doi:10.3969/j.issn.1000-436x.2013.04.015

自适应网络应用特征发现方法

王变琴[1,2]，余顺争[1]
(1. 中山大学 信息科学与技术学院，广东 广州 510006；2. 中山大学 教学实验中心，广东 广州 510006)

摘 要：提出了一种应用特征的自动提取方法。该方法首先从应用层载荷中提取关键词序列，然后通过负例子集合和冗余处理生成候选特征集，最后通过基于识别率的自适应机制选择应用特征。实验结果表明，该方法提取的应用特征能精确识别不同的协议。

关键词：应用特征；关键词挖掘；自适应特征提取；应用识别
中图分类号：TP393 **文献标识码**：B **文章编号**：1000-436X(2013)04-0127-11

Adaptive extraction method of network application signatures

WANG Bian-qin[1,2], YU Shun-zheng[1]
(1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China;
2. Education & Experiment Center, Sun Yat-sen University, Guangzhou 510006, China)

Abstract: An adaptive application signature extraction method (AdapSig) was proposed. AdapSig firstly extracted keyword sequences from payloads of packets, and then negative examples and redundancy filter were used to generate candidate signatures. An adaptive mechanism based on self-identification rate was applied to select the final si ture. Experimental results show that application signatures extracted by AdapSig have high accuracy for various protocols.

Key words: application signature; keyword minging; adaptive signature extraction; application identification

1 引言

网络流量的准确识别和分类是提供差异性服务质量（QoS, quality of service）保障、入侵检测（intrusion detection）、流量监控（traffic monitoring）及计费管理（accounting）等的基础。然而，面对网络应用的快速发展，传统的端口应用识别法[1]逐渐失效。基于流（flows）统计特征的流量分类方法[2,3]不能进行应用的精确识别。基于应用特征（signature）的识别方法[4,5]简单、准确度较高，在实际中被广泛应用，但是如何自动获取精确特征是该方法面临的主要问题。

现有方法主要通过查阅应用层协议标准文档寻找已知协议特征，或者利用 Wireshark、Tcpdump 等工具对网络上采集的包含应用层数据的分组进行分析及对比，提取应用协议特征。前者对于不公开文档及不断出现的新协议无能为力，后者虽然不需要协议规范文档，但其寻找特征的效率与可信度都较低，同时协议版本不断更新，新协议不断涌现的现实给这种人工或半人工分析方法带来巨大挑战。自然地，不依赖于具体协议，能够自动从应用层流量中获取应用协议特征的方法研究受到关注，并已有一些初步研究成果。Haffner 等人[6]利用有监督学习法构造应用识别特征，Ma 等人[7]采用无监督学习法构造协议特征模型，这些方法均生成模型特征（model signature），而非显式字符串特征（explicit string signature）。Byung-Chul 等人[8]提出一种基于最长公共字符串（LCS, longest common substring）

收稿日期：2012-01-06；修回日期：2012-06-14
基金项目：国家高技术研究发展计划（“863”计划）基金资助项目（2007AA01Z449）；国家自然科学基金—广东联合基金重点项目资助（U0735002）；国家自然科学基金资助项目（60970146）
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (2007AA01Z449); The Key Program of National Natural Science Foundation of China—Guangdong Joint Funds (U0735002); The National Natural Science Foundation of China (60970146)

的应用特征自动提取方法——LASER，该方法易造成漏报。Wang等人[9]在提取公共字符串的基础上，利用序列比对方法提取由公共字符串组成的子序列作为应用特征，这种方法对数据集要求较高。赵咏等人[10]提出一种基于语义信息的文本类多协议特征自动发现方法——TPCAD，其通用性有限。刘彬等人[11]利用经典 Apriori[12]算法从编码的报文载荷中提取特征。龙文等人[13]对 Apriori 算法进行改进，使其适合流量数据，相比原算法其效率明显提高，然而在支持度阈值较低时，其效率显著降低。Ye等人[14]提出一种应用特征自动生成系统-AutoSig，该方法利用子树（Subtree）结构构造包含一个公共字符串或多个公共字符串序列的特征，但没有考虑特征的专有性，以及它们之间可能存在的包含关系，易造成误报和冗余特征。

为了解决上述问题，本文提出一种新的应用特征发现方法，该方法在挖掘应用协议关键词的基础上，进一步发掘由其组成的应用特征。本文主要贡献有：1) 提出一种较为通用的特征发现方法，能够发现关键词或关键词序列构成的特征；2) 提出一种基于树结构的关键词提取算法，有效地提高了关键词的挖掘效率；3) 通过负例子、冗余过滤保证了特征识别应用协议的正确率，剔除了冗余特征；4) 自适应特征选择机制的设计，能够获得满足识别率的精简特征集，有效地节省人工资源，提高特征提取的自动化程度。

2 特征提取问题

定义 1 会话（session）是指一次通信建立和结束之间的所有报文构成的序列，记为 $s = m_1 m_2 m_3 \perp m_N$，其中，$m_i(1 \le i \le N)$为报文。在本文中它与 TCP 协议的连接、UDP 协议中的双向流的概念是等同的。相同会话的报文的五元组（SrcIP, DestIP, SrcPort, DestPort, Protocol）信息相同。待测应用协议的所有会话组成样本会话集（session set）S，记为 $S=\{s_1, s_2, \cdots, s_{J-1}, s_J\}$，其中，$s_j(1 \le j \le J)$为会话。

每个 session 中的 message 是由有序字节构成的字节序列（byte sequence），message 中连续字节构成一个字节串（或称字符串），记为 $bs = b_1 b_2 b_3 \perp b_L$，其中，$b_l(1 \le l \le L)$为单字节，$L$ 为字符串长度。本文约定可打印的字节或字符串用 ASCII 码格式表示，其他字节用十六进制（hexadecimal）格式表示，形式为“\x*ff*…”，其中，*ff* 表示 2 位十六进制数字。

协议关键词（keyword）其实就是满足一定条件（位置和频度）的字符串。同种应用协议的关键词组成关键词集，记为 $K=\{k_1, k_2, \cdots, k_{P-1}, k_P\}$，其中，$k_p(1 \le p \le P)$为关键词，它可以是协议的命令字、类型码、状态码、定界符等。

定义 2 应用特征（application signature）是指能够识别应用协议类型的关键词或关键词组合。从已知应用特征分析可知，特征有多种形式，可以是协议中的一个关键词，也可以是同时出现的多个关键词组成的序列。多个关键词可能出现在相同 message 中，称之为关键词组（keyword group）；不同 messages 中的关键词组成的序列称为关键词序列（keyword sequence）。

