}=j, i\neq j]\\ &=\alpha_t(i)a_{ij}b_{j,1}(o_{t+1})\beta_{t+1}(j,1), i,j\in\boldsymbol{S}\end{aligned} \tag{26}$$

随机过程在 t 时刻的状态 i 的概率为

$$\gamma_t(i)\equiv P[s_t=i, o_1^T] \tag{27}$$

由于

$$\begin{cases}P[s_t^{t+1}=j, o_1^T]=p[s_t=j, o_1^T]-p[s_{t]}=j, o_1^T]\\ P[s_t^{t+1}=j, o_1^T]=p[s_{t+1}=j, o_1^T]-p[s_{[t+1}=j, o_1^T]\end{cases} \tag{28}$$

可推导得以下递归式

$$\gamma_t(j)=\gamma_{t+1}(j)+\sum_{\substack{i\in S\\ i\neq j}}\big(\xi_t(j,i)-\xi_t(i,j)\big) \tag{29}$$

递推计算的初始化条件为: $\gamma_T(i)=\alpha_T(i)$。

为了更新模型的相位进化概率，定义以下 2 个变量

$$\gamma_t(i,k)\equiv p[o_1^T, s_t=i, p_t=k\,|\,\lambda]=\alpha_t(i,k)\beta_t(i,k), i\in\boldsymbol{S} \tag{30}$$

$$\begin{aligned}&\gamma_t(i,k,k+1)\equiv p[o_1^T, s_t=i, s_{t+1}=i, p_t=k, p_{t+1}=k+1\,|\,\lambda]\\ &=\alpha_t(i,k)p_i(k)b_{i,k+1}(o_{t+1})\beta_{t+1}(i,k+1), i\in\boldsymbol{S}, k\in\boldsymbol{K}\end{aligned} \tag{31}$$

报文模型的参数更新公式

$$\hat{\pi}_i=\frac{\gamma_1(i)}{\sum_{i\in\boldsymbol{S}}\gamma_1(i)}, i\in\boldsymbol{S} \tag{32}$$

$$\hat{a}_{ij}=\frac{\sum_t\xi_t(i,j)}{\sum_j\sum_t\xi_t(i,j)}, i,j\in\boldsymbol{S} \tag{33}$$

$$\hat{p}_i(k)=\frac{\sum_t \gamma_t(i,k,k+1)}{\sum_t \gamma_t(i,k,k+1)+\sum_t \gamma_t(i,k,K)},i\in \boldsymbol{S},k<K \tag{34}$$

$$\hat{b}_{i,k}(c)=\frac{\sum_t \gamma_t(i,k)I(o_t-c)}{\sum_t \gamma_t(i,k)},i\in \boldsymbol{S},k\in \boldsymbol{K} \tag{35}$$

4.2 基于 Viterbi 算法的字段划分

通过使用大量报文集对 LRIHMM 训练得到模型的估计参数后，便可以基于 Viterbi 算法推断具有最大似然概率的字段长度。因此，首先定义 Viterbi 变量

$$\begin{cases}\delta_t(i,k)=\max\limits_{s_1s_2\mathrm{L}\,s_{t-1}} P[s_1^{t-1},o_1^t,s_t=i,p_t(i)=k\mid\lambda],i\in\boldsymbol{S},k\in\boldsymbol{K}\\ \delta_t(i)=\max\limits_{s_1s_2\mathrm{L}\,s_{t-1}} P[s_1^{t-1},o_1^t,s_t=i\mid\lambda],i\in\boldsymbol{S}\end{cases} \tag{36}$$

Viterbi 变量的初始化为

$$\delta_1(i,k)=\begin{cases}\pi_i b_{i,1}(o_1),k=1\\ 0, \quad \text{其他}\end{cases}$$
$$\delta_1(i)=\delta_1(i,1)\left(1-p_i(1)\right) \tag{37}$$

