基于闭合序列模式挖掘的未知协议格式推断方法

张洪泽[1] 洪 征[1] 王 辰[2] 冯文博[1] 吴礼发[1]
(中国解放军陆军工程大学指挥控制工程学院 南京 210000)[1]
(中国人民解放军32179部队 北京 100000)[2]

摘 要 现有的基于网络流量的协议格式推断方法只提取报文关键字的平坦序列,并没有考虑报文关键字之间的顺序、并列与层次关系的结构特性;此外,报文样本中的噪音往往导致关键字识别的准确率偏低。文中提出了一种自动识别未知协议报文关键字并推断报文结构的方法。所提出的方法在收集未知协议实体程序通信报文的基础上,采用二阶段闭合模式挖掘策略对通信报文实施闭合序列模式挖掘,识别协议关键字并生成包含具有关键字组合关系的关键字序列;在此基础上提取关键字之间的顺序、并列以及层次关系,进而推断报文结构。协议关键字识别过程中采用设置最小支持度阈值的方法,可直接分析实际网络中包含噪音的报文样本,保证了关键字识别的准确率。实验结果表明,所提出的协议格式推断方法被应用于文本协议和二进制协议时,对报文关键字识别与报文结构推断均能取得理想的推断效果。

关键词 协议逆向工程,网络流量,协议格式推断,闭合序列模式挖掘,报文结构推断
中图法分类号 TP398.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.06.011

Closed Sequential Patterns Mining Based Unknown Protocol Format Inference Method

ZHANG Hong-ze[1] HONG Zheng[1] WANG Chen[2] FENG Wen-bo[1] WU Li-fa[1]
(Institute of Command and Control Engineering, Army Engineering University of PLA, Nanjing 210000, China)[1]
(Unit 32179 of PLA, Beijing 100000, China)[2]

Abstract Current protocol format inferring methods based on network traffic can only extract flat sequence of keywords, and they do not consider the structural features of message keywords, such as sequential, hierarchical and parallel relation between the keywords. Additionally, the noise in message samples always lead to low recognition accuracy of keywords. This paper presented a method to automatically identify keywords of unknown protocol message and infer the message structure. Based on the collected communication messages of the unknown protocol, the method implements two-phase closed sequential patterns to identify protocol keywords and generate keywords sequence with keyword composition relation, extract sequential, hierarchical and parallel relation of the keywords, and then infer messages structure inference. To ensure recognition accuracy of the keywords, the method analyzes message samples directly containing noise by setting minimum support in keywords identification procedure. Experimental results show that the proposed method performs well in keywords identification and message structure inference for both text protocol and binary protocol.

Keywords Protocol reverse engineering, Network traffic, Protocol format inference, Closed sequential patterns mining, Message structure inference

1 引言

网络协议作为网络通信的核心要素,它的质量直接关系到通信的稳定性、可靠性和安全性。对网络协议进行分析,发掘网络协议及其具体实现程序中存在的安全漏洞,并及时实施安全防护,有助于减少安全问题的发生。然而,现有的协议分析软件仅支持对已知协议的分析,例如著名的开源软件Wireshark可以解析2000余种已知协议,对于协议规范没有公开的各类协议则无能为力。在这种情况下,研究人员尝试利用协议逆向分析技术来获取未知协议规范。

到稿日期:2018-05-22 返修日期:2018-08-25 本文受国家重点研发计划项目(2017YFB0802900)资助。
张洪泽(1993-),男,硕士生,主要研究方向为信息安全;**洪 征**(1979-),男,博士,副教授,主要研究方向为信息安全,E-mail:hz5215@163.com(通信作者);**王 辰**(1990-),男,硕士,研究实习员,主要研究方向为信息安全;**冯文博**(1994-),男,硕士生,主要研究方向为信息安全;**吴礼发**(1968-),男,博士,教授,主要研究方向为信息安全。

协议逆向分析技术以协议格式和协议状态机的获取为目标。协议格式的获取主要是推断协议关键字、报文结构以及字段语义等信息。协议状态机的获取是在协议格式信息的基础上识别整个协议运行过程中存在的协议状态，并分析协议状态之间的转换关系[1]。依据研究对象的不同，协议逆向分析可分为基于执行轨迹(Execution Trace Based)的逆向分析技术与基于网络流量(Network Traffic Based)的逆向分析技术两类[2]。基于执行轨迹的逆向分析技术通过监视协议实体对报文的处理过程以及各报文片段的使用方式获得报文格式信息。基于网络流量的逆向分析技术是基于这样一种考虑：每个协议报文都是协议规范的具体实例，相同类型的协议报文具有相似性，这种相似性能够反映报文格式中相对稳定的部分，基于这种相似性可以推断协议报文的格式。与基于执行轨迹的逆向技术相比，基于网络流量的逆向技术的收集分析样本过程更容易、自动化程度更高，并且应用范围更为广泛[3]，本文将从基于网络流量分析的角度来研究协议格式的推断方法。

未知协议格式的推断通常建立在协议关键字识别的基础之上。基于网络流量的协议关键字识别的目标可分为以下两种。1)只以获取报文中独立的关键字为目标。例如，Luo等[4]提出的 AutoReEngine 方法具有较高的关键字识别准确率[3]，该方法通过采用 Apriori 算法挖掘频繁字符串，其中位置变化频率小于阈值的频繁字符串被认为是协议关键字；Zhang 等[5]提出的 ProWord 原型系统通过引入自然语言处理的断词和短语识别技术来实现协议关键字的识别；此外，一些学者引入隐半马尔科夫模型[6]或采用最大似然估计[7]等技术实现协议关键字的识别。但上述方法只能获取报文中单独的关键字，忽略了关键字之间的组合约束关系[8]，难以进行报文结构推断和字段语义提取研究。2)对整个报文进行划分并提取报文特征，以识别关键字并提取关键字之间的组合关系为目标。例如，PI 项目[9]通过引入生物信息学的序列比对算法，尝试对目标协议进行分析；协议逆向工具 Netzob[10]也采用序列比对方法，但对于位置变化大或者变长字段过多的报文，该方法的推断结果的准确率较低。Cui 等[8]提出 Discoverer 方法，该方法将报文分为 Token 片段，通过 Token 片段聚类，找出相似结构的协议数据，并获得部分语义信息。Kureger 等[11]提出 PRISMA 方法，该方法通过 n-gram 方法识别协议关键字，通过 Pearson 相关系数计算获得关键字之间的相关性，并推断协议的语义模板。这些方法尽管能够在一定程度上获取协议关键字之间的组合关系，但也只是获取了平坦的关键字序列，并没有充分挖掘报文的结构信息，也就是报文关键字之间的顺序、并列以及层次关系。

