# 基于流的加密流量识别方法

陈玉祥 [1, 2]，程光 [1, 2]

（1.东南大学计算机科学与工程学院，南京，210096；
2.教育部计算机网络与信息集成重点实验室（东南大学），南京，210096）

**摘　要：**加密流量识别是当前网络管理、流量分析和异常识别的基础，为此，本文提出一种加密流量识别方法。该方法首先对网络数据包按照五元组进行组流，然后将数据包 TCP 报头上层的数据作为有效负载，分别计算整条流负载的 Monte Carlo PI 估计值以及负载前 1K 的相对熵，然后使用特征选择算法得到特征向量，根据机器学习方法的分类结果从而得出最终的加密流量识别结果。实验表明，该方法识别准确率较高且误判率较低，并能实现实时识别和处理。

**关键词：**加密流量识别；Monte Carlo PI 估计值；相对熵；SVM 分类器

# Encrypted Traffic Identification Approach Based on Data Stream

Chen Yuxiang[1, 2], Cheng Guang[1, 2]

(1. School of Computer Science and Engineering, Southeast University, Nanjing 210096;
2. Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096)

**Abstract:** Encrypted traffic identification is a hot topic of current network management, traffic analysis and anomaly recognition. Based on this, we propose a method for encrypted traffic identification in the paper. We firstly combine network data packets in accordance with five-tuple rule, then treat the data upon TCP packet header of every flow as the payload and calculate the Monte Carlo PI estimates of every entire flow as well as the relative entropy of the first 1K of the payload. After this, apply the feature selection algorithm to obtain corresponding feature vector. According to the classification results of the machine learning method SVM, we gain the final encrypted traffic identification results. Experimental results show that the approach can achieve higher correct identification rate, real-time identification and processing with a low false positive rate.

**Key words:** Encrypted traffic identification; Monte Carlo PI estimates; Relative entropy; SVM classfier

## 0 引言

近年来，随着网络技术的高速发展，越来越多的公司、机构、政府加快发展自身在网络上的应用和服务，而网络技术在推动产品创新和变革的同时，网络安全问题也得到了人们越来越多的关注。网络通信加密技术是保障网络数据安全性的重要手段，保证了在网络上所传输的数据不被第三方窃取。而加密通道严重威胁网络信息安全，恶意软件通过加密通道技术绕过防火墙和入侵识别系统[1]将机密信息发送到外网，如僵尸网络[2]、木马和高级持续性威胁（APT）[3]。因此，加密流量的有效识别和检测对维护用户权益，保障网络安全运行有着重要意义。

本文提出一种加密流量识别方法。该方法首先对网络数据包按照五元组（源地址，源端口，协议号，宿地址，宿端口）并根据 FIN、RST 或超时结束等标识进行组流，然后将整条流中的数据包 TCP 报头上层的数据作为有效负载，分别计算每条流负载的 Monte Carlo PI 估计值以及负载前 1K 的相对熵，然后使用特征选择算法得到特征向量，根据机器学习方法的分类结果从而得出最终的加密流量识别结果。其中，训练分类器的过程中使用的是有监督的 SVM 方法，输入是有标签的流的特征向量，

**作者简介：**陈玉祥，（1990-），男，硕士研究生，E-mail: yxchen@njnet.edu.cn；程光，（1973-），男，教授，E-mail: gcheng@njnet.edu.cn

输出是相应的分类器模型。

本文安排如下：第二节介绍近年来国内外主要的加密流量识别研究的相关现状；第三节介绍了基于流的加密流量识别方法；第四节介绍加密流量识别方法的实验结果并进行分析；最后是对全文进行总结。

# 1 相关研究

广义上来说，加密流量是由加密算法生成的流量。实际上，加密流量主要是指在通信过程中所传送的被加密过的实际明文内容。若用明文 HTTP 协议下载一个加密文件，这种流量不能作为加密流量，因为协议本身是不加密的。