定义 $\Delta_t^{(k)}(j)$ 表示在 t 时刻状态 j 的具有最大似然概率的状态持续长度，$\Delta_t^{(s)}(j)$ 则表示在 t 时刻状态 j 的具有最大似然概率的前一个状态。

$$\delta_t(j,k)=\begin{cases}\max\limits_{i\in\boldsymbol{S}}\delta_{t-1}(i)a_{ij}b_{j,1}(o_t),k=1\\ \delta_{t-1}(i,k-1)p_j(k-1)b_{j,k}(o_t),k>1\end{cases} \tag{38}$$

$$\delta_t(j)=\max_{k\in\boldsymbol{K}}\delta_t(j,k)\left(1-p_j(k)\right) \tag{39}$$

$$\Delta_t^{(k)}(j)=\arg\max_{k\in\boldsymbol{K}}\left(\delta_t(j,k)\left(1-p_j(k)\right)\right) \tag{40}$$

$$\Delta_t^{(s)}(j)=\arg\max_{i\in\boldsymbol{S}}\left(\delta_{t-\Delta_t^{(k)}(j)}(i)a_{ij}\right) \tag{41}$$

Viterbi 反推最佳状态序列的回溯过程。

首先，令 $t=T$，那么最佳状态序列的最后一个状态为

$$s_t^{(*)}=\arg\max_{i\in\boldsymbol{S}}\left(\delta_t(i)\right) \tag{42}$$

该状态的长度为

$$\tau_t^{(*)}=\Delta_t^{(k)}(s_t^{(*)}) \tag{43}$$

各状态对应的字段为

$$f_{t-\tau_t^{(*)}+1}=\left\langle w_{t-\tau_t^{(*)}+1},l_{t-\tau_t^{(*)}+1},s_{t-\tau_t^{(*)}+1}\right\rangle=\left\langle o^t_{\,t-\tau_t^{(*)}+1},\tau_t^{(*)},s_t^{(*)}\right\rangle \tag{44}$$

其中，$w_{t-\tau_t^*+1}$ 为字段的值，$l_{t-\tau_t^*+1}$ 为字段的长度，$s_{t-\tau_t^*+1}$ 为字段对应的隐状态。

最佳状态序列上的其他时刻的状态以及字段可由以下 Viterbi 的递归过程推导

$$\tau_t^{(*)}=\Delta_t^{(k)}(s_t^{(*)}) \tag{45}$$

$$f_{t-\tau_t^{(*)}+1}=\left\langle w_{t-\tau_t^{(*)}+1},l_{t-\tau_t^{(*)}+1},s_{t-\tau_t^{(*)}+1}\right\rangle=\left\langle o^t_{\,t-\tau_t^{(*)}+1},\tau_t^{(*)},s_t^{(*)}\right\rangle \tag{46}$$

$$s^t_{t-\tau_t^*+1}=s_t^{(*)} \tag{47}$$

$$s^{(*)}_{t-\tau_t^{(*)}}=\Delta_t^{(s)}(s_t^{(*)}) \tag{48}$$

$$t=t-\tau_t^{(*)} \tag{49}$$

以上递归过程一直进行到找到 $s_1^{(*)}$ 为止。最后，将所有字段 $f_{(*)}$ 按下标 (*) 从小到大重新排列并重新编号，下标最小的编号为 1，次之为 2，…，最后一个编号为 R，即总共有 R 个字段

$$f_1=\langle w_1,l_1,s_1\rangle \tag{50}$$

$$f_2=\langle w_2,l_2,s_2\rangle \tag{51}$$

M

$$f_R=\langle w_R,l_R,s_R\rangle \tag{52}$$

对于任意 r=1,2,…,R，当1≤s_r≤N 时，f_r 为一个关键词字段，其中，w_r 为对应的协议关键词。否则，当 $N<s_r\leqslant M$ 时，f_r 为一个数据字段，w_r 是一个非协议关键词的普通字符串。