文献[12]指出报文结构属性在协议逆向及应用中至关重要。报文结构信息是实现关键字语义提取的前提，通过报文的结构信息可以构建关键字所在报文的完整格式语法树，利用报文结构信息构建报文语义模板(Message Semantic Template)[11]也是实现状态机逆向推断的基础。在协议模糊测试时，利用报文结构信息有助于了解程序解析的轨迹，可以减少冗余测试用例的生成。但是已有报文结构推断研究多数是利用基于执行轨迹的逆向技术[12-13]，鲜有利用基于网络流量的逆向技术来推断报文结构的方案。

此外，目前基于网络流量的逆向分析方法在收集未知协议报文样本时存在以下问题。1)对于未知网络协议，我们无法准确得知协议特征，无论采取何种协议识别方法，都可能导致目标报文样本会混入其他协议报文[14]；同时，实际网络环境中通信链路的质量问题会导致样本内经常出现一些时序混乱、完整性缺失的报文，本文将这些报文统称为噪音。协议噪音对最终的分析结果会造成干扰，文献[2]提出在协议逆向预处理过程中需要剔除原始样本中的干扰；但是文献[14]指出在协议格式未知的情况下，无法确定实际网络环境中截获协议的样本中是否包含噪音，要想消除协议噪音更是困难，这种矛盾影响着协议逆向结果的准确性。另一方面，为保证逆向分析的准确率，原则上应以完备样本集作为分析对象，但是获取完备样本集的难度较大，因此实际分析过程中往往收集尽可能多的协议样本来构建报文样本集[3]。然而，过多的报文样本可能存在耗时超过可容忍的界限或因内存溢出而导致计算进程不得不中断等问题。

针对以上问题，本文提出报文结构提取方法，通过设计二阶段闭合模式挖掘策略，对网络流量进行闭合序列模式挖掘，进而识别关键字，生成包含关键字组合关系的关键字序列，再依据关键字序列的信息推断报文结构，获得包含关键字顺序关系、平行关系、层次关系的协议格式巴克斯范式(Backus Normal Form，BNF)，提高了协议逆向结果的应用价值。同时，所提方法采用二阶段策略，可以使得闭合模式挖掘满足分析较大规模报文样本的需求，关键字的出现频率只需大于设定阈值即可被识别，这样有助于降低样本内协议噪音的影响，保证关键字识别的准确率。

2　协议格式推断问题的分析

从通信协议的角度看，协议实体程序需要将传输的数据序列化为字节流在网络中进行传输。字节流中往往使用一些固定模式的字符串来表达特定含义，有的用来标识报文类型，例如版本号(如 HTTP 协议的“HTTP/1.0”)、协议名称等，有的用来传递相关的控制信息，例如命令码(如 HTTP 协议的“GET”)等，这些具有特定含义的固定模式的字符串被称为协议关键字。协议规范中通常会定义一些关键字，这些关键字以固定模式字符串在通信数据中频繁出现，可以基于频繁模式挖掘思想提取这些固定模式字符串，识别协议关键字。无论是以字节为单位的文本协议，还是以比特为单位的二进制协议，都是如此。为便于描述，后文针对文本协议进行讨论。

现有协议逆向方法通常将协议关键字作为报文字段划分的基础，把前后两个关键字之间的载荷内容作为一个变量字段，那么报文格式可以描述成“关键字字段＋变量字段”的形式。如图 1 所示，HTTP 报文样例被划分为 $K_1 \parallel D_1 \parallel K_2 \parallel \cdots \parallel K_n \parallel D_n$，“$\parallel$”代表字符串的拼接，$K$ 是关键字，D 是变量字

段，变量字段的长度可以为 0。例如，K_1 为关键字 GET，D_1 为变量字段/cgi-bin/whois.pl。

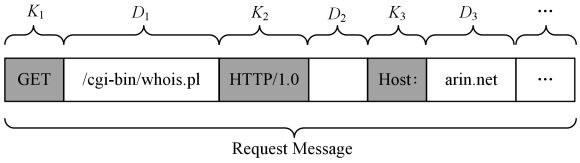


图 1 HTTP 协议请求报文的示例

Fig. 1 Example of HTTP protocol request message

根据现有研究经验[11,15]，协议格式的协议字段、报文结构以及字段语义中缺失任何要素都难以构建合法的报文，会限制协议逆向结果的应用。然而，现有的报文分段方法只分析平坦的协议关键字序列，并没有充分考虑协议关键字之间的结构关系，导致逆向结果中报文的结构信息不完整。

协议实体程序主要依据协议关键字对报文进行解析，分析报文的含义。协议实体程序对报文的解析主要包括如下 3 种方式。

1)从左至右的解析。采用这种解析方式意味着相应的协议关键字之间具有顺序关系(Sequential)。如图 2 所示，在一条合法的报文上，关键字“GET”必须出现在关键字“HTTP/1.0”之前，关键字“GET”与关键字“HTTP/1.0”之间属于顺序关系。

2)报文存在层次化解析。底层的关键字与关键字之间的变量字段通常需要组合后作为复合字段使用，所构成的复合字段具有独立的语义，底层的关键字与上层的关键字之间属于层次关系(Hierarchical)。如图 2 所示，HTTP 请求报文中的字段“Request-Line”具有独立的语义，它包含关键字“GET”与“HTTP/1.0”，用于表示 HTTP 协议请求。字段“Request-Line”与关键字“GET”之间构成层次关系。

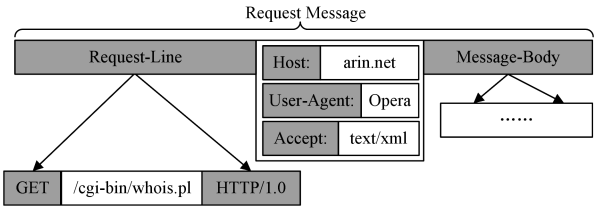


图 2 HTTP 协议请求报文结构的示例

Fig. 2 Example of HTTP protocol request message structure

此外，协议实体程序可能先搜寻特定关键字再执行相应的关键字语义，这种关键字在报文中互换位置时对报文信息的处理没有影响，它们之间属于并列关系(Parallel)。例如，在 Web 服务器处理 HTTP 请求报文时，报文解析代码使用 while 语句构造循环，使用多个 if 语句查找指定关键字并解释语义，直至查找到所有关键字才退出循环[11]。那么，实际的通信报文中，关键字“Host:”“User-Agent:”与“Accept:”在报文上出现的先后顺序可以是不固定的，如图 2 所示，它们之间属于并列关系。