关键词作为协议常量是组成特征的基本元素，按其在 message 中出现的位置信息，可以将其划分成 2 类。

1) 固定偏移关键词，这种关键词出现在 message 的固定位置，它们常常是应用层协议报头中的特征串，包括协议名称、版本号等；也可以是应用层协议控制信息中的特征串，包括命令码、状态码，以及定界符等。一般情况下，这些关键词出现在 message 开始或最后的若干个字节处。

2) 非固定偏移关键词，和前者的唯一区别在于其偏移不是常量，而是变量。

通常选择具有固定偏移的关键词或其组合序列就可以构成识别应用协议的特征。本文从样本 session 集中自动提取具有固定偏移的关键词，进而发现识别应用协议的特征。

3 自适应特征发现方法

本文提出一种自适应特征提取方法 AdapSig (adaptive application signature extraction method)，其框架如图 1 所示。AdapSig 首先对 traces 文件进行预处理，生成样本 session 集；然后从样本 session 集中提取关键词集；再以关键词集作为候选集挖掘相同 message 中关键词构成的关键词组集；在其基础上，挖掘不同 messages 中的关键词构成的关键词序列集；通过后期负例子、冗余过滤消除其中专有性较弱或冗余的序列；最后，通过自适应机制选出满足识别率要求的应用特征。

3.1 数据预处理

按照会话的定义，将待测应用协议的 traces 文件处理成会话集。对于每个 session 中数据的选择，

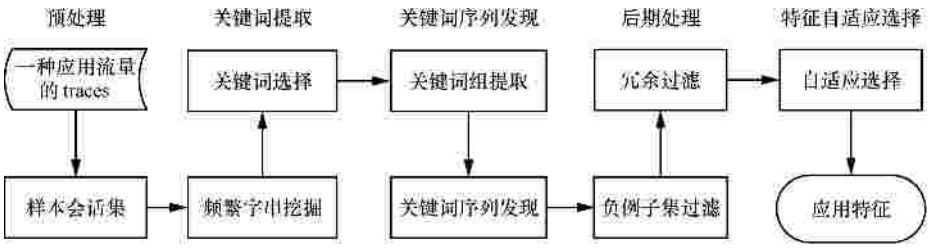

图 1　系统结构

常用的一种方法是仅选择 session（connection 或 flow）的前 n byte[6,14]。当前面的某个或某几个 messages 的数据内容较多时，这种选择可能会漏掉后面具有特征的 message。根据关键词的一般分布规律可以选择每个 session 中的所有或部分 messages，以及每个 message 开始的前 B_n byte 和最后的 L_k byte，如图 2 所示，这样可以提高获取关键词的准确性和效率。

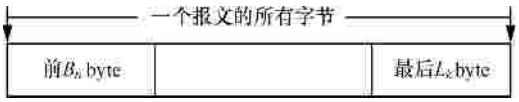

图 2　一个报文载荷数据的选择

3.2　关键词提取

关键词挖掘算法由频繁字符串挖掘和关键词选择 2 个过程组成。

3.2.1　频繁字符串挖掘

频繁字符串挖掘算法建立在如下一些相关概念和基础理论上。

定义 3　支持度（support）：给定 session 集 S，$s\in S$ 是其中的一个 session，如果 s 包含字符串 $string$，则令 $s(string)=1$，否则令 $s(string)=0$。称 $Sup(string)=\sum_{s\in S} s(string)/|S|$ 为 $string$ 在 S 上的支持度，其中，$|S|$ 为 S 中包含的 session 总数。

定义 4　频繁字符串（frequent byte string）：给定 session 集 S、字符串 $string$ 及最小支持度 $Minsup\in(0,1)$，当 $Sup(string)\geq Minsup$ 时，称该 $string$ 为 S 中的频繁字符串。S 中所有的频繁字符串组成频繁字符串集（frequent byte string set）F。

性质 1　如果 x 属于频繁字符串集 F，即 $x\in F$，则 x 的子串也属于 F；如果 x 不属于频繁字符串集 F，即 $x\notin F$，则 x 的所有超字符串（包含 x 的字符串）都不属于 F。

已有研究[15]证明 FP-Growth 算法是一种高效的频繁集挖掘算法，受其启发，结合网络流量字节有序的特点，定义一种新的树结构（称之为频繁字节树，FB-tree)来存储报文中的频繁字节序列，利用树结构挖掘频繁字符串，称之为基于 FB-tree 的频繁字符串挖掘（FreqBSminFB, frequent byte string mining based on FB-tree）方法，其基本思想为：首先按照偏移（字节距离报文首字节的位置，规定首字节的偏移量为零）逐次挖掘报文中不同位置上的频繁字节项；然后将每个报文中的位置频繁字节项的关联和频度信息压入到一棵 FB-tree 中；最后利用 FB-tree 挖掘频繁字符串集。

1) 1- 位置频繁字节挖掘

报文中不同位置上的相同字节值其意义可能不同。为了将不同位置的相同字节内容区分开，且同时使内容直接与位置相关联，在首次扫描会话集挖掘频繁字节时，按报文中字节的偏移逐次进行，分别挖掘不同位置上的频繁字节，其结果称为 1- 位置频繁字节集。

2) FB-tree 树的构造

根据性质 1，由于非频繁字节对挖掘频繁字符串没有贡献，仅需要根据报文中的频繁字节序列挖掘出所包含的频繁字符串，因此第 2 次扫描会话集，将每个报文中的非频繁字节用某个特殊值（例如 - 1）替换，转换报文为位置频繁字节序列，然后利用 FB-tree 记录会话集中每个报文的 1- 位置频繁字节。下面给出 FB-tree 的基本定义。

定义 5　FB-tree 是如下的一种树结构。

① 由 3 部分组成：标记为 $root$ 的根节点；作为根节点的子孙节点（sub-root）；存储叶节点指针的索引（index）。

② 除 $root$ 外，树中每一个节点可以表示成 4 元组：名称（$node\text{-}name$）、位置（$node\text{-}pos$）、支持计数（$node\text{-}sup$）、父节点指针（$node\text{-}parent$）。其中，名称记录该节点所代表的位置频繁字节的名字；位置记录该字节在报文中的偏移量；支持度计数记录包含该节点的会话数；父节点指针指向其父

节点。

③ Index记录树的各分支路径对应的叶节点指针，以方便后续对树的访问。

根据定义构造FB-tree：创建FB-tree的根节点，以*root*标记。对于*S*中转换过的报文，根据标记将每个报文分为一个或几个连续的字节串存储于数组*msg*中，调用函数*CreatTree*(*msg*, *root*)将数组*msg*的信息加入到FB-tree中。函数*CreatTree*(*msg*, *root*)完成的功能：如果*root*有子女使得*N.node-name*=*p.node-name*，并且*N.node-pos*=*p.node.pos*（其中，*p*为*msg*的第一个字节），则*N*的支持度计数增加1；否则创建一个新节点*N*，将其支持度计数设置为1，链接它到根节点*root*，并且将该节点作为该分支的叶节点。如果*msg*非空，递归地调用函数*CreatTree*(*msg*, *root*)。