加密流量识别技术已引起国内外研究人员的极大关注。文献[4]提出了一种加密流量实时识别方法 RT-ETD 来识别加密与未加密流量。分类器能够实时识别是因为只需要处理每条流中的第一个数据包，通过第一个数据包的有效载荷的熵估计进行识别。实验结果表明该方法识别准确率超过 94%，可以预识别高带宽网络中的加密流量。文献[5]提出一种基于加权累积和检验的加密流量盲识别方法，该方法利用加密流量的随机性，对负载进行累积和检验，根据报文长度加权综合，最终实现在线普适识别。该方法无需解密操作，也无需匹配特定内容，实现了对加密流量的普适识别。文献[6]提出一种通过两个互补方法来实时分类 Skype 流量的框架。虽然数据分组的整个数据流被加密，但格式数据分组中的信息协议报头，以测试统计分析 Pearson 的申请协议报头 4 个字节中的卡方，仍然可以发现差异。然后通过协议数据流的已知和未知的 Skype 的头可以有效地匹配，以确定具体的协议数据流。文献[7]在分析流量加密导致特征变化的基础上提出一种特征估计方法 EFM，采用 PPTP 和 IPsec 两种隧道技术加密流量，通过计算未加密流量与加密流量的相关性从 49 种特征中选取 29 种未加密流量与加密流量强相关的特征，并根据相关性特征采用机器学习分类方法分类加密与未加密混合流量，并比较三种不同机器学习算法（SVM, NB 核估计和 C4.5）的性能。文献[8]使用多种监督学习分类方法（如 C4.5，adaboost，GP，SVM，RIPPER，Naïve Bayes）来识别 SSH 和非 SSH，以及 Skype 和非 Skype，该方法无需端口号，IP 地址和有效载荷，可以很好的适用于不同网络环境，也可以用于其他加密流量的分类。文献[9-12]也都是采用机器学习的方法对流量进行识别分类。

流量识别的前提是针对不同的应用或协议有明显的区分特征，加密流量识别和未加密流量识别的本质区别在于：加密使得得用于区分的特征发生了改变。流量加密后的变化可以概括如下：首先，IP 报文的明文内容更改为密文。第二，流量加密后有效载荷的统计特征（如随机性或熵）发生改变。第三，流量加密后流统计特性发生改变，如包长度，间隔时间和包数。而本文所提出的方法正是利用流量加密后有效载荷的混乱性和随机性发生变化，从而用来对加密流量进行识别。

# 2 基于流的加密流量识别方法

## 2.1 算法描述

熵理论目前被广泛应用于信息安全领域的数据分析和异常检测[13]。“熵”是用来表示能量分布均匀程度的术语，能量分布越均匀，熵就越大[14]。香农提出的“信息熵”概念解决了对信息的量化度量问题，他指出信息熵描述的是信源的不确定性，能有效地表现出同一属性上对应数据的集中和分散情况。

在大多数网络通信应用中，网络会话中出现的字符是有统计规律的（见表 1），常见字符出现的频率高，非常见字符出现的频率低，如大小写英文字母和阿拉伯数字出现频率高。如果对网络连接进行加密，常见字符和非常见字符都成了乱码，出现的频率就趋于相等。

表 1 英文字母出现概率

| 字符 | 出现概率 | 字符 | 出现概率 |
|---|---|---|---|
| A | 0.0788 | N | 0.0706 |
| B | 0.0156 | O | 0.0776 |
| C | 0.0268 | P | 0.0186 |
| D | 0.0389 | Q | 0.0009 |
| E | 0.1268 | R | 0.0594 |
| F | 0.256 | S | 0.0634 |
| G | 0.0187 | T | 0.0978 |
| H | 0.0573 | U | 0.0280 |
| I | 0.0707 | V | 0.0102 |
| J | 0.0010 | W | 0.0214 |

（接上表）

| 字符 | 出现概率 | 字符 | 出现概率 |
|---|---|---|---|
| K | 0.0060 | X | 0.0016 |
| L | 0.0394 | Y | 0.0202 |
| M | 0.0244 | Z | 0.0006 |

广义上来说，熵计算有两种方法。其一是信息论中的香农熵，另一个则是 Tsallis 熵（标准 Boltzmann-Glibbs 熵的归一值）[15]。本文中使用了传统的香农熵进行流负载的熵值计算：以一字节（8 比特）作为一个码元符号并逐字节得到字符熵。对常见文件类型文件以及加密流量包有效负载的字符熵值（使用 ENT 开源工具进行计算）进行了统计，结果如图 1 所示（其中，pcap 格式文件内容是加密流量包的有效负载；压缩文件中，考虑了两种常见的压缩格式：1-15 号为.rar 形式，16-30 号为.zip 形式）。