通常，报文结构被定义为：

$$\begin{cases} G::=K_1K_2(K_3|K_4) \\ K_4::=\{K_{4-1}K_{4-2}\} \end{cases}$$

其中，G 是协议格式的巴克斯范式[16]，符号 K_i 表示协议中第 i 个关键字；K_1K_2 代表关键字的顺序关系，$K_3|K_4$ 代表关键字的并列关系，$K_4::=\{K_{4-1}K_{4-2}\}$ 表示字段 K_4 可拆分为关键字 K_{4-1} 与 K_{4-2}，K_4 与 K_{4-1} 和 K_{4-2} 满足关键字的层次关系。图 2 所示 HTTP 协议请求报文对应的协议巴克斯范式如下：

$$\begin{cases} G::=Request\text{-}Line(Host:|User\text{-}Agent:|Accept:) \\ \qquad Message\text{-}Body \\ Request\text{-}Line::=\{GET\ HTTP/1.0\} \\ Message\text{-}Body::=\cdots \end{cases}$$

从报文结构的角度看，协议关键字之间的结构关系会在大量的网络通信报文中表现出来，因此可以通过频繁模式挖掘提取这些相对稳定的结构关系，进而推断报文的结构属性。

序列模式挖掘(Sequence Pattern Mining)是频繁模式挖掘的扩展，它适用于分析有序数据。闭合序列模式挖掘是一种基于约束的序列模式挖掘[17]，相比于序列模式挖掘，它的挖掘过程更高效且结果更精简。由经验可知，网络协议报文是字符的有序序列，并且关键字是频繁字符串的闭合模式，因此本文将闭合序列模式挖掘的方法应用于协议格式的推断过程中。

3 协议格式推断的流程

本文所提出的协议格式推断方法主要包括报文预处理、协议关键字识别以及报文结构推断 3 个阶段。

基于网络流量的协议逆向分析技术依据通信报文之间的相似性进行分析，因此首先需要对捕获的网络通信报文进行预处理，即将格式相似的报文聚集在一起。本文的报文预处理方法参考文献[18]：将报文样本按照五元组划分为若干个会话(Session)，每个报文以各自在会话中的先后顺序标记序号，具有相同序号的报文作为一类。由于网络传输中可能出现报文丢包、乱序以及交互顺序不一致等因素，此时每个类内报文序号相同并不能保证报文是相同类型的，因此还需进一步对每个类内报文提取载荷字符序列；然后通过计算类内各个报文之间的最频繁、最长公共串找到相同类型的报文，将同类型的报文组成一个报文组(Messages Group)。每个报文组内，字符序列集合作为协议格式推断算法的输入。

完成报文预处理操作之后，将依次进行关键字识别和报文结构推断。协议关键字识别的流程为：首先基于闭合序列模式挖掘提取固定模式字符串，再采用关键字识别策略识别属于关键字的固定模式字符串。报文结构推断以提取的关键字序列为基础，进一步区分不同关键字之间的顺序、并列以及层次关系，进而推断报文结构。下面将依次介绍协议关键字识别与报文结构推断的方法。

4 关键字的识别

关键字识别以报文字符序列集合作为输入，基于闭合序列模式挖掘方法识别协议关键字。为了方便对关键字识别过程进行描述，首先给出一些概念的定义。

4.1 相关概念

报文由顺序排列的字符组成，一条报文可以表示成字符的有序集合，即 $L_i=\langle e_1,e_2,\cdots,e_n\rangle$，其中 e_j 表示一个字符。报文组由多条报文组成，一个报文组可以表示成 $C_L=\{L_1,L_2,\cdots,L_m\}$。如：表 1 中的报文组 $C_L=\{L_1,L_2,\cdots,L_5\}$ 由 5 条报文所组成，它们属于相同类型的报文。

表 1　报文组示例

Table 1　Example of messages group

ID	Messages
L_1	GET /cgi-bin/whois.pl HTTP/1.0 Host:arin.net User-Agent:Opera Accept:text/xml
L_2	GET /index.html HTTP/1.0 Host:www.yahoo.com User-Agent:Mozilla/5.0 Accept:text/xml
L_3	GET /HTTP/1.0 Host:www.google.com User-Agent:IE4.0 Accept:text/xml
L_4	GET /images/go.gif HTTP/1.0 Host:www.foobar.com Accept:*/* User-Agent:Opera/9.20
L_5	GET static/llbY0YcG.html HTTP/1.0 User-Agent:Mozilla/4.0 Accept:text/css Host:129.174.88.71

定义 1（邻接子序列，Contiguous Subsequence）　对于两个序列 $t_1=\langle a_1,a_2,\cdots,a_i\rangle\subseteq L_i$ 和 $t_2=\langle b_1,b_2,\cdots,b_j\rangle\subseteq L_i$，$t_1$ 是 t_2 的子序列，当且仅当存在整数 $k_1,k_2,\cdots,k_i$，满足 $1\leqslant k_1<k_2<\cdots<k_i\leqslant j$，$a_1=b_{k_1}$，$a_2=b_{k_2}$，$\cdots$，$a_i=b_{k_i}$，此时的 t_2 被称为 t_1 的一个超序列。如果字符 a_n 与 $a_{n+1}(1\leqslant n<i)$ 在报文 L_i 中位置相邻，那么 t_1 是 t_2 的邻接子序列。