3) FB-tree树的挖掘

会话集被压缩到FB-tree之后，其中，频繁字符串的挖掘就转换成对FB-tree进行挖掘。从FB-tree的定义及构建过程，可以得出以下一些重要性质。

性质2 在FB-tree中，除根节点外，同一条路径中的相邻2个节点*p*和*s*，若*p*是*s*的父节点，则有*p*的支持度不小于*s*的支持度，即$Sup(p)\geq Sup(s)$。

性质3 在FB-tree中，每条从根节点*root*开始到节点x_i的分支路径b：$root\rightarrow x_1\rightarrow\cdots\rightarrow x_i$，代表一个字符串$x=x_1,\cdots,x_i(1\leq i\leq n$，$n$为报文长度)，其支持度等于节点$x_i$的支持度，即$Sup(x)=Sup(x_i)$。

定理1 在FB-tree中，每条从根节点*root*开始到叶节点x_l的分支路径：$root\rightarrow x_1\rightarrow\cdots\rightarrow x_l$，代表的字符串$x=x_1,\cdots,x_l(1\leq l\leq n$，$n$为报文长度)，如果$x_l$的支持度计数$Sup(x_l)\geq Minsup$，则$x$为频繁字符串。

证明 由性质3，可知$Sup(x)=Sup(x_l)$。如果叶节点x_l的支持度计数$Sup(x_l)\geq Minsup$，则有$Sup(x)\geq Minsup$，故x为频繁字符串。

根据定理1，提出一种基于叶节点深度优先搜索的挖掘策略。从每条分支的叶节点开始挖掘其存储的频繁字符串，其大致过程为：①从FB-tree的叶节点开始，检查节点的支持度，如果某节点的支持度不小于*Minsup*，则将此节点至*root*的节点取出，组成一组频繁字符串；②对FB-tree的每个叶节点，重复步骤①直到遍历完FB-tree。

与基于Apriori算法[11]或其改进方法[13]相比，FreqBSminFB不需要生成候选集，以及对其候选项进行支持度计数，它直接利用树结构存储连续的字节序列，并同时进行支持度计数，有效地减少了对会话集的扫描次数，叶节点优先搜索的分支挖掘方式减少了对树节点的访问。

3.2.2 关键词选择

据性质1，获得的*F*中会有许多包含关系的频繁字符串，其中只有某些字符串是期待的关键词。为了获得期待的关键词，采用规则1和规则2对频繁字符串进行取舍。

规则1 若字符串$x\subset y$，并且$Sup(x)=Sup(y)$，即y出现的次数与x相等时，则认为所有的y出现时都包含x，选择y。

规则2 若字符串$x\subset y$，并且$Sup(x)>Sup(y)$时，按如下规则选择：1)设字符串长度阈值为l_{th}，如果x的长度$len(x)\leq l_{th}$(一般选取$l_{th}\leq 2$)，则剔除x；2)当$len(x)>l_{th}$，并且$Sup(y)>t\cdot Sup(x)$时，则保留y作为关键词，否则选择x。根据实验经验，一般情况下选择$t\geq 0.8$比较合适。

3.3 关键词序列发现

在挖出关键词之后，首先进行关键词组集挖掘，然后再进行候选特征集挖掘。在关键词集中，只有满足某种约束条件（即存在于同一个message中，并且互不重叠）的关键词才可以组成候选关键词组。关键词组的挖掘实际上同频繁字符串挖掘一样，是一种具有约束条件的序列挖掘。再以关键词组集K_g为候选集，挖掘同session中出现的关键词组成的序列集K_s。与频繁字符串和关键词组不同，关键词序列集的元素之间没有位置限制，可以采用任何已有的序列挖掘算法，实验中采用改进的AprioriAll[16]算法。若要获得较为完备的特征集，在提取关键词组和关键词序列集时，支持度阈值s_g和s_s均可以设为0。

3.4 后期过滤

应用特征应具有专有性（privacy）。一种协议的关键词序列可能会出现在其他应用协议中，例如，FTP协议的关键词序列“220”(包含一个关键词的序列）则同时出现在POP3、SMTP等协议中，因此，引入以下概念和过滤规则对关键词序列集进行过滤。

定义6 负例子集（negative example set）E是指来自于待测协议session之外的其他协议的sessions。实际中，尽可能选择与待测协议相近的或

者容易混淆的协议。例如，FTP、POP3 以及 SMTP 协议的数据集可以互为负例子集，因为它们的行为较为相似，存在共享关键词序列。

定义 7　负支持度（negative support）Sup_N 是指定关键词序列 k_s 在负例子集中获得的支持度，即 $Sup_N=|E(k_s)|/|E|$，其中，$|E|$为负例子集中的会话数量，$|E(k_s)|$为包含关键词序列 k_s 的负例子会话数量。

负例子过滤：对于关键词序列 $k_s\in K_s$，如果 $k_s\in E$，且 $Sup_N(k_s)>Sup_{N_{th}}$（$Sup_{N_{th}}$ 为负支持度阈值），则将 k_s 从 K_s 中删除。若要严格保证关键词序列的专有性，则令 $Sup_{N_{th}}=0$，即过滤出现在任何负例子集中的关键词序列。

关键词序列集中可能存在相互包含的元素，例如，在 FTP 协议的关键词序列集中同时包含“USER”和“USER、PASS”2 个元素，显然，“USER”的存在可能使“USER、PASS”在识别协议时失效。为了兼顾识别率和正确率，按照如下策略进行处理。

冗余过滤：若关键词序列 k_s' 是 k_s 的子序列，1) 当 $Sup(k_s')=Sup(k_s)$ 时，则剔除 k_s'；2) 当 $Sup(k_s')>Sup(k_s)$ 时，则剔除 k_s。即，对于相互包含的 2 个元素，当它们的支持度相同时，保留项数较多的元素，这样可以保证低误报率，提高识别正确率；当它们的支持度不同时，保留项数较少的元素，因为它们都满足作为特征的条件，但是项数较少的元素的支持度高，可以确保高识别率。

通过后期过滤，增强了特征识别的正确率，同时减少了特征集中不必要的冗余。过滤后的关键词序列称为候选特征集（candidate signature）C_s，即存在关系 $C_s\subseteq K_s$。