5 实验结果

本文在配置为 2.93 GHz 的双核 CPU、2GB 内存、操作系统为 Windows XP 的 PC 上基于 C/C++ 和 Matlab 实现了所提出的方法。为了评价 LRIHMM 的有效性和准确性，同时还实现了本文所提出的方法与 2 个经典方法（文献[16]方法和 Discoverer 方法[17]）做比较。

5.1 实验数据

Discoverer 处理长度大于 2 048 byte 的报文时，采用的方法是截尾，即只保留报文的前 2 048 byte。这种处理方法是合理的，因为大多数报文的报文格式主要体现在报文的头部，报文头部之后的部分通常为用户相关的数据，该部分数据不但不能促进，反而会妨碍报文格式的推断。为了与 Discoverer 统一比较标准，以及降低系统处理数据的计算时间，LRIHMM 也采用了相同的数据截尾处理。但是 PI

项目是保留了原有系统的处理方法，即对数据分组没有作截尾处理。

本文所采用的实验数据采集于真实网络环境(中山大学信息科学与技术学院的网络出口)，如表 1 所示。所采集的网络流量首先经过过滤噪音、重构会话、重组报文和长报文截断等处理，得到纯净无噪的报文集。

表 1　　实验数据集

协议名称	连接个数	数据分组数目	数据规模
HTTP	244 443	4.1×10^6	9 870.3
FTP	27 292	225 823	19.1
SMTP	18 091	111 736	57.2
POP	5 442	423 885	86.6
SSDP	15 184	315 929	78.9
BitTorrent	389 105	1.7×10^6	512.4

5.2　评价标准

真阳性(*TP*)是指被正确识别的协议关键词数量。

假阳性(*FP*)是指被错误识别的协议关键词数量。

假阴性(*FN*)是指没有被识别的协议关键词数量。

本文从准确率、召回率和 *F*1 值 3 个指标评价推断协议关键词的实验结果，定义如下。

准确率(ρ)：被正确识别的协议关键词数量占被识别的协议关键词总数的比例，即 $\rho=\dfrac{TP}{TP+FP}$。

召回率(γ)：被正确识别的协议关键词数量占真实协议关键词总数的比例，即 $\gamma=\dfrac{TP}{TP+FN}$。

*F*1 值(*f*)：$f=\dfrac{2\rho\gamma}{\rho+\gamma}$。

报文格式的评价指标为报文格式的覆盖率，定义如下。

覆盖率：实验推断的报文格式所覆盖的报文占所有报文总数的比例。

5.3　结果分析

5.3.1　举例

表 2~表 4 分别列举了 3 个系统输出的 HTTP 报文格式。LRIHMM 推断的报文格式是以字段序列的形式输出，每个报文都可划分为关键词字段和紧接着关键词字段的数据字段。关键词字段的字段值为常量，在报文中频繁出现，可作为报文中字段的分界标志，还具备相关的语义信息，如某些关键词指示当前通信的状态。

表 2　　LRIHMM 输出的 HTTP 消息格式

字段序号	字段	属性
F(1)	K(“GET /”)	协议关键词
F(2)	VD	可变字段
F(3)	K(“HTTP/1.1”)	协议关键词
F(4)	K(“Host:”)	协议关键词
F(5)	VD	可变字段
F(6)	K(“User-Agent:”)	协议关键词
F(7)	VD	可变字段
F(8)	K(“Accept:”)	协议关键词
F(9)	VD	可变字段
F(10)	K(“Content:”)	协议关键词
F(11)	VD	可变字段
F(12)	K(“Connection:”)	协议关键词
F(13)	VD	可变字段
F(14)	K(“Referer:”)	协议关键词
F(15)	VD	可变字段
F(16)	K(“Cookie:”)	协议关键词
F(17)	VD	可变字段
M	M	M