沿用序列模式挖掘的支持度（Support）与置信度（Confidence）的概念：给定一个报文组 C_L 与 C_L 内的某条报文 L_i，将 C_L 中的报文总数记作 $|C_L|$，L_i 的某个子序列记作 t，C_L 中包含子序列 t 的报文的个数记作 $|C_{L_t}|$，则 $|C_{L_t}|$ 与 $|C_L|$ 的比值为 t 在报文组 C_L 上的支持度，记作 $Sup_{C_L}(t)=\dfrac{|C_{L_t}|}{|C_L|}$；如果 L_i 的两个子序列 t_i 与 t_j 不重合，将 t_i 与 t_j 组合成一个新序列，记作 $t_i\cup t_j$，那么 $t_i\cup t_j$ 在报文组 C_L 上的支持度与 t_i 在报文组 C_L 上的支持度的比值为 t_i 至 t_j 的置信度，记作 $Conf_{C_L}(t_i\rightarrow t_j)=\dfrac{Sup_{C_L}(t_i\cup t_j)}{Sup_{C_L}(t_i)}$，置信度越高表示当 t_i 出现时，t_j 出现的可能性就越大。在闭合序列模式挖掘时，需设定最小支持度阈值与最小置信度阈值，分别用 Min_Sup 与 Min_Conf 表示。

定义 2（闭合频繁邻接段，Closed Frequent Contiguous Segment）　给定最小支持度阈值 $Min\_Sup\in(0,1)$，对于报文组 C_L 内某条报文 L_i 的某个邻接子序列 t，当 $Sup_{C_L}(t)\geqslant Min\_Sup$ 时，称 t 为 C_L 上的频繁邻接段。如果 t 的超序列的支持度都不大于 t 的支持度，那么 t 是闭合的，称 t 为 C_L 的闭合频繁邻接段。为了阐述方便，用 s 表示闭合频繁邻接段，将报文组 C_L 中所有的闭合频繁邻接段 $s_1,s_2,\cdots,s_n$ 汇集在一起组成集合 $S_{C_L}=\{s_1,s_2,\cdots,s_n\}$。以表 1 为例，如果设定 $Min\_Sup=0.9$，$\langle GET\rangle$ 在表 1 中的 5 条报文中均出现，其支持度是 1，因此 $\langle GET\rangle$ 是频繁邻接段；$\langle GET\rangle$ 再添加任意邻接的字符后，支持度均小于 1，因此 $\langle GET\rangle$ 是闭合频繁邻接段。当 $Min\_Sup=0.9$ 时，表 1 中包括的闭合频繁邻接段的示例如表 2 所列。

表 2　闭合频繁邻接段示例

Table 2　Example of closed frequent contiguous segment

ID	Closed Frequent Contiguous Segment
s_1	$\langle$GET:$\rangle$
s_2	$\langle$Host:$\rangle$
s_3	$\langle$HTTP/1.0$\rangle$
s_4	$\langle$User-Agent:$\rangle$
s_5	$\langle$Accept:$\rangle$

定义 3（闭合频繁序列，Closed Frequent Sequence）　报文组 C_L 上的频繁邻接段 s 构成的有序集合 $\theta_s=\langle s_i,\cdots,s_j,\cdots,s_k\rangle$，$\theta_s$ 满足 $Sup_{C_L}(\theta_s)\geqslant Min\_Sup$，$\theta_s$ 中各个 s 的先后顺序与在报文中出现的先后顺序相同，并且 θ_s 的超序列的支持度都不大于 θ_s 的支持度时，称 θ_s 为闭合频繁序列。

以表 1 为例，如果设定 $Min\_Sup=0.9$，$\langle\langle GET\rangle,\langle HTTP/1.0\rangle,\langle Host:\rangle\rangle$ 在表 1 中 5 条报文中均出现，其支持度是 1，并且其再添加任意 s 后 θ_{s1} 的支持度均小于 1，因此 $\langle\langle GET\rangle,\langle HTTP/1.0\rangle,\langle Host:\rangle\rangle$ 是闭合频繁序列。当 $Min\_Sup=0.9$ 时，表 1 对应的闭合频繁序列如表 3 所列。

表 3　闭合频繁序列示例

Table 3　Example of closed frequent sequence

Number	Closed Frequent Sequence
θ_{s1}	$\langle\langle$GET$\rangle,\langle$HTTP/1.0$\rangle,\langle$Host:$\rangle\rangle$
θ_{s2}	$\langle\langle$GET$\rangle,\langle$HTTP/1.0$\rangle,\langle$User-Agent:$\rangle\rangle$
θ_{s3}	$\langle\langle$GET$\rangle,\langle$HTTP/1.0$\rangle,\langle$Accept:$\rangle\rangle$

定义 4（闭合序列模式，Closed Sequential Pattern）　报文组 C_L 中所有 θ_s 的集合为闭合序列模式，记作 $F_{C_L}=\{\theta_{s1},\theta_{s2},\cdots,\theta_{sn}\}$。

由上文的基本定义可知，报文序列获取的闭合频繁邻接段 s 是出现频率高并且闭合的字符串，对于每个闭合频繁邻接段，可直接采用启发式策略来确定其是否为协议关键字。

4.2 关键字识别策略

为了有效地识别协议关键字，本文以报文组中的闭合频繁邻接段为基础，提出两个关键字识别策略。

1）在报文中，距离报文起始位置或者结束位置固定偏移长度的频繁字符串往往是协议关键字。例如，迅雷协议应用层载荷的前 4 个字节 0×39，0×00，0×00，0×00 表示协议的控制信息，属于迅雷协议的关键字，而其他位置出现这 4 个字节仅为偶然或其他含义。本文规定，若存在 $s_i(s_i\in S_{C_L})$ 在报文中相对报文起始位置或者结束位置有固定的偏移长度，则可以将 s_i 视为一个协议关键字 K。

2）协议中的关键字通常由多个字符组合而成，在报文样本中关键字通常以字符串的形式频繁出现，并且关键字对应的字符串子集或者超集都不再是关键字。例如，HTTP/1.0 协议报文载荷连续出现的字符 0×48，0×54，0×54，0×50，0×2f，0×31，0×2e，0×30 表示协议的版本信息，属于 HTTP/1.0 协议的关键字，这些连续字符的位置不固定，并且不论删除或者添加某个字符后其均不再是关键字。识别此类关键字时首先需要删除集合 S_{C_L} 中被其他元素包含的元素，以