3.5　自适应特征选择

在 AdapSig 中，*Minsup* 值的设置会严重影响结果，它直接决定关键词的数量。然而，由于应用协议及其关键词分布的差异，*Minsup* 值的设置可能不同。尽管可以通过反复测试获其最优值，但此过程需要人工参与，并且耗时。为此，设计了一种自适应反馈调整机制使特征发现过程尽量自动化，减少人工参与。

定义 8　自识别率（self-identification rate）r 是指利用从 C_s 中选出的特征组成的精简特征集（minimum signature set）S_{min} 对目标应用协议的识别率，即 $r=|S(S_{min})|/|S|$，其中，$|S|$为待测应用协议的 session 数量，$|S(S_{min})|$为被 S_{min} 识别的 session 数量。

指定一个自识别率阈值 r_{th} 作为自适应算法的结束条件，如果 r_{th} 达到较高的标准，则认为得到的特征集是较为完备的，能够准确的识别待测应用协议流量，自适应选择算法终止。

精简特征集、候选特征集以及关键词序列集的关系为：$S_{min}\subseteq C_s\subseteq K_s$。关键词序列经过了负例子、冗余过滤处理，已经保证了其作为特征的专有性。自适应选择时，需要获得足够的特征数量来满足识别率要求。

采用图 3 所示的流程自适应选择特征，其步骤如下。

1) 参数初始化（$S_{min}=f$，$r=0$）。

2) 按照支持度从高到低进行选择，相同支持度的候选特征同时入选精简特征集 S_{min}。

3) 用精简特征集 S_{min} 测试会话集 S，获得对应的自识别率 r。

4) 若 $r>r_{th}$ 或者 C_s 为空集（取完所有的特征），则结束；否则，重复步骤 2)~4)。

有些协议由于数据集的缘故，即便其候选特征集为空时，也无法满足识别率要求，此时，$S_{min}=C_s$。

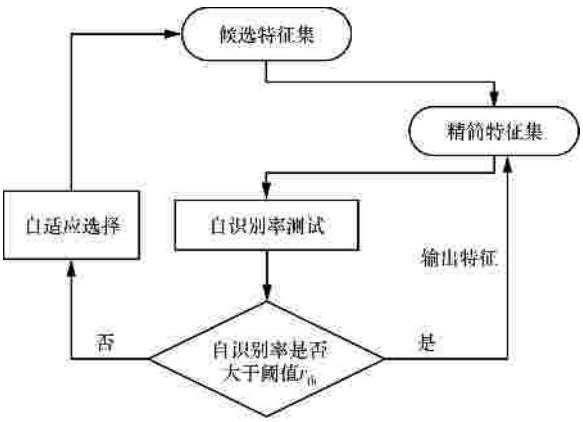

图 3　自适应选择机制

4　实验评估

实验环境为一台 PC 机（CPU：Intel Core 2 Duo E6550 2.33 GHz，内存：0.99GB），操作系统为 Windows，测试工具为 MATLAB 7.11.0。为了验证算法的性能，利用 Wireshark 在 CERNET 某小区网络出入口处采集包含载荷的真实流量，采集时间为 2009 年 1 月 8 日~2009 年 1 月 15 日。随机选择了 3 个不同时段（每个时段持续 5min）的数据，从中提取了目前流行的 6 种应用（HTTP、FTP、SMTP、POP3、MSN、BT）进行测试，如表 1 所示。将每

种应用的数据平均分成 4 组进行，其中，3 组用于特征提取（考虑测试机内存及耗时等因素，用其中 2 组数据分别进行特征提取，另一组用于 2 次特征提取时的自识别率测试，最后的特征是 2 次实验结果的综合）。另外一组数据作为测试集，用于特征评估。待测协议之外的其他协议会话集为其负例子。

表 1 样本会话集

协议	会话数量	分组数量	容量/Mbyte
HTTP	880	57 568	77.2
FTP	920	20 616	102.24
SMTP	652	32 884	60.32
POP3	512	600 072	72.8
MSN	556	52 460	18.52
BT	1 184	297 594	44.7

4.1 参数确定

本文的 AdapSig 的参数包括每个 session 的 message 数量 N，每个 message 的字节数 n，支持度阈值 $Minsup$、s_g、s_s，关键词选择参数 l_{th} 和 t，负支持度阈值 $Sup_{N_{th}}$，以及自识别率阈值 r_{th}。选择每个 session 中的所有 messages 以期获得完备特征集；理论上，可以选择每个 message 的前 $B_n$byte 和最后的 L_k byte，但对待测的已知协议分析发现，其特征一般分布在 message 开始的前 20byte 内，故选择 $n=B_n=20$；$Minsup$ 的选择会严重影响结果，如果 $Minsup$ 太小，结果会包含噪音；如果 $Minsup$ 太大，有可能漏掉有效的关键词。多次实验结果对比发现，当取 $Minsup$=0.02，适合所有测试 session 集。对比不同参数 len、t 值获取的关键词集，发现当 l_{th}=2、t=1 时，各种测试协议均取得较好的结果；设 s_g、s_s、$Sup_{N_{th}}$ 均为 0，期待获得完备特征集并严格保证特征的专有性；一般基于显式字符串特征的协议识别率都在 90%以上，故设 r_{th}=90%。

4.2 特征评估

自适应获得的精简特征集，如表 2 所示，为便于比较，表中也列出了其他方法 AutoSig[14] 和 L7-Filter[5] 获得的显式字符串特征，其中，AutoSig 与本文的 AdapSig 的思想较为接近，L7-Filter 的特征用正则表达式（RegExp, regular expression）表示（可打印字符或字符串均用小写表示，但在识别匹配时不区分大小写），主要靠人工分析获得，被认为是最精确的。

由表 2 知，本文的 AdapSig 获得的精简特征集中的特征与 L7-filter 的 RegExp 中的主要特征项（关键词）数量相当，明显少于 AutoSig 得到的特征。在内容上等同或接近 L7-filter 的 RegExp 中的主要特征项，只是 RegExp 更加精细，包含部分报文格式，而 AutoSig 中的特征普遍较长，多为包含多个元素的序列。

特征的质量用识别率和正确率评估。利用 TP（true positive）表示正确识别应用协议的 session 数量，FN（false negative）表示将该应用协议的 session 识别为其他协议的 session 数量，FP（false positive）表示将其他应用协议的 session 识别为该应用协议的 session 数量，则识别率和正确率定义如下。