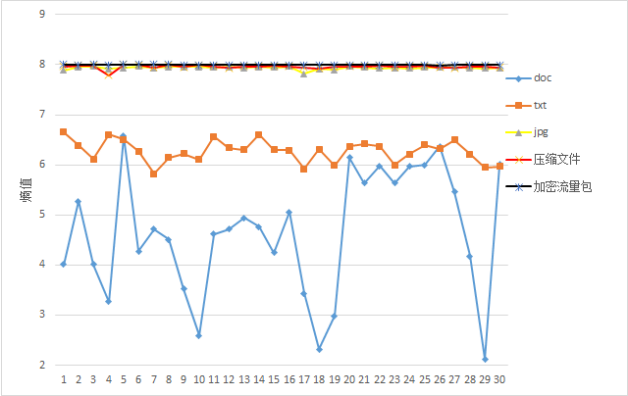



图 1 各种文件类型离线计算熵值图

由图可知，doc、txt 文件熵值略小，jpg（属压缩文件范畴）、压缩文件、加密流量负载熵值较大（趋近于 8）。基于字符熵以及传统文件中字符出现频率统计规律的存在，基于信息熵的方法能够有效地将低熵值数据流和高熵值数据流区分开。但是，单纯地使用这个方法，存在不足：从字符熵的大小上来看，jpg、压缩文件和加密流量几乎相等。基于信息熵的方法并不能将压缩文件和加密流量很好地区分开来。

数据压缩[16]的原理就是找出那些重复出现的字符串，然后用更短的符号代替，从而达到缩短字符串的目的。本质上，所谓“压缩”就是找出文件内容的概率分布，将那些出现概率高的部分代替成更短的形式。所以，内容越是重复的文件，就可以压缩地越小。相应地，如果内容毫无重复，就很难压缩。极端情况就是，遇到那些均匀分布的随机字符串，往往连一个字符都压缩不了。所以压缩过后的文件中字符偏向于均均匀分布，从而熵值较大，在这一点上与数据加密类似。

通常，Monte Carlo PI 估计法能够将加密文件和压缩文件进行有效区分。Monte Carlo 的基本思想是：当所要求解的问题是某种事件出现的概率或者是某个随机变量的期望时，可以通过某种实验的方法得到该事件的概率，或者这个随机变量的平均值，并用它们作为问题的解。在本文中，Monte Carlo PI 估计法通常是用来从给定的随机坐标（x,y）集中来估计 PI 的值。越均匀分布的数据点集所得到的 PI 估计值越接近其真实值，越准确的估计值表明其数据点集的随机性越强，从而可以根据 Monte Carlo PI 估计误差来表征数据集的随机性大小。对加密流量包有效负载、jpg 文件和压缩文件的 Monte Carlo PI 估计误差进行了统计，结果如图 2 所示。

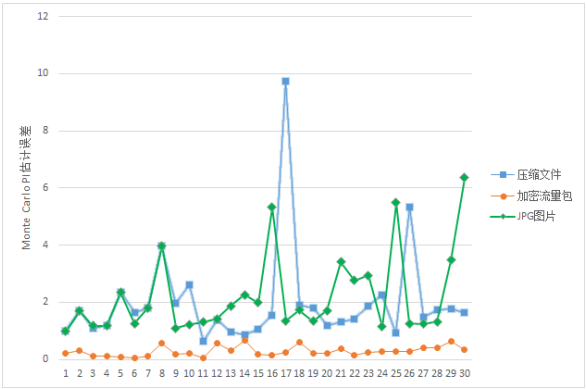



图 2 三种文件类型的 Monte Carlo PI 估计误差图

由图可知，压缩文件和 jpg 文件的 Monte Carlo PI 估计误差较大，而加密流量包有效负载的则比较小（在 1%以内），从而基于 Monte Carlo PI 估计误差值能够将加密流量和 jpg 以及压缩文件流量区分开来。究其原因：根据文件压缩的原理，从局部来看，字符的出现规律会体现为一定的随机性，但是需要注意的是，文件在压缩前字符出现的概率仍是符合上述的统计规律的，从而压缩后从整体上来看体现出来的是一种伪随机性，并不会像加密流量那样表现出较高的随机性。