排除字符之间偶然组合成为关键字的一部分而非完整关键字的字符串的情况。对集合 S_{C_L} 执行删除操作后得到集合 S'_{C_L}，S'_{C_L} 内剩余的任意闭合频繁邻接段 s_i 不论添加还是删掉字符，均不再属于 S'_{C_L}，因此可以将 s_i 视为关键字 K，S'_{C_L} 视为这类关键字的集合。如果有的关键字之间 *Data* 变量的字段长度一直是 0，那么利用上述分析方法识别关键字时可能得到多个关键字的连续组合。通常，关键字长度不超过 10 个字符[14]，因此本文将长度超过 10 的字符串看作由多个关键字按序邻接组合而成，这类关键字之间邻接并具有顺序关系，并不影响关键字的提取与报文结构的推断。

上述关键字识别策略是基于闭合频繁邻接段而提出的，因此挖掘报文的闭合频繁邻接段是协议关键字识别的首要步骤，接下来将详细介绍挖掘报文闭合频繁邻接段的方法。

4.3 两阶段闭合序列模式挖掘方法

从序列模式挖掘自身的特点可知，采用传统的闭合序列模式挖掘方法直接得到的是闭合序列模式 F_{C_L}。但由于报文数据具有序列长、数据稠密等特点，采用传统闭合序列模式挖掘方法，以报文内字符作为分析单元，将导致内存消耗巨大、计算时间过长等问题。本文针对性地提出两阶段闭合序列模式挖掘方法，其能够提高效率，降低内存消耗。

两阶段报文闭合序列模式挖掘方法的第一阶段为分割阶段(Segment Phase)：针对报文集合，搜索所有的闭合频繁邻接段，并采用关键字识别策略判断这些闭合频繁邻接段是否为关键字。第二阶段为模式挖掘阶段(Pattern Mining Phase)：使用第一阶段被推断为关键字的闭合频繁邻接段生成由多个关键字组成的闭合频繁序列 $\theta_s=\langle s_1,s_2,\cdots,s_k\rangle$，最终得到报文组内所有 θ_s 的集合 F_{C_L}。

4.3.1 分割阶段

分割阶段以报文组 C_L 作为输入，以字符为基本分析单元，以闭合频繁邻接段集合 S_{C_L} 的获取为目标。本文基于改进 CCSpan 算法[19]挖掘闭合频繁邻接段 s，进而获得集合 S_{C_L}，主要实施步骤如下。

Step1 候选字符片段的生成。候选字符片段由报文组 C_L 提取，利用 n-gram 模型[20]将报文分成字符片段，字符片段中的字符保持原有的顺序和邻接属性。对于一个报文组，以固定长度对报文进行切分，在下一次切分时长度增加 1，依次类推。例如，对于报文部分片段 GET/index.html，在 N_1-切分阶段，将切分报文序列生成的所有 1-字符片段组成集合 $N_1=\{G,E,T,/,\cdots,.,h,t,m,l\}$，$N_2$-切分阶段生成的所有 2-字符片段组成集合 $N_2=\{GE,ET,T,\cdots,tm,ml\}$，直至 $N_{15}=\{\mathrm{GET}/index.html\}$。

Step2 候选字符片段的筛选。报文组中每个报文都被离散化为很多字符片段，其中许多字符片段不符合闭合频繁邻接段的要求，因此首先依据频繁的特征对候选字符片段进行筛选。频繁特征筛选是指如果一个候选字符片段的支持度不小于支持度阈值，那么该候选字符片段被视为是频繁的；否则，候选字符片段为非频繁的，应将其丢弃。筛选过程中首先对 1-字符片段集合 N_1 开始频繁性检测，然后对 2-字符片段集合 N_2 进行新一轮频繁性检测，以此类推，直至对最长字符片段集合进行频繁性检测后结束。对于 k-字符片段集合 N_k，详细的频繁性检测步骤如下：

1)对候选字符片段(长度为 m)进行频繁性检测，如果该候选字符片段的支持度不小于支持度阈值，那么该候选字符片段是频繁的；否则，该候选字符片段不是频繁的，应该被丢弃。

2)如果候选字符片段经过步骤 1)检测后确认不是频繁的，那么由序列模式挖掘的先验原理[17]可知，该候选字符片段的超序列也不是频繁的。因此，在长度大于 m 的字符片段集合内属于候选字符片段的超序列都不是频繁的，在后续检测过程中不再对其进行频繁性检测。

依次重复执行上述筛选过程，直至集合 N_k 中所有的候选字符片段都检测完毕，最终生成一个完整的 k 长度频繁的字符片段集合 $F_k=\{f_{k1},f_{k2},\cdots,f_{kn}\}$。该集合与前一轮检测所生成的$(k-1)$长度频繁的字符片段集合 $F_{k-1}=\{f_{(k-1)1},f_{(k-1)2},\cdots,f_{(k-1)n}\}(k\geqslant 2)$一起作为闭合检测阶段的输入，用来判断$(k-1)$长度频繁的字符片段是否闭合。

Step3 闭合性检测。实施闭合性检测判定的依据是：如果字符片段的超序列支持度均小于该片段的支持度，那么该片段符合闭合性要求。

以字符片段集合 F_{k-1} 和 $F_k(k\geqslant 2)$为例，在进行闭合性检测判定时，在 F_{k-1} 中任何一个频繁的字符片段 f_{k-1} 是闭合的，当且仅当 F_k 不存在包含并且支持度大于或等于 f_{k-1} 的 f_k。对于 F_{k-1} 中的任意某个频繁字符片段 f_{k-1}，判断 F_k 内是否存在 f_k 满足前子序列(删除字符片段的最后一个字符得到其前子序列)以及后子序列(删除字符片段的第一个字符得到其后子序列)与 f_{k-1} 一致以及它们的实际支持度相等，如果存在 f_k 满足条件，那么 f_k 包含 f_{k-1}，f_{k-1} 是非闭合的，否则 f_{k-1} 是闭合的。对集合 F_{k-1} 中的所有元素均按此方法进行判断，那么集合 F_{k-1} 中所有闭合片段是否闭合都能够被识别出来，从而可获得所有$(k-1)$长度的闭合邻接频繁段；再以字符片段集合 F_k 和 F_{k+1} 作为输入，继续依据此方法判断，从而获得所有 k 长度的闭合邻接频繁段。以此类推，最终获得所有的闭合邻接频繁段。

以上是分割阶段的处理流程，该阶段获得闭合频繁邻接段的集合，其中存在很小部分的闭合频繁邻接段不是关键字，还需依据关键字识别策略来进一步确认关键字；再以关键字为基础，将报文划分为“关键字字段＋变量字段”的形式来作为下一阶段的输入。