定义 9 识别率（IR, identification rate）：指示应该正确识别的 session 中有多少被正确识别，可以

表 2 特征提取结果

协议	本文的 AdapSig 获得的精简特征	AutoSig 获得的特征	L7-filter 的特征(RegExp)
HTTP	“GET\x20/”|“HTTP/1.”	“HTTP/1.”|“GET\x20/,HTTP/1.”	“http/(0\.9|1\.0|1\.1)[1-5][0-9][0-9] [\x09-\x0d -~]*…|post[\x09-\x0d -~]*…” truncated
BT	“\x13BitTorrent\x20protocol”	“\x13BitTorrent\x20protocol, \x00\x00\x00\x00\x00”	“^(\x13bittorrent protocol|...” truncated
FTP	“220\x20,USER\x20\”| “USER\x20anonymous\x0d\x0a”	“QUIT\x20\x0d\x0a”|“PASS\x20”|“USER\x20”|“220\x20” |“USER\x20anonymous\x0d\x0a”|…(6signatures)	“^220[\x09-\x0d -~]*ftp”
POP3	“-ERR\x20”|“+OK\x20”	“USER\x20+PASS\x20”|“erver,+OK\x20”|...(7signatures)	^(\+ok |-err)
SMTP	“HELO”|“MAIL\x20FROM:” |“RCPT\x20TO:”|“DATA\x0d\x0a”| |“250\0x20,QUIT\x0d\x0a”	No test the application protocol	“^220[\x09-\x0d-~]*(e?smtp|simplemail)”
MSN	“VER\x20”|“MSG\x20”|“CVR\x20” |“USR\x20”	“ANS\x20,mail.com”|“mail.com,OUT\x0d\x0a”|“\x0d\x0aC” |”MSG\x20”|“BYE\x20,mail.com” |”ANS\x20,\x20OK\x0d\x0a”|…(7signatures)	“ver[0-9]+msnp[1-9][0-9]?[\x09-\x0d-~] *cvr0\x0d\x0a$|usr1[!-~]+[0-9]+\x0d\x0a$ |ans 1 [!-~]+[0-9]+\x0d\x0a$”

注：特征之间用符号“|”隔开，特征内的元素之间用逗号(“,”)隔开，不可打印的字节值用“\x*ff*”表示，其中，*ff* 为 2 个数字的十六进制值。

反映特征的完备程度，其计算公式为

$$IR = TP/(TP+FP) \times 100\% \quad (1)$$

定义 10　正确率（PR, precision rate）：指示识别结果中有多少是正确的，可以反映特征的区分度好坏，其计算公式为

$$PR = TP/(TP+FN) \times 100\% \quad (2)$$

由于网络流量识别要求在线实时环境，缓冲区读进的 message 数量会随着时间的增长而增加，因此测试过程模拟了读取不同 message 数量的识别效果，据此可以确定精确识别 session 所需要的 message 数量。对于本文的 AdapSig，选择每个 message 的前 20byte 进行匹配，其他 2 种方法 AutoSig 和 L7-filter 按照它们本身的要求，选择每个 message 的全部字节进行匹配。测试结果如图 4~图 9 所示。

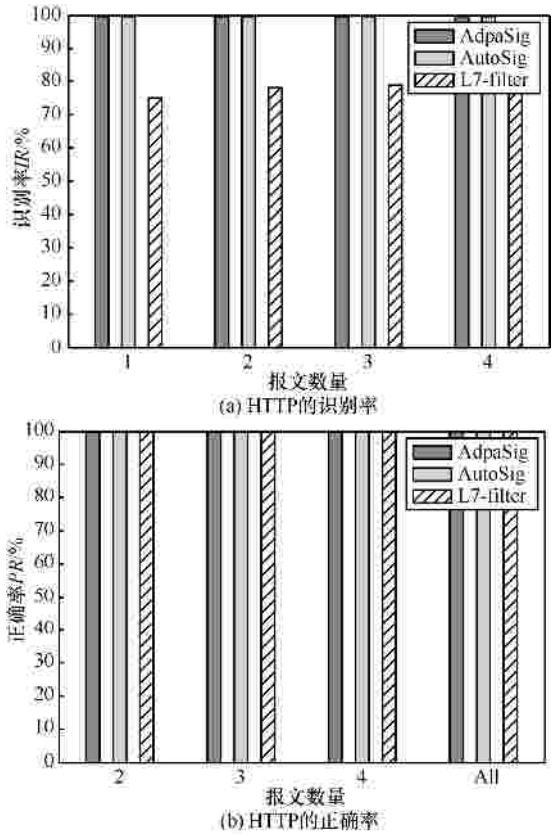

图 4　HTTP 协议的识别率和正确率

对于 HTTP 协议，AdapSig 和 AutoSig 的特征包含在 GET 请求 message 和响应 message 中，从第 2 个 message 开始就可以获得 99.49%的识别率，其漏报是因为极少数 sessions 中包含非 GET（而是 POST、PUT）请求 message，同时缺少响应 message。L7-filter 的 RegExp 以 HTTP 的响应 message 格式和 POST 请求 message 格式来进行匹配，但由于测试数据集中包含大量 GET 请求 message 而非 POST 请求 message，对于（分组丢失造成）没有响应 message 的 session 不能识别，其识别率低于 AutoSig 和本文的 AdapSig。3 种方法的正确率均为 100%。

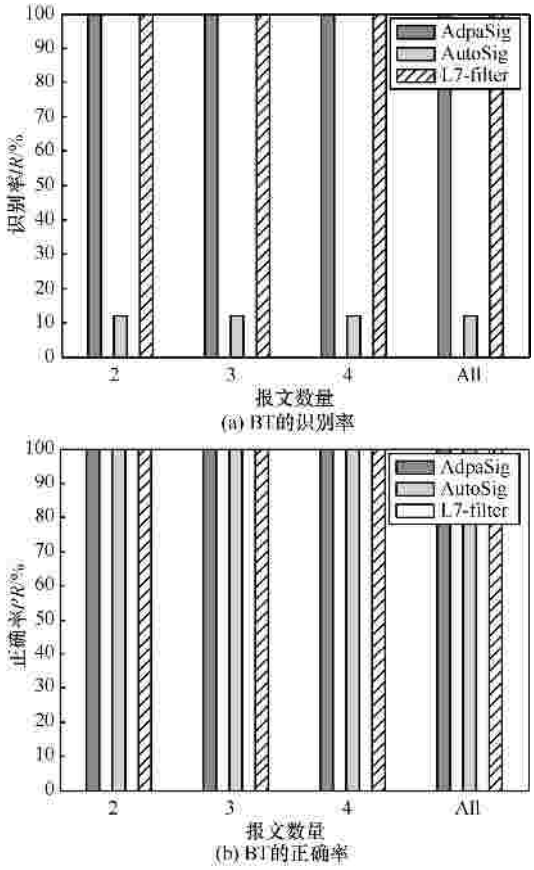

图 5　BT 协议的识别率和正确率

对于 BT 协议，本文的 AdapSig 获取的特征与 L7-filter 相同，该特征出现在 Peer 和 Peer 之间的握手 message，从 session 的第 1 个 message 开始就能获得 100%的识别率。而 AutoSig 的识别率很低。由于测试数据集中有很多 session 只包含字符串“\x13BitTorrent\x20protocol”，不包含字符串“\x00\x00\x00\x00\x00”。查阅 BT 规范获知，在“\x13BitTorrent\x20protocol”字段后确有一个 8 byte 的保留字段用于协议扩展，无扩展时全为，有扩展时从低位开始使用，因此，即便有零字符串出现，也未必一定是连续 5 个的“\x00\x00\x00\x00\x00”。3 种方法的正确率均为 100%。