为此，在以上统计分析的基础上，可以根据这两个特征尺度将加密流量从流量数据中识别出来，

但是考虑到在实际应用场景中，对每一条流的有效负载进行字符熵值的实时计算是不现实的，所以有必要对计算对象进行约简，并从多维度来考量计算对象的熵值特征,从而在牺牲一定识别率的情况下，保证系统较高的运行效率（时间上和空间上）。

下面引入相对熵，并使用特征选择算法选择特定特征子集构成特征向量，然后使用机器学习方法 SVM 算法进行分类器模型的学习，之后再用得到的分类器对已接收的流量进行加密与否的判别。

### 2.2 Monte Carlo PI 估计误差

针对 Monte Carlo PI 估计误差的计算，本文借鉴了 ENT 开源工具中的计算方法。算法的伪代码如表 2 所示。

表 2 Monte Carlo PI 估计误差算法

Monte Carlo PI 估计误差算法

(1) 输入：数据流 Flow

(2) 输出：Monte Carlo PI 估计误差

(3) 伪代码：

```
Flow.mcount=0; Flow.mp=0; Flow.inmount=0;
Flow.monte[MONTEN];  // MONTEN=6
for( i=0; i<Flow.packetnum;++i )
①{
   for( j=0; j<Flow.packet[i].bytenum;++j )
   ②{
      Flow.monte[Flow.mp++]=Flow.packet[i].byte[j];
      if(Flow.mp>=MONTEN)
      ③{
          Flow.mp = 0; Flow.mcount++;
          for (mj = 0; mj < MONTEN / 2; mj++)
          ④{
montex = (montex * 256.0) + tcp_pool[pf].monte[mj];
montey = (montey * 256.0) + tcp_pool[pf].monte[(MONTEN /
2) + mj];
            }④
if ((montex * montex + montey *  montey) <= INCIRC)
    tcp_pool[pf].inmont++;
        }③
     }②
   }①
Flow. MontePIerror = 100.0 * (fabs(PI - 4.0 * Flow.inmont)
/ Flow.mcount)) / PI);
```

### 2.3 相对熵

针对文件 F，文件中的每一个字节都可以当作集合 S 中的一个元素 $S_i$，这样便可以得到文件中所有字节的熵。更一般地，可以将文件 F 中任意的 K 个连续字节当作一个元素，并计算给定文件中所有 K 个连续字节所组成的新集合 S'的熵值。

下面定义 $f_k$ 代表所有的 K 个连续的字节所组成的集合，$h_k$ 为该集合所对应的的相对熵，则相对熵的计算有：

$$h_k = -\frac{\sum_{i=1}^{|f_k|}\frac{m_{ik}}{m-k+1}log_2\left(\frac{m_{ik}}{m-k+1}\right)}{log_2(m-k+1)}$$

$$= \frac{1}{(m-k+1)log_2(m-k+1)}[\sum_i m_{ik}log_2(m-k+1) - \sum_i m_{ik}log_2 m_{ik}]$$

$$= 1 - \frac{1}{(m-k+1)log_2(m-k+1)}\sum_i m_{ik}log_2 m_{ik} \qquad (1)$$

在（1）式中，$m_{ik}$代表$f_k$集合中第 i 个元素出现的频数，其中$\sum_{i=1}^{|f_k|} m_{ik} = m-k+1$。在本方法中，令文件 F 的大小为 m 字节，则对于任意的 i（$i \in [1,m]$），$h_i$的值便可作为该文件的一个特征。因此，对一个大小为 m 的文件，定义它的相对熵向量为$\{h_1, h_2, \cdots h_n\}$。此外，通过将一个字节分成高 4 位和低 4 位这两个部分，并将其作为一个独立的元素，以类似的方法进行计算从而我们得到$h_0$。