表 3　　Discoverer 输出的 HTTP 消息格式

序号	Token
1	c(t，“GET”)
2	v(t)
3	c(t,“rep. . . ”)
4	v(t)
5	c(t,“int. . . ”)
6	v(t)
7	c(t,“HTTP/1.1”)
8	c(t,“Host:”)
9	v(t)
10	c(t,“.com”)
11	v(t)
12	c(t,“User. . . ”)
13	v(t)
14	c(t,“ocspd”)
15	v(t)
16	c(t,“(unknown”)
17	v(t)
18	c(t,“version)”)
19	v(t)
20	c(t,“CFNetwork”)
21	v(t)
22	c(t,“Darwin”)
23	v(t)
24	c(t,“(x86 64)”)
25	v(t)
26	c(t,“Conne. . . ”)
27	v(t)
28	M

表 4　PI 输出的 HTTP 消息格式

字节序号	字节内容	ASCII 值	字节属性
1	0x47	‘G’	常量
2	0x45	‘E’	常量
3	0x54	‘T’	常量
4	0x20	‘ ’	空格
5	0x20	‘ ’	空格
6	0x20	‘ ’	空格
7	0x20	‘ ’	空格
8	0x20	‘ ’	空格

Discoverer 输出的报文格式表现为 token 序列。有些 token 的值是常量，有些 token 的值和长度都是可变的。如表 3 所示，c(t,“GET”)、c(t,“HTTP/1.1”)、c(t,“ocspd”)和 c(t,“(x86_64)”) 都是常量 token，其中，前 2 个 token 的值是本文定义的协议关键词，后 2 个 token 的值是报文中的用户数据的一些参数值，在报文格式中并无意义，是冗余的 token。

PI 对输入的报文执行序列比对算法，得到的结果是多个报文的公共子字符串。由于 PI 所采用的序列比对算法处理的其他单元是字节，所以得到的结果是一个公共字节序列。表 4 为输入的 1 000 个报文的序列比对结果，得到 HTTP 请求报文的格式。该格式只包含一个协议关键词(“GET”)和若干空格，而更多其他协议关键词却没有出现。

从以上例子可看出，LRIHMM 输出的报文格式与 Discoverer 输出的报文格式相似，但是 LRIHMM 输出的报文格式比 Discoverer 输出的报文格式更为简洁，更为准确。LRIHMM 输出的报文格式中出现的只有关键词而没有与用户相关的参数等冗余数据，而 Discoverer 输出的报文格式则会出现一些与真实报文格式无关的 token，例如$“ocspd”$和$“(x86_64)”$。PI 输出的报文格式过于泛化，使报文格式退化为报文中的一个特征字符串，从而丢失了很多与格式密切相关的信息。

5.3.2　准确率与召回率

实验结果的准确率和召回率分别如表 5 和表 6 所示。需要说明的是，在计算准确率和召回率时，真实关键词指的是实验数据集里出现过的协议关键词，任何在实验数据集里没出现过的协议关键词将不作考虑。从实验结果来看，LRIHMM 的准确率和召回率都比 Discoverer 和 PI 系统的要高。

表 5　协议关键词的准确率/%

系统	HTTP	FTP	SMTP	POP	SSDP	BitTorrent
LRIHMM	76.0	97.0	70.0	95.8	81.4	66.7
Discoverer	7.2	23.3	19.2	22.8	33.9	5.3
PI	100	100	20.0	16.7	35.6	33.3

表 6　协议关键词的召回率/%

系统	HTTP	FTP	SMTP	POP	SSDP	BitTorrent
LRIHMM	87.0	92.9	85.7	84.0	74.1	100
Discoverer	78.3	60.7	64.3	40.0	33.3	100
PI	4.4	3.6	7.1	4.0	18.5	50.0