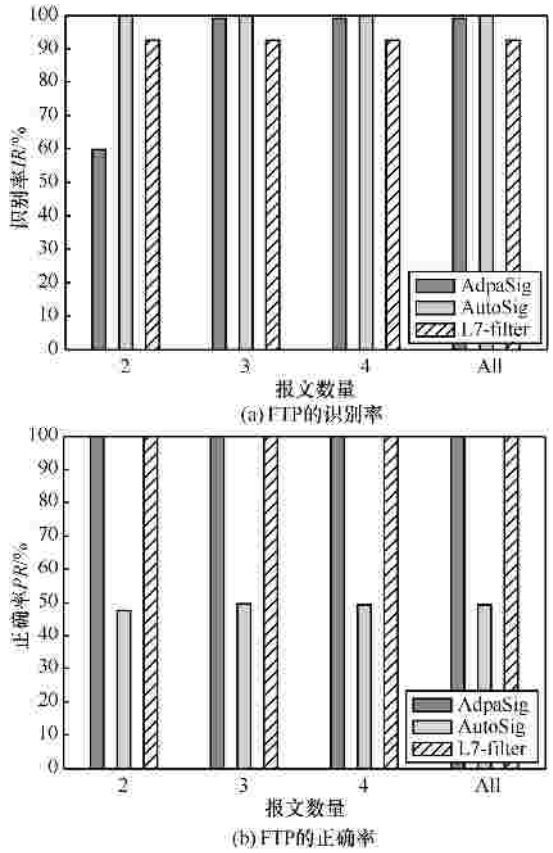

图 6 FTP 协议的识别率和正确率

FTP 协议精简特征集包含的 2 个特征分布在 session 的前 2 个 messages 中，并且其支持度较高，在 session 的第 2 个 message 开始就能获得 99.13% 的识别率，之后随 message 数量的增加识别率不再发生变化。AutoSig 虽具有 100%的识别率，但其特征“220\x20”、“QUIT\x20”也同时出现在 SMTP 中，造成将 35.58%的 SMTP 协议 session 识别为 FTP 协议，同样由于其特征“PASS\x20”、“USER\x20”、“QUIT\x20”及其序列也同时出现在 POP3 中，造成将 35.94%的 POP3 协议 session 识别为 FTP 协议，使得其正确率很低。对于 L7-filter，由于 FTP 服务的标题可以自由设定，字符串“220”后可以不带“FTP”字符串，因而造成部分 session 不能匹配，使识别率降低。本文的 AdapSig 和 L7-filter 的正确率均为 100%。

POP3 协议的精简特征集包含“-ERR\x20”和“+OK\x20”2 个特征，它们不同时出现在 session 的前 2 个 messages 中，在 session 的第 2 个 message 时就能获得 95.06%的识别率。之后随 message 数量的增加，并没有增加匹配的特征，识别率不再增加。AutoSig 缺失了“-ERR\x20”特征，同时将特征

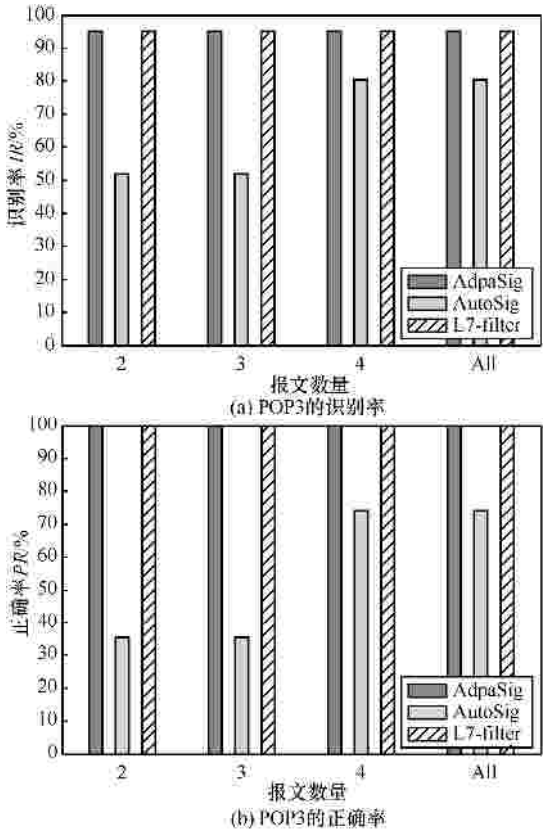

图 7 POP3 协议的识别率和正确率

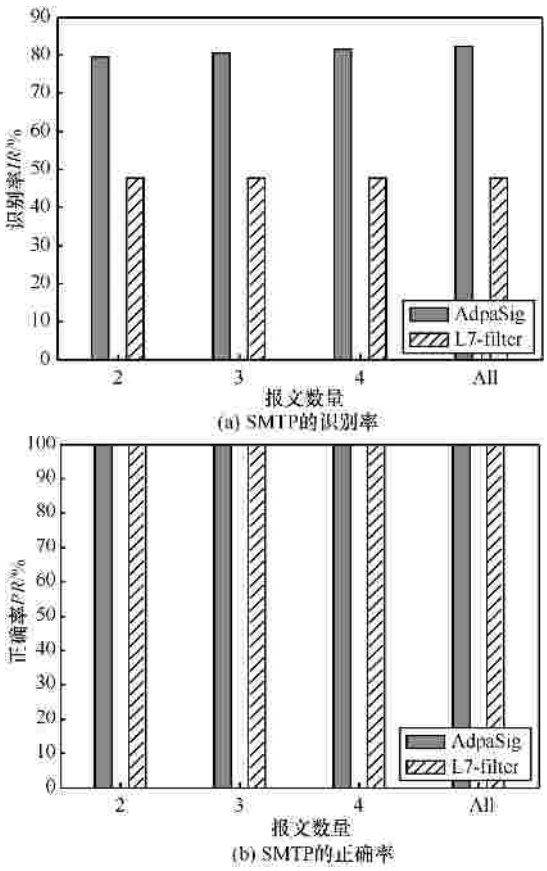

图 8 SMTP 协议的识别率和正确率

“+OK\x20”包在了其他的序列中，意味着要同时匹配其中的其他元素，导致漏报而影响识别率。此外，由于其特征“USER\x20,PASS\x20”也同时出现在 FTP 协议中，造成高误报，而 AdapSig 和 L7-filter 均未发现误报。