### 2.4 特征选择

如前所述，对一个特定的文件，相应的特征有 h1，h2，……hn 等，但是在实际应用中，可能存在不相关的特征，特征之间也可能存在相互依赖。此外，考虑到实时加密流量识别应用场景的需要，需要进行特征选择，剔除冗余特征，在保证所构建出来的分类器具有比较好的识别效果的同时，减少运行时间，提高系统识别效率。

已有文献[17],[18]等已经介绍了减少特征量、降低特征空间的特征选择算法。本文所选用的特征选择算法是最简单但比较有效的序列前向选择算法（SFS，Sequential Forward Selection），该方法可以有效地提高分类性能。该算法的特征子集 X 从空集开始，每次选择一个特征 x 加入特征子集 X，使得特征函数 J(X)最优。简单说就是，每次都选择一个

使得评价函数的取值达到最优的特征加入，其实就是一种简单的贪心算法。我们使用 SVM 算法并选择不同的特征子集来构造分类器，从而尽量保证最大化识别的准确率。重复此过程，直到所选择的特征子集构造的分类器的识别效果达到较佳状态。最终，通过 SFS 算法进行特征选择后的特征子空间为 $\emptyset_{svm}=\{h_0,h_1,h_2,h_3,p_{\text{error}}\}$，其中$p_{\text{error}}$代表 Monte Carlo PI 估计误差值。

### 2.5 缓存大小选择

如之前整条流负载字符熵的计算，相对熵特征向量的计算也应该在整条流的负载上进行。但是考虑到在实际应用场景中，对每一条流的有效负载进行相对熵向量的实时计算是不现实的，所以有必要对计算对象进行约简，从而在牺牲一定识别率的情况下，保证系统较高的运行效率(时间上和空间上)。

文献[15],[19],[20]等中都是使用了数据流的前面几个数据包进行加密流量的识别。此外，熵是一种统计意义上对特定对象混乱度的度量，从统计学的角度来说，可以通过抽样调查的方法从整体中抽取部分样本来对整体进行估计。用样本来推断总体，从局部来认识全局，是重要的统计思想。

本文借鉴了文献[15]中的实验结果，针对流有效负载前 1K 的内容进行相对熵的计算，对于总的缓存大小不足 1K 字节的流，考虑到加密通信信道的性质（在实际观测中，几乎所有的加密流的有效负载的大小都是大于 1000 字节的），可认为其不是加密流，予以排除。

### 2.6 SVM 算法

支持向量机（SVM，support vector machine）是一种分类算法，通过寻求结构化风险最小来提高学习机泛化能力，实现经验风险和置信范围的最小化，从而达到在统计样本量较少的情况下，亦能获得良好统计规律的目的。通俗来讲，它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

## 3 实验结果与分析

### 3.1 系统流程图

系统流程如图 3 所示。主要用 C 语言进行编写，运行环境为 Linux（内核 2.6 及以上版本），需要的第三方软件及 API 包括：libpcap、pthread。

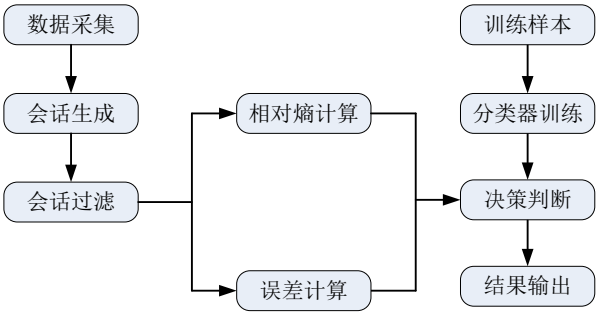



图 3 加密流量识别系统流程图

### 3.2 数据集

本文中实验数据集的获取过程为：使用 4 台普通主机向 FTP 服务器以加密方式传送数据来获得加密流量（拓扑结构如图 4 所示），通过嗅探捕获在数据传输过程中产生的数据包。另外还捕获主机在正常通信时的数据样本流量。最后将这这些数据样本使用 Wireshark 的 mergecap 命令将其整合成一个 pcap 文件，形成最终数据集。