4.3.2 模式挖掘阶段

闭合序列模式挖掘阶段的主要工作是生成由多个闭合频繁序列构成的闭合序列模式 F_{C_L}。F_{C_L} 内的每个闭合频繁序列 θ_s 是关键字序列，由于关键字间隔长度可变，θ_s 内每个 s 的间隔长度也是可变的，因此相比于第一阶段，第二阶段挖掘的目标不再具有邻接属性，并且该阶段是以关键字作为字段划分基础的已分段报文，需要处理的数据规模变小，可直接采用已有的通用闭合序列模式挖掘算法。

该阶段选择的闭合序列模式挖掘算法是 BIDE 算法。BIDE 算法由 Wang 等提出[21]，它是挖掘闭合序列模式的一种高效算法。BIDE 算法基于双向扩展的闭合序列检查技术以及向后扫描搜索空间的剪枝技术，可以有效挖掘闭合序列模式。该算法在挖掘闭合序列的过程中，在内存中不需要维护以往的频繁闭合项，能够节省内存空间和运行时间。在该阶段的处理过程中，以报文“关键字字段＋变量字段”的分段形式作为输入，算法为每个前缀$\langle s_i,\cdots,s_j\rangle$建立伪投影库，计算它的向前扩展项的个数，并循环调用子 bide 算法，计算向后扩展项的个数，如果$\langle s_i,\cdots,s_j\rangle$既没有向前扩展项也没有向后扩展项，则$\langle s_i,\cdots,s_j\rangle$是闭合频繁序列$\theta_s$。最终获得报文组$C_L$对应的闭合序列模式$F_{C_L}$。文献[21]中已经详细地描述了 BIDE 算法的执行过程，限于篇幅，本文不再赘述。

采用两阶段算法的优点是第一阶段使用邻接属性，减小了搜索范围；第二阶段算法的操作单元是闭合频繁段，在获取频繁序列时，序列长度的每一次增长是一个闭合频繁邻接段s而非一个字符e。这种方法利用第一阶段的分段处理减小了第二阶段数据处理的规模，提高了计算效率，可以使闭合序列模式挖掘方法适用于较大规模的报文样本分析。第二阶段获取的关键字序列保留了关键字之间的组合关系信息，为下一步的报文结构推断奠定了基础。需要指出的是，该方法提取的关键字序列作为协议特征可应用于协议识别领域，本文精简、高效的提取过程为协议识别提供了一种思路。

4.4　含噪音的报文样本的关键字识别

在协议逆向分析过程中，所捕获的报文样本中主要包含两种类型的噪音：1）归属于其他协议的报文，这类噪音在报文捕获阶段混入，它与目标协议并没有直接联系；2）完整性缺失的协议报文，这类报文包含的信息不完整。这两类噪音对关键字识别都是不利的。

为了方便实验验证，现有的多数协议逆向方法选取的分析协议通常是已知协议，实验时依据先验知识对报文进行预处理，并将获得的无噪音报文作为实验样本。然而，在实际协议逆向分析时，无法获取未知协议的特征，无论是按协议特征分类，还是按五元组进行处理，都无法保证得到纯净的报文样本，很多关键字识别方法在分析含有噪音的报文样本时效果都不理想[14]。在实际网络环境中，理想的协议逆向方法中关键字识别应当允许样本中存在噪音，而不需要将目标协议报文与噪音报文严格区分开来，同时仍可以达到较高的关键字识别准确率。

本文提出的关键字识别方法在报文组内挖掘闭合频繁邻接段，然后依据关键字识别策略判定闭合频繁邻接段是否为关键字。这种方法对报文样本的要求相对宽松，在报文组中存在噪音的情况下可以准确识别关键字。因为闭合频繁邻接段在报文组中的出现频率不小于设定的最小支持度阈值，而噪音在报文中往往是低概率地出现，所以报文中的噪音不会影响闭合频繁邻接段的挖掘。此外，本文的关键字识别策略针对闭合频繁邻接段的位置或包含的信息来确定协议关键字，并不要求关键字在每条报文中都出现，噪音的存在不会影响关键字的识别。

5　报文结构的推断

报文结构的推断阶段是以关键字识别阶段获得的闭合序列模式F_{C_L}为分析对象，识别不同关键字之间的顺序、并列以及层次关系，最终获得协议格式的巴克斯范式。

5.1　关键字间顺序关系与并列关系的分析

报文组C_L获取的F_{C_L}内闭合频繁序列θ_s是确定为顺序关系的关键字序列。由于每个关键字序列不包含重复关键字，为了方便进行集合的运算操作，将每个关键字序列θ_s映射成关键字的集合，记作θ_s'。例如，关键字序列$\theta_s=\langle\langle GET\rangle,\langle HTTP/1.0\rangle,\langle Host:\rangle\rangle$映射成$\theta_s'=\{\langle GET\rangle,\langle HTTP/1.0\rangle,\langle Host:\rangle\}$。将报文组$C_L$对应的所有$\theta_s'$组成的集合记作$F'_{C_L}$，那么$F'_{C_L}$内所有$\theta_s'$的交集可记作：

$$\bigcap F'_{C_L}=\theta'_{s1}\bigcap\theta'_{s2}\bigcap\cdots\bigcap\theta'_{s3} \tag{1}$$

由式(1)可知，$\bigcap F'_{C_L}$是在报文组C_L内具有顺序关系的关键字集合。例如，表 1 对应的$\bigcap F'_{C_L}=\{\langle GET\rangle,\langle HTTP/1.0\rangle\}$，其中关键字“GET”与“HTTP/1.0”具有严格的顺序关系。

F'_{C_L}内所有θ_s'的并集可记作：

$$\bigcup F'_{C_L}=\theta'_{s1}\bigcup\theta'_{s2}\bigcup\cdots\bigcup\theta'_{s3} \tag{2}$$

由式(2)可知，$\bigcup F'_{C_L}$是报文组C_L所有关键字的集合。$\bigcup F'_{C_L}$与$\bigcap F'_{C_L}$的差集ψ为候选并列关系的关键字集合，ψ内关键字之间可能是并列关系，也可能非并列关系，具体关系仍需要进一步判断。判断的基本思想是选取任意的关键字组合，由于闭合频繁序列是满足顺序关系的关键字，因此如果组合内关键字可同时出现在一条报文L上，并且不同时出现在一个闭合频繁序列上，那么它们才是并列关系。例如，表 1 中报文组对应的关键字“Host:”与“User-Agent:”同时出现在同一条报文L上，但是没有同时出现在任何闭合频繁序列θ_s上，则可认为“Host:”与“User-Agent:”之间满足并列关系。推断关键字之间的并列关系的具体流程主要包括 3 步。