SMTP 协议的特征分布在多个 messages 中，并且它们的支持度都不高，因此识别率随着 message 数量的增加而提高。由于数据集的缘故（session 不完整，出现大量的分组丢失现象，缺乏包含特征的 message）使其不能获得 90%以上的识别率。AutoSig 没有测试该协议。L7-filter 的 RegExp 中的字符串“esmtp”、“smtp”或“simple mail”并非在每个 SMTP 协议 session 中都出现，故导致其识别率低于本文的 AdapSig。2 种方法的正确率均为 100%。

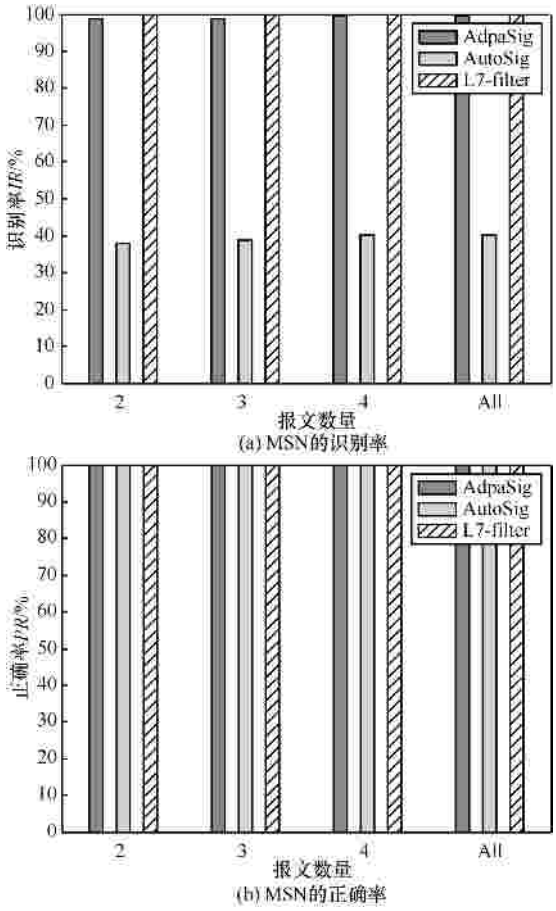

图 9　MSN 协议的识别率和正确率

MSN 协议的特征均出现在 session 的前 2 个 messages 中，在 session 的第 2 个 message 开始就可以获得 100%的识别率。L7-filter 的 RegExp 主要包含“ver”、“usr”和“ans”3 个特征与本文的 AdapSig 的结果类似，也获得 100%的识别率。AutoSig 获得的特征中缺少较为频繁的特征“VER\x20”和“USR\x20”，故其识别率较低。3 种方法的正确率均为 100%。

对于这 3 种方法，当 session 中的 message 偏移 N=4 时（不含建立 session 的协商 messages），2 种指标几乎都达到最佳值，之后不再有明显变化，说明多数协议在 session 的前 4 个 messages 中就可以提取到较为完备、准确的协议特征。此时，对于本文的 AdapSig，识别率均在 95.06%~100%（除 SMTP 协议只有 81.57%）之间，高于或等于 AutoSig 或 L7-filter 的识别率，表明获得的特征精确。本文的 AdapSig 的准确率与 L7-filter 方法相同均为 100%，表明经过负例子过滤的特征也具有很强的专有性，同样能够正确识别协议。而 AutoSig 对某些协议（FTP 协议和 POP3 协议）的正确率则较低，说明其部分特征的专有性不够。

4.3　效率评估

对于本文的 AdapSig 方法，关键词提取算法是核心。在 *Minsup* 相同的条件下，采用不同的挖掘算法可以获得相同的关键词结果，但其效率差异很大。为了评价关键词挖掘算法的效率（从读取样本会话集开始到输出关键词所耗费的时间），选择与本文最为相关的方法[13]测试比较，称其为 Cons_Apriori 算法。

图 10（2 组训练会话集上的平均耗时）比较了 FreqBSminFB 和 Cons_Apriori 算法在不同 *Minsup*(0.01~1)下的效率。从中看出，多数情况下，FreqBSminFB 的效率明显高于 Cons_Apriori，尤其是包含关键词数量较多的协议（如 FTP 和 MSN）更为显著，并且随着 *Minsup* 的降低，前者的运行时间表现为稳定略增，后者则表现为剧增。这是因为对于 Cons_Apriori 算法，当 *Minsup* 降低时，频繁字符串集的元素增加，下次迭代中的候选集的元数随之增加，导致扫描会话集 S 的次数剧增，其性能快速下降，表现为较为剧增的变化曲线。而对于 FreqBSminFB 算法，*Minsup* 的降低不会影响其对会话集 S 的扫描次数，只是造成树的节点增加，而且这种增加与 *Minsup* 降低所引入的节点增加是线性的，表现为较为平稳或缓慢增加的变化曲线。