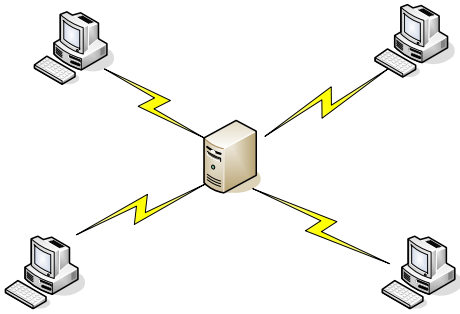


图 4 数据采集网络拓扑结构

在传统的监督学习中，分类器通过对大量有标记的（labeled）训练实例进行学习，从而建立模型用于预测未标记实例的类别。收集大量未标记（unlabeled）实例是相当容易的，而获取大量有标记实例则相对较为困难。在这里，为了简化训练分类器的过程，训练样本的特征是对已标记的加密流量和非加密流量进行特征提取得到的，从而可以用来方便地对 SVM 分类器进行快速训练。

本文选取一个 Web 访问会话流和 TLS 加密会话流，由网络分析工具 Wireshark 查看会话数据流内容，如图 5、图 6 所示。

图 5 Web 访问会话流

图 6 TLS 加密会话流

### 3.3 评估策略

为了评价识别算法的性能，本文选用了 3 种常用的评价指标，包括查准率（Precision）、查全率（Recall）和综合评价（F-measure）。

下面对这些常用的评价指标做简单描述。真正 TP 代表实际类型为加密的样本中被正确标记的样本数；假负 FN 代表实际类型为加密的样本中被误标识为非加密的样本数；假正 FP 代表实际类型为非加密的样本中被误标识为加密的样本数。根据这些概念，给出衡量识别系统查准率（Precision）、查全率（Recall）和综合评价（F-measure）的形式化描述。

$$\text{查准率 } precision = \frac{TP}{TP+FP} \quad (2)$$

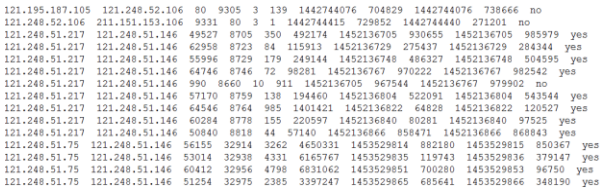
$$\text{查全率 } recall = \frac{TP}{TP+FN} \quad (3)$$

$$F-Measure = \frac{2\times presion\times recall}{presion+recall} \quad (4)$$

查准率和查全率体现了识别方法的识别效果，F-Measure 是根据查准率 Precision 和查全率 Recall 二者给出的一个综合的评价指标，当 F-Measure 较高时则比较说明方法比较理想。

### 3.4 算法结果分析

本文选取了 73 个加密通信流和 276 个正常数据通信流，数据包大小为 224M，然后使用加密流量识别系统对这些会话数据流进行检测。实验结果文件如图 7 所示，经过统计后的实验结果如表 2 所示。其中实验结果文件的每行内容分别代表源 IP、宿 IP、源端口、宿端口、包数、字节数、流开始时间（秒、微秒）、流结束时间（秒、微秒）、识别结果。



```
121.195.187.105  121.248.52.106   80   9305   3   139   1442744076  704829   1442744076  738666   no
121.248.52.106   211.151.153.106  9331 80     3   1     1442744415  729852   1442744440  271201   no
121.248.51.217   121.248.51.146   49527  8705   350   492174   1452136705  930655   1452136705  985979   yes
121.248.51.217   121.248.51.146   62958  8723   84    115913   1452136729  275437   1452136729  284344   yes
121.248.51.217   121.248.51.146   55996  8729   179   249144   1452136748  486327   1452136748  504595   yes
121.248.51.217   121.248.51.146   64746  8746   72    98281    1452136767  970222   1452136767  982542   yes
121.248.51.217   121.248.51.146   990    8660   10    911      1452136705  967544   1452136767  979902   no
121.248.51.217   121.248.51.146   57170  8759   138   194460   1452136804  522091   1452136804  543544   yes
121.248.51.217   121.248.51.146   64546  8764   985   1401421  1452136822  64828    1452136822  120527   yes
121.248.51.217   121.248.51.146   60284  8778   155   220597   1452136840  80281    1452136840  97525    yes
121.248.51.217   121.248.51.146   50840  8818   44    57140    1452136866  858471   1452136866  860843   yes
121.248.51.75    121.248.51.146   56155  32914  3262  4650331  1453529814  882180   1453529815  8503677  yes
121.248.51.75    121.248.51.146   53014  32938  4331  6165767  1453529835  119743   1453529836  379147   yes
121.248.51.75    121.248.51.146   60412  32956  4798  6831062  1453529851  700280   1453529853  96750    yes
121.248.51.75    121.248.51.146   51254  32975  2385  3397247  1453529865  685641   1453529866  348190   yes
```