从表 5 可看到，Discoverer 的准确率比 LRIHMM 要低得多。Discoverer 递归地将 token 序列聚类，然后在每一个子类中将相对频繁的 token 作为协议关键词。一些 token 在数据集里不是频繁项，但是被聚类后，在子类中就变成了频繁项，从而将过多的 token 判为关键词，导致假阳性过高，降低准确率。

在 PI 系统中，虽然 HTTP 和 FTP 的准确率高达 100%，但是它们的召回率太低，不足 5%。因为 PI 对实验数据本身的结构要求很高，即要求数据本身应该具有某种对齐性，如 ICMP 协议的报文，对应字段在不同的报文中出现的位置是一致的。但是对于 HTTP 和 FTP 这类的文本型协议而言，一些关键词在报文中出现的位置是可变的，因此，PI 对这类报文的处理效果较差。另外，还可以观察到，PI 的召回率太低，HTTP、FTP、SMTP 和 POP 的召回率不足 10%，这是因为 PI 系统挖掘的关键词个数太少，一般只有一个或几个，导致召回率过低。

如图 3 所示，LRIHMM 的 $F1$ 值比 Discoverer 和 PI 系统都要高。这意味着，LRIHMM 挖掘协议关键词的结果要比 2 个对比算法的结果要好得多。

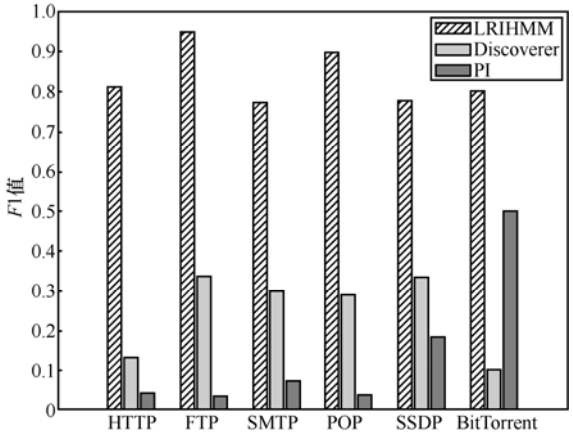

图 3　$F1$ 值对比

5.3.3 报文格式覆盖率

如图 4 所示，在 LRIHMM 中，HTTP、SSDP 和 BitTorrent 的报文格式覆盖率高达 100%。在 Discoverer 中，SSDP 和 BitTorrent 的报文格式覆盖率也为 100%，但是其他协议的报文格式覆盖率却比 LRIHMM 的要低。

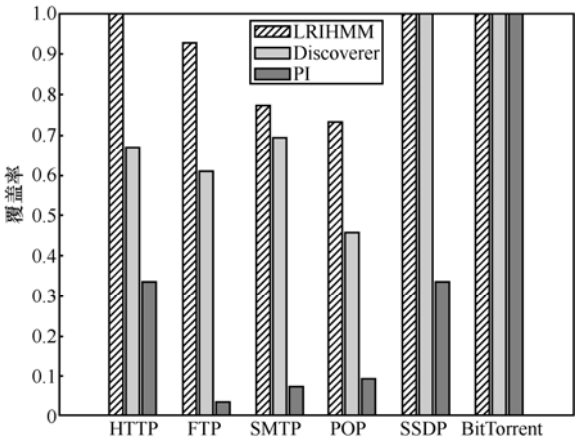

图 4　报文格式的覆盖率

5.3.4 复杂度分析

在每个时刻 t，更新 $\alpha_t(j,1)$ 的计算复杂度为 $O(M^2)$，更新 $\alpha_t(j,k)$ 的计算复杂度为 $O(MK)$，更新 $\alpha_t(j)$ 的计算复杂度为 $O(MK)$。因此，计算前向变量的总复杂度为 $O(M^2T+MKT)$。

在每个时刻 t，更新 $\beta_t(i,k)$ 的计算复杂度为 $O(MK)$，而更新 $\beta_t(i)$ 的计算复杂度为 $O(M^2)$，因此，更新前向变量的总计算复杂度为 $O(M^2T+MKT)$。