Step1　构造关键字组合。集合ψ内的关键字之间的任意组合作为一个元素构成新的集合ψ_{PS}。

Step2　排除不合法关键字组合。对于任意$m_{PS}\in\psi_{PS}$，遍历一次报文组，如果没有一条报文同时包含m_{PS}内的各个关键字，则m_{PS}属于不合法的关键字组合。m_{PS}的任意超集$m_{Superset}$也是不合法关键字组合，这是因为$m_{Superset}$中会包含m_{PS}涉及的关键字组合。

Step3　排除具有顺序关系的关键字组合。如果关键字集合m_{PS}属于某个闭合频繁序列，则关键字集合m_{PS}所涉及的关键字不具有并列关系，同时，这些m_{PS}的超集$m_{Superset}$也不具有并列关系；否则，m_{PS}内的关键字满足并列关系。对于满足并列关系的关键字组合m_{PS}，如果ψ_{PS}内有其他元素是m_{PS}的子集，那么这些子集应该从ψ_{PS}中删除，由于m_{PS}的子集中的关键字尽管满足并列关系，但不是并列关系关键字的最大组合，因此不需要重复记录。

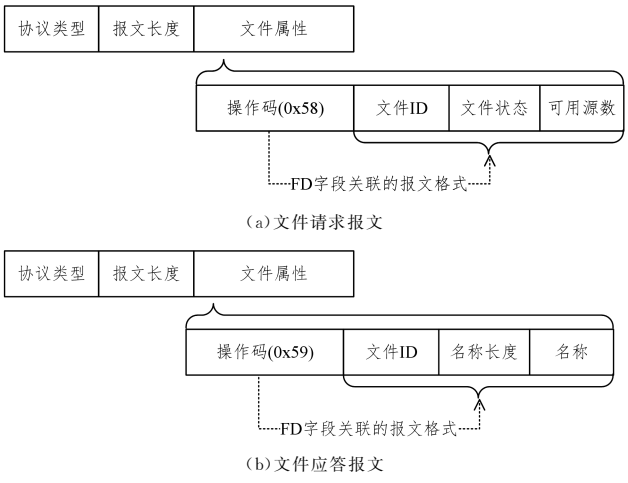
按照上述方法依次排除不符合要求的关键字组合，最终

集合 ψ_{PS} 中剩余的关键字组合具有并列关系。

通过以上推断，可得到报文组中具有顺序关系的关键字集合 $F_{Key_{Seq}}=\bigcap F'_{C_L}=\{K_i,K_{i+1},\cdots\}$，关键字 K_i 在报文上必出现在 K_{i+1} 前，其用 BNF 范式表示为 $K_iK_{i+1}\cdots$；同时得到以满足并列关系关键字组合为元素的集合 $F_{Key_{Par}}=\{\{K_n,K_{n+1},\cdots\},\cdots\}$，集合元素为并列关系关键字的组合，例如元素 $\{K_n,K_{n+1},\cdots\}$ 内的关键字 K_n 与 K_{n+1} 满足并列关系，其用 BNF 范式可表示为 $K_n|K_{n+1}|\cdots$。

5.2 关键字间层次关系的分析

协议报文中，格式标识字段(Format Distinguisher，FD)与后续的报文格式紧密关联，但 FD 字段的取值不同，后续的报文格式(由若干个关键字组成)也不一样。在层次化的报文结构中，通常由 FD 字段与其后续的报文格式组成报文的子结构[6]。如图 3 所示，eMule 协议中 FD 字段(操作码)为关键字“0×58”时，其后续关联的报文格式为“文件 ID”“文件状态”等关键字；而当 FD 字段为 0×59 时，其后续关联的报文格式却为“文件 ID”“名称长度”等关键字。可以将关键字“0×58”与关联的报文格式组合作为子结构，形成层次化结构的报文。



(a)文件请求报文

(b)文件应答报文

图 3 eMule 协议报文的示例

Fig. 3 Example of eMule protocol message

关键字间的层次关系分析的目的是判断报文中哪些关键字属于 FD 字段，并分析 FD 字段后续关联的报文格式。以 FD 字段作为层次结构的开始标识，关联的报文格式的尾部位置作为结束标识，进而分析协议关键字之间的层次关系。

由 FD 字段的特点可知存在 $K_{FD}\leftrightarrow h$ 的表达形式，其中 K_{FD} 字段为 FD 字段，h 是所关联的报文格式(h 为$\langle K_n,\cdots,K_m\rangle$)，意味着报文中 FD 字段与对应关联的报文格式 h 通常成对出现。从置信度的角度看，通过设定合理的最小置信度阈值 Min_conf，FD 字段与 h 将满足公式：

$$Conf_{C_L}(K_{FD}\rightarrow h)\geqslant Min\_conf \tag{3}$$

$$Conf_{C_L}(h\rightarrow K_{FD})\geqslant Min\_conf \tag{4}$$

为了判断报文中哪些关键字是 FD 字段，本文基于报文组 C_L 获取的 F_{C_L} 搜索满足条件：

$$Conf_{C_L}(K_i\rightarrow\langle K_n,\cdots,K_m\rangle)\geqslant Min\_conf \tag{5}$$

$$Conf_{C_L}(\langle K_n,\cdots,K_m\rangle\rightarrow K_i)\geqslant Min\_conf \tag{6}$$

其中，K_i 视为 FD 字段，$\langle K_n,\cdots,K_m\rangle$ 视为报文格式 h。搜索过程采用序列模式挖掘中的关联分析算法[22]，限于篇幅，搜索过程不再赘述。

采用上述搜索方式可确认报文组 C_L 内所有 FD 字段的集合 $F_{Key_{FD}}=\{K_{FD_1},K_{FD_2},\cdots\}$，假设 K_{FD_1} 关联的报文格式是$\langle K_p,K_{p+1},K_{p+2}\rangle$，关键字 K_{FD_1} 与 K_p,K_{p+1},K_{p+2} 组成字段 K_s，K_s 与 K_{FD_1}、K_p 与 K_{p+1} 等具有层次关系，其用 BNF 范式表示成 $K_s::=\{K_{FD_1}K_pK_{p+1}K_{p+2}\}$。