图 7 实验结果文件

表 3 实验结果统计

| | 69 个加密通信流 | | 276 个正常通信流 | |
|---|---|---|---|---|
| 检测结果 | 正确 | 错误 | 正确 | 错误 |
| 数量 | 61 | 8 | 272 | 4 |

由式（3）、（4）、（5）可得到本次实验的查准率、查全率和综合评价分别为 93.85%、88.41%、91.04%。从结果可以看出，本识别系统的效果较好，可以对加密流量进行检测。此外，利用同样的数据集本文也对仅使用相对熵的算法的实验结果结果进行了统计，结果如表 3 所示。

表 4 仅使用相对熵的结果统计

| | 69 个加密通信流 | | 276 个正常通信流 | |
|---|---|---|---|---|
| 检测结果 | 正确 | 错误 | 正确 | 错误 |
| 数量 | 60 | 9 | 234 | 42 |

同理可得，仅使用相对熵的算法的查准率、查全率和综合评价分别为 58.82%、86.96%、70.18%。将这两种方法的结果进行比较，如图 8 所示。

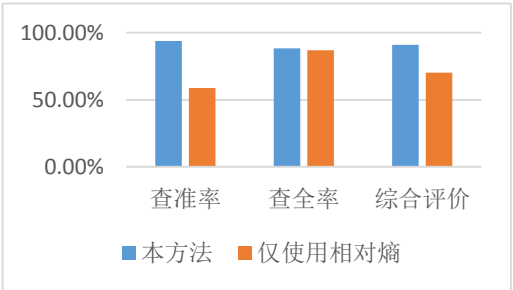



图 8 两种方法的实验结果对比图

由图可知，本方法的效果要优于仅使用相对熵的识别算法。对该实验结果进行分析：单纯地基于相对熵的方法会将压缩流量识别为加密流量，所以在正常流量中会存在一定的误报，对比表 3 和表 4 可知，38 个正常的通信流被识别为加密流量从而提高误报率，而对加密通信流的识别这两种方法的结果偏差不是很大。另外，我们对所有加密流中识别出来的和未识别出来的流进行包数和字节数的统计，得到散列图如图 9 所示。

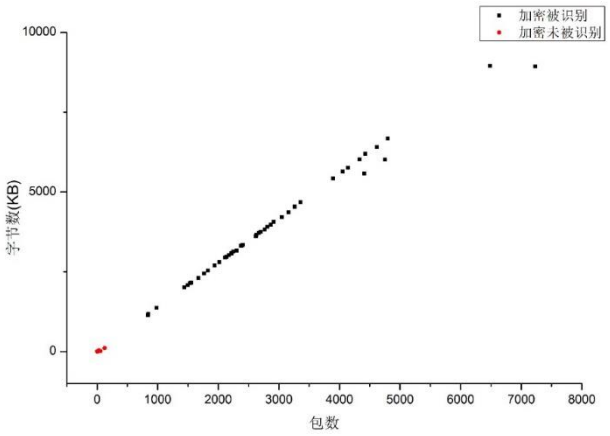



图 9 加密流包数和字节数的分布图

由上图可以看出，未被识别的加密流大多包数和字节数都比较小，而被识别出来的包数和字节数都较大，说明本识别系统对通信数据量较小的加密流识别效果不佳。这一点从本文所使用的方法上可以进行解释。混乱性和随机性是一种从统计学角度进行度量的性质，如果样本数据量少，可能并不能从这两个特征尺度对其进行考量，从而必然会产生一种情况，那就是数据量较小时本方法会失效。