综上所述，LRIHMM 学习算法的总复杂度为 $O(M^2T+MKT)$。

6 结束语

本文提出一种新型的隐半马尔可夫模型（非齐次左—右型级联隐马尔可夫模型）。传统的隐半马尔可夫模型只能刻画不同状态之间的转移规律以及状态的持续时间长度分布规律。与传统的隐半马尔可夫模型不同，本文所提出的 LRIHMM 模型不但能刻画状态之间的转移规律，还能描述各状态的内部相位变化规律。

本文应用 LRIHMM 于协议逆向工程中，对应用层网络协议报文建模，并推断协议的报文格式。实验证明，LRIHMM 不但能描绘报文的字段跳转规律，还能揭示不同字段内部的性质（即左—右型马尔可夫性质）。基于最大似然概率准则可以确定协议关键词的长度，并推断协议关键词，最终可以重构协议的报文格式。与现有的相关工作对比可知，本文提出的方法具有很高的准确率、召回率和覆盖率，验证了本文所提出方法的有效性和准确性。

本文提出的基于最大似然概率的协议关键词长度确定方法，具有严密的数学基础，不管是理论分析还是实验结果都比前人工作中基于经验的关键词长度确定方法（例如基于最长公共子序列的方法）要合理得多。另外，与 PI 项目不同，本文提出的方法对协议关键词在报文中出现的顺序也没有特殊的要求，大大提高了报文格式的准确率。

参考文献：

[1] 赵咏, 姚秋林, 张志斌, 等. TPCAD: 一种文本类多协议特征自动发现方法[J]. 通信学报, 2009, 30(10A): 28-35.
ZHAO Y, YAO Q L, ZHANG Z B, et al. TPCAD: a text-oriented multi-protocol inference approach[J]. Journal on Communications, 2009, 30(10A):28-35.

[2] 张树壮, 罗浩, 方滨兴. 面向网络安全的正则表达式匹配技术[J]. 软件学报, 2011, 22(8): 1838-1854.
ZHANG S Z, LUO H, FANG B X. Regular expressions matching for network security[J]. Journal of Software, 2011, 22(8): 1838-1854.

[3] CABALLERO J, SONG D. Automatic protocol reverse-engineering: message format extraction and field semantics inference[J]. Computer Networks, 2013, 57(2): 451-474.

[4] TRIDGELL A. How samba was written[EB/OL]. Http://www.samba.org/ftp/tridge/misc/french_cafe.txt 2003.

[5] Pidgin[EB/OL]. http://www.pidgin.im/.2014.

[6] Rdesktop: a remote desktop protocol client[EB/OL]. http://www.rdesktop.org/.2014.

[7] KIM H, CHOI Y, LEE D. Efficient file fuzz testing using automated analysis of binary file format[J]. Journal of Systems Architecture, 2011, 57: 259-268.

[8] 李伟明, 张爱芳, 刘建财, 等. 网络协议的自动化模糊测试漏洞挖掘方[J]. 计算机学报, 2011, 34(2): 242-255.
LI W M, ZHANG A F, LIU J C, et al. An automatic network protocol fuzz testing and vulnerability discovering method[J]. Chinese Journal of Computers, 2011, 34(2): 242-255.

[9] IETF[EB/OL]. http://www.ietf.org/.2014.

[10] Internet2 netflow statistic [EB/OL]. http://netflow.internet2.edu, 2012.

[11] WEI X, GOMEZ L, NEAMTIU I, et al. ProfileDroid: multi-layer profiling of android applications[C]//18th Annual International Conference on Mobile Computing and Networking. ACM, c2012: 137-148.

[12] DAI S, TONGAONKAR A, WANG X, et al. Networkprofiler: towards automatic fingerprinting of android apps[C]//2013 Proceedings IEEE, INFOCOM. c2013.809-817.