5　结束语

本文提出的自适应特征提取方法能够自动发现关键词序列特征，并能够根据需要自动从候选特

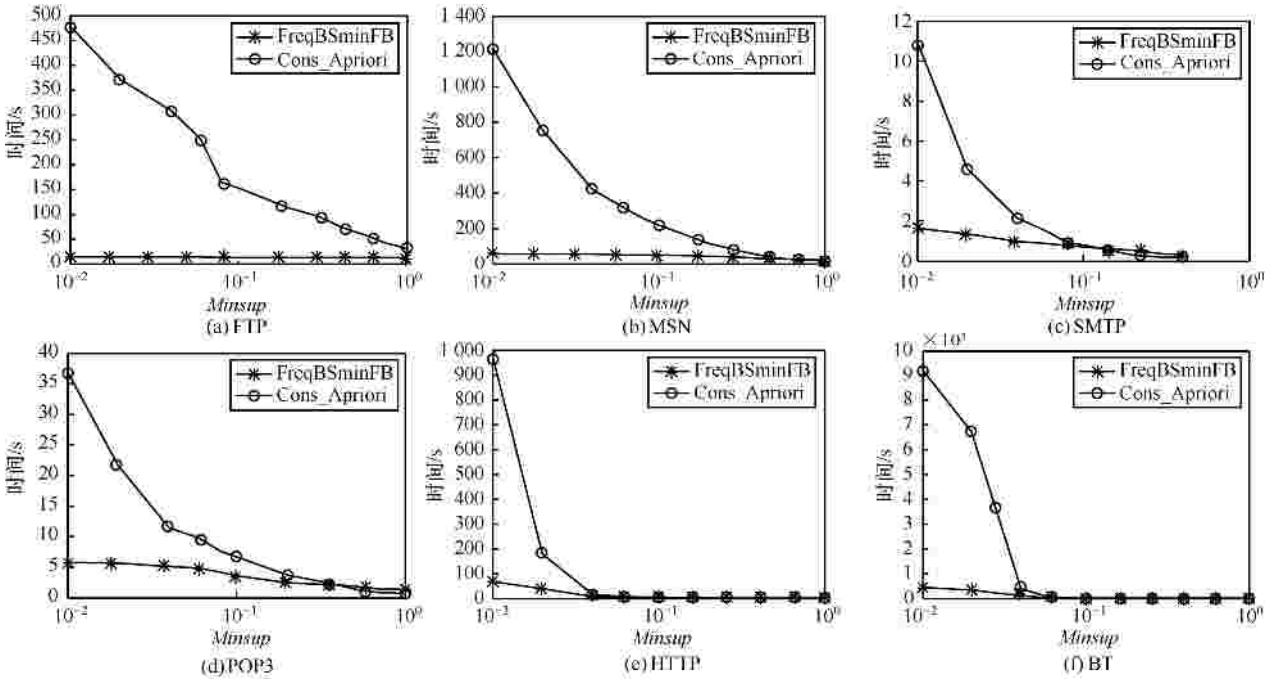

图 10 FreqBSminFB 和 Cons_Apriori 效率比较

征集中选出满足需要的组成精简特征集，减少了人工参与，提高了方法的自动化程度。并且选择目前较为流行的已知应用的流量进行了有效性测试，取得不错的效果。然而这种方法是否适合其他应用类型，尤其像增长较快的网络视频、音频以及 P2P 类的其他应用类型还需要进行广泛的验证。此外，如何与实际的应用识别系统结合，例如，与 L7-Filter 结合，将应用特征自动转换成需要的格式（如正则表达式），实现应用特征库的自动扩充与更新是未来的研究工作。

参考文献：

[1] Internet assigned numbers authority (IANA) [EB/OL]. http://www.iana.org/assignments/port-numbers, 2012.

[2] BERNAILLE L, TEIXEIRA R, AKODKENKUT I, *et al*. Traffic classification on the fly[J]. ACM SIGCOMM Computer Communication Review, 2006, 36(2):23-26.

[3] CROTTI M, DUSI M, GRINFOLI F, *et al*. Traffic classification through simple statistical fingerprinting[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(1):1-16.

[4] SEN S, SPATSCHECK O, WANG DM. Accurate, scalable in-network identification of P2P traffic using application signat [A]. Proceedings of WWW2004[C]. New York, USA, 2004. 512-521.

[5] Application layer packet classifier for linux[EB/OL]. ttp://17-filter. Sourceforge.net, 2012.

[6] HAFFNER P, SEN S, SPATSCHECK O, *et al*. ACAS: automated construction of application signatures[A]. Proceedings of ACM SIGCOMM MineNet Workshop[C]. Philadelphia, USA, 2005.197-202.

[7] MA J, LEVCHENKO K, KREIBICH C, *et al*. Unexpected means of protocol inference[A]. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement[C]. Rio de Janeriro, Brazil, 2006. 313-326.

[8] PARK B C, WON Y J, KIM M S, *et al*. Towards automated application signature generation for traffic identification[A]. Proceedings of the IEEE/IFIP Network Operations and Management Symposium[C]. Salvador, Bahia, Brazil, 2008. 160-167.

[9] WANG Y, XIANG Y, ZHOU W L, *et al*. Generating regular expression signatures for network traffic classification in trusted network management[J]. Journal of Network and Computer Applications, 2011, 17: 1-9.

[10] **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.

[11] **刘兴彬，杨建华，谢高岗等**. **基于** Apriori **算法的流量识别特征自动提取方法**[J]. **通信学报**, 2008, 29(12):51-59.
LIU X B, YANG J H, XIE G G, *et al*. Automated mining of packet signatures for traffic identification at application layer with Apriori

algorithm[J]. Journal on Communications, 2008, 29(12):51-59.

[12] AGRAWAL R, SRIKANT R. Fast algorithms for mining assoc ion rules[A]. Proceedings of the 20th International Conference on Very Large Data Bases[C]. San Francisco, CA, USA, 1994.487-499.

[13] **龙文，马坤，辛阳等. 适用于协议特征提取的关联规则改进算法**[J]. **电子科技大学学报**, 2010, 39(2):302-305.
LONG W, MA K, XIN Y, *et al*. Improved association rules algorithm for protocol signatures extracting[J]. Journal of University of Electronic Science and Technology of China, 2010, 39(2):302-305.

[14] YE M J, XU K, WU J P, *et al*. AutoSig-automatically generating signatures for applications[A]. IEEE Ninth International Conference on Computer and Information Technology[C]. Xiamen, China, 2009. 104-109.

[15] HAN J, PEI J, YIN Y. Mining frequent patterns without ndidate generation[A]. Proceedings of the 2000 ACM SIGMOD[C]. Dalas, TX,USA, 2000.1-11.

[16] AGRAWAL R, SRIKANT R. Mining sequential patterns[A]. Proceedings of the Eleventh International Conference on Data Engineering[C]. 1995.3-14.

作者简介：

王变琴（1963-），女，陕西蒲城人，博士，中山大学高级工程师，主要研究方向为网络安全与数据挖掘。

余顺争（1958-），男，江西景德镇人，博士，中山大学教授、博士生导师，主要研究方向为计算机网络与网络安全。

（上接第 126 页）

[11] ZHANG P, DENG G M, ZHAO Q. An automatic experimental platform for differential electromagnetic analysis on cryptographic ICs[A]. Proceedings of the Second International Symposium on T t Automation & Instrumentation-ISTAI 2008[C]. Beijing, China, 2008. 1078-1082.

作者简介：

张鹏（1976-），男，湖北罗田人，信息保障技术重点实验室博士后，主要研究方向为芯片安全防护技术。

王新成（1969-），男，湖南娄底人，博士，信息保障技术重点实验室高级工程师，主要研究方向为集成电路芯片设计。

周庆（1964-），男，江苏泰兴人，信息保障技术重点实验室高级工程师，主要研究方向为信息安全技术。