## 4 结束语

本文提出了一种基于流的加密流量识别方法。该方法使用了相对熵和 Monte Carlo PI 估计误差来对加密流量进行识别。基于相对熵的方法主要是从流量加密后的内容混乱性角度进行考虑，而 Monte Carlo PI 估计误差估计是从加密后内容的随机性角度考量。在得到特征向量后，使用从训练样本得到的 SVM 分类器对待检测数据流进行决策判断，最终给出待检测流量中加密数据流的相关信息。实验结果表明，该方法识别准确率较高且误判率较低，优于仅使用相对熵作为特征向量的方法并能实现实时识别和处理。下一步的工作将主要集中在研究如何对通信数据量较小的数据流进行有效地识别并且使用不同的机器学习算法对分类器模型进行训练从而能够得到较优的分类器。此外，本文的方法仍然需要使用有效负载作为系统的输入，但是解析数据包负载会触犯隐私，这是本方法必须要面对的问题。

## 参考文献

[1] Fadlullah Z M, Taleb T, Vasilakos A V, et al. DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis[J]. IEEE/ACM Transactions on Networking (TON), 2010, 18(4): 1234-1247.

[2] Gu G, Perdisci R, Zhang J, et al. BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection[C]//USENIX Security Symposium. 2008: 139-154.

[3] Tankard C. Advanced Persistent threats and how to monitor and deter them[J]. Network security, 2011, 2011(8): 16-19.

[4] Dorfinger P, Panholzer G, John W. Entropy estimation for real-time encrypted traffic identification (short paper)[M]. Springer Berlin Heidelberg, 2011.

[5] 赵博, 郭虹, 刘勤让, 等. 基于加权累积和检验的加密流量盲识别算法[J]. 软件学报, 2013, 24(6): 1334-1345.

[6] Bonfiglio D, Mellia M, Meo M, et al. Revealing skype traffic: when randomness plays with you[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 37-48.

[7] Okada Y, Ata S, Nakamura N, et al. Comparisons of machine learning algorithms for application identification of encrypted traffic[C]//Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on. IEEE, 2011, 2: 358-361.

[8] Alshammari R, Zincir-Heywood A N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection[J]. Computer networks, 2011, 55(6): 1326-1350.

[9] Erman J, Mahanti A, Arlitt M, et al. Semi-supervised network traffic classification[C]//ACM SIGMETRICS Performance Evaluation Review. ACM, 2007, 35(1): 369-370.

[10] Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification [J]. ACM SIGCOMM Computer

Communication Review, 2006, 36(5): 5-16.

[11] Xie G, Iliofotou M, Keralapura R, et al. SubFlow: towards practical flow-level traffic classification[C]//INFOCOM, 2012 Proceedings IEEE. IEEE, 2012: 2541-2545.

[12] He G, Yang M, Luo J, et al. A novel application classification attack against Tor[J]. Concurrency and Computation: Practice and Experience, 2015.

[13] 李文忠，左万利，赫枫龄. 一种基于信息熵的多维流数据噪声检测算法[J].计算机科学，2012,39（2）：123-144.

[14] 丁世飞，朱红，许新征，等.基于熵的模糊信息测度研究[J].计算机学报，2012,30(8)：139-151

[15] Y. Wang, Z. Zhang, L. Guo, and S. Li, “Using entropy to classify traffic more deeply,” inProc. IEEE Int. Conf. Netw., Archit., Storage,2011,pp. 45–52

[16] 吴乐南等. 数据压缩[M]. 电子工业出版社, 2005

[17] 徐燕，李锦涛，王斌等. 基于区分类别能力的高性能特征选择方法[J].软件学报，2008，19(1)：82-89

[18] 徐峻岭，周毓明，陈林，等. 基于互信息的无监督特征选择[J]. 计算机研究与发展，2012，49(2) :372-382.

[19] Bernaille L, Teixeira R. Early recognition of encrypted applications [M]//Passive and Active Network Measurement. Springer Berlin Heidelberg, 2007: 165-175.

[20] Maiolini G, Baiocchi A, Iacovazzi A, et al. Real time identification of SSH encrypted application flows by using cluster analysis techniques[M]//NETWORKING 2009. Springer Berlin Heidelberg, 2009: 182-194.