[13] LEE S W, PARK J S, LEE H S, et al. A study on smart-phone traffic analysis[C]// IEEE Network Operations and Management Symposium (APNOMS), c2011: 1-7.

[14] FALAKI H, LYMBEROPOULOS D, MAHAJAN R, et al. A first look at traffic on smartphones[C]//10th ACM SIGCOMM Conference on Internet Measurement. ACM, c2010: 281-287.

[15] NARAYAN J, SHUKLA S K, CLANCY T C. A survey of automatic protocol reverse engineering tools[J]. ACM Computing Surveys, 2016, 48(3):1-26.

[16] BEDDOE M A. Network protocol analysis using bioinformatics

algorithms[EB/OL]. http://www.4tphi.net/~awalters/PI/PI.html, 2004.
[17] CUI W, KANNAN J, WANG H. Discoverer: automatic protocol reverse engineering from network traces[C]//16th USENIX Security Symposium on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association, c2007:1-14.
[18] WANG Y, YUN X, SHAFIQ M. A semantics aware approach to automated reverse engineering unknown protocols[C]// 20th IEEE International Conference on Network Protocols (ICNP). c2012: 1-10.
[19] ZHOU Z, ZHANG Z, LEE P. Toward unsupervised protocol feature word extraction[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(10): 1894-1906.
[20] ZHANG Z, ZHANG Z B, LEE P P, et al. ProWord: an unsupervised approach to protocol feature word extraction[C]//2014 Proceedings IEEE INFOCOM. c2014: 1393-1401.
[21] HE L, WEN Q, ZHANG Z. A TLV Structure semantic constraints based method for reverse engineering protocol packet formats[J]. Journal of Networking Technology, 2014, 5(1): 9.
[22] LI T, LIU Y, ZHANG C. A noise-tolerant system for protocol formats extraction from binary data[C]//2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA). c2014: 862-865.
[23] TAO S, YU H, LI Q. Bit-oriented format extraction approach for automatic binary protocol reverse engineering[J]. IET Communications, 2016,10(6):709-716.
[24] MENG F, LIU Y, ZHANG C. State reverse method for unknown binary protocol based on state-related fields[J]. Telecommunication Engineering, 2015, 55(4): 372-378.
[25] MENG F, LIU Y, ZHANG C. Inferring protocol state machine for binary communication protocol[C]//2014 IEEE Workshop on in Advanced Research and Technology in Industry Applications (WARTIA). c2014: 870-874.
[26] GASCON H, WRESSNEGGER C, YAMAGUCHI F. Pulsar: stateful black-box fuzzing of proprietary network protocols security and privacy in communication networks[M]//Springer International Publishing, 2015: 330-347.
[27] 肖明明, 余顺争. 基于文法推断的协议逆向工程[J]. 计算机研究与发展, 2013, 50(10):2044-2058.
XIAO M M, YU S Z. Protocol reverse engineering using grammatical inference[J]. Journal of Computer Research & Development, 2013, 50(10):2044-2058.
[28] 游翔, 葛卫丽. 飞信协议识别与多元通联关系提取方法[J]. 现代电子技术, 2014(21): 19-23.
YOU X, GE W L. Protocol identification and multi‐conversation relationship extraction in Fetion[J]. Modern Electronics Technique, 2014(21): 19-23.
[29] 岳旸, 孟凡治, 张春瑞, 等. 面向二进制数据帧的聚类系统[J]. 计算机应用研究, 2015(3): 909-916.
YUE Y, MENG F Z, ZHANG C R, et al. Cluster system for binary data frame[J]. Application Research of Computers, 2015(3): 909-916.
[30] 琚玉建, 谢绍斌, 张薇. 网络协议帧切分优化过程研究与仿真[J]. 计算机仿真, 2015(1): 318-321.
JU Y J, XIE S B, ZHANG W. Research and simulation of optimization process for network protocol frame segmentation[J]. Computer Simulation, 2015(1): 318-321.
[31] LI T, LIU Y, ZHANG C. A novel method for delimiting frames of unknown protocol[C]//2014 IEEE Workshop on Electronics, Computer and Applications. c2014:552-555.
[32] CABALLERO J, YIN H, LIANG Z. Polyglot: automatic extraction of protocol message format using dynamic binary analysis[C]//14th ACM Conference on Computer and Communications Security. New York, NY, USA, ACM, c2007: 317-329.
[33] CABALLERO J, POOSANKAM P, KREIBICH C. Dispatcher: enabling active botnet infiltration using automatic protocol reverse-engineering[C]//16th ACM Conference on Computer and Communications Security. New York, NY, USA, ACM, c2009: 621-634.
[34] CABALLERO J, SONG D. Automatic protocol reverse-engineering: Message format extraction and field semantics inference[J]. Computer Networks, 2013, 57(2): 451-474.
[35] ZHAO L, REN X, LIU M. Collaborative reversing of input formats and program data structures for security applications[J]. China Communications, 2014, 11(9): 135-147.
[36] LIN Z, ZHANG X, XU D. Reverse engineering input syntactic structure from program execution and its applications[J]. IEEE Transactions on Software Engineering, 2010, 36(5): 688-703.
[37] CUI B, WANG F, HAO Y. A taint based approach for automatic reverse engineering of gray-box file formats[J]. Soft Computing. 2015:1-16.
[38] WANG Z, JIANG X, CUI W. ReFormat: automatic reverse engineering of encrypted messages[M]. Berlin: Springer, 2009.
[39] ZHAO R, GU D, LI J. Automatic detection and analysis of encrypted messages in malware[J]. Information Security and Cryptology, 2014, 8567: 101-117.
[40] LIN W, FEI J, ZHU, Y. A method of multiple encryption and sectional encryption protocol reverse engineering[C]//2014 Tenth International Conference on Computational Intelligence and Security (CIS). c2014: 420-424.
[41] LI M, WANG Y, HUANG Z. Reverse analysis of secure communication protocol based on taint analysis[C]//2014 Communications Security Conference, c2014:1-8.
[42] 石小龙, 祝跃飞, 刘龙, 等. 加密通信协议的一种逆向分析方法[J]. 计算机应用研究, 2015(1): 214-221.
SHI X L, ZHU Y F, LIU L, et al. Method of encrypted protocol reverse engineering[J].Application Research of Computers, 2015(01): 214-221.
[43] JELINEK F. Continuous speech recognition by statistical methods[J]. Proceedings of the IEEE, 1976, 64: 532-556.
[44] BAKIS R, Continuous speech recognition via centisecond acoustic states[J]. The Journal of the Acoustical Society of America, 1976, 59(S1): 97.
[45] LUO J Z, YU S Z, Position-based automatic reverse engineering of network protocols[J]. Journal of Network and Computer Applications, 2013, 36(3): 1070-1077.
[46] YU S Z, Hidden semi-Markov models[J]. Artificial Intelligence, 2010, 174(2): 215-243.
[47] RABINER L. A tutorial on hidden Markov models and selected applications in speech recognition[J]. Proceedings of the IEEE, 1989, 77(2): 257-286.
[48] YU S Z, KOBAYASHI H. An efficient forward-backward algorithm for an explicit-duration hidden Markov model[J]. IEEE Signal Processing Letters, 2003. 10(1):11-14.

作者简介：

罗建桢（1984-），男，广东阳春人，博士，广东技术师范学院讲师，主要研究方向为协议逆向工程、未来网络。

余顺争（1958-），男，江西南昌人，博士，中山大学教授、博士生导师，主要研究方向为信息安全、信号处理、无线网络。

蔡君（1981-），男，湖南邵阳人，博士，广东技术师范学院副教授，主要研究方向为流量优化、未来网络。