5.3 报文结构推断算法

基于上文的相关描述，算法 1 给出报文结构推断的具体过程。

算法 1 报文结构推断算法

输入：C_L 的闭合序列模式 F_{C_L}，最小支持度阈值 Min_Sup，最小置信度阈值 Min_Conf，由 F_{C_L} 转换的集合 F'_{C_L}

输出：ϕ_{BNF}

1. $\bigcup F'_{C_L}=\theta'_{s1}\bigcup\theta'_{s2}\bigcup,\cdots,\bigcup\theta'_{sn},\psi=\varnothing,\xi=\varnothing$
2. $F_{Key_{Seq}}=\bigcap F'_{C_L}=\theta'_{s1}\bigcap\theta'_{s2}\bigcap\cdots\bigcap\theta'_{sn}$
3. for θ'_{si} in F'_{C_L}:
4. $\upsilon_i=\bigcup F'_{C_L}-\theta'_{si}\in\xi,\psi=\psi\bigcup\upsilon_i$//构造集合 ψ
5. end for
6. $2^{\psi}=\{x|x\subseteq\psi\},\psi_{PS}=2^{\psi}-\xi-\{\varnothing\}$
//构造集合 ψ_{PS}
7. for m_{PS} in ψ_{PS}://遍历集合 ψ_{PS}
8. if($\neg\exists L,m_{PS}$in L) or($\exists\theta_{si}\in F_{C_L},m_{PS}\subseteq\theta_{si}$):
9. $m_{PS},m_{Superset}\in\eta$
10. else:
11. $m_{Subset}\in\eta$
12. end for
13. $F_{Key_{Par}}=\psi_{PS}-\eta$//构造并列关键字组合的集合
14. find $Conf_{C_L}6(K_i\rightarrow\langle K_n,\cdots,K_m\rangle)\geqslant Min\_conf$
and $Conf_{C_L}(K_i\rightarrow\langle K_n,\cdots,K_m\rangle)\geqslant Min\_conf$:
15. $K_i\in F_{Key_{FD}},\langle K_n,\cdots,K_m\rangle\in F_h$
//构造 FD 关键字的集合 $F_{Key_{FD}}$
16. ReStructure($F_{Key_{Seq}},F_{Key_{Par}},F_{Key_{FD}},F_h$)
18. return ϕ_{BNF}//返回报文对应的 BNF 范式

如算法 1 所示，报文结构的推断主要包括 5 个步骤。

Step1 算法 1－2 行计算 F'_{C_L} 内所有 θ'_s 的交集，即 $\bigcap F'_{C_L}=\theta'_{s1}\bigcap\theta'_{s2}\bigcap\cdots\bigcap\theta'_{sn}$，$\bigcap F'_{C_L}$ 为顺序关系关键字的集合 $F_{Key_{Seq}}$。

Step2 算法 3－5 行首先构造 $\bigcup F'_{C_L}$ 与 $\bigcap F'_{C_L}$ 的差集 ψ 并将其作为候选并列关系的关键字，再构造以候选并列关系的关键字组合为元素的集合 ψ_{PS}。

Step3 算法 6－13 行遍历集合 ψ_{PS}，依次排除集合 ψ_{PS} 内不符合并列关系条件的关键字组合的元素，最终获得集合 $F_{Key_{Par}}$ 内元素是满足并列关系关键字组合的最大集合。

Step4 算法 14－15 行获取 F_{C_L} 内满足最小置信度阈值的关键字 K_{FD} 与 h，最后得到 K_{FD} 的集合 $F_{Key_{FD}}$ 与相关联的报文格式 h 的集合 F_h。

Step5 算法 16－18 行调用 *ReStructure* 函数，函数输入

的参数是 $F_{Key_{Seq}}$，$F_{Key_{Par}}$，$F_{Key_{FD}}$ 与 F_h。*ReStructure* 函数实现报文序列的递归遍历，$F_{Key_{Seq}}$ 内各个关键字之间满足顺序关系关键字，$F_{Key_{Par}}$ 内各个关键字之间满足并列关系；以 $F_{Key_{FD}}$ 内的 FD 字段作为层次开始标识，将 FD 字段相关联的报文格式的尾部位置作为结束标识，进而逐层提取层次化的报文结构。由左至右遍历关键字序列，直到关键字序列结束为止。最终，函数返回包含关键字的顺序关系、并列关系、层次关系的报文结构的 BNF 范式，并最终确认完整的报文结构。

6　实验结果与分析

为了验证所提出的协议格式推断方法的有效性，主要从关键字识别、抗噪声干扰能力以及报文结构推断效果 3 方面进行实验验证。

6.1　实验准备

实验中涉及的改进 CCSpan 算法基于 C＃语言开发，BIDE 算法由开源数据挖掘 SPMF 平台[23]提供，关联分析算法由 Weka 平台[24]提供。为保证报文样本的多样性，样本主要由 3 部分组成：1）MACCDC 数据集[25]；2）DARPA 数据集[26]；3）实际网络环境捕获的报文样本。选取其中具有代表性的协议，文本类型有 HTTP 协议、FTP 协议与 SMTP 协议，二进制协议有 SMB 协议、DNS 协议与 eMule 协议，报文样本的统计如表 4 所列。

表 4　报文样本的统计信息

Table 4　Statistical information of messages samples

协议类型	会话个数/个	报文条数/条	数据集来源
HTTP	1000	23366	实际网络环境捕获
FTP	650	10404	DARPA
SMTP	1000	14135	DARPA
SMB	2500	29130	MACCDC
DNS	2000	4630	MACCDC 与实际网络环境捕获
eMule	1000	54520	实际网络环境捕获

6.2　关键字识别效果的评估

采用准确率、召回率与 $F1$ 值作为关键字识别效果的评估指标。准确率为 $\rho=\dfrac{|TP|}{|TP|+|FP|}$，召回率为 $\gamma=\dfrac{|TP|}{|TP|+|FN|}$，$|TP|$ 是指被正确识别的协议关键字数量，$|FP|$ 是指被错误识别的关键字数量，$|FN|$ 是指没有被识别的协议关键字数量。

准确率 ρ 指被正确识别的关键字数量占被识别的协议关键字总数的比例。召回率 γ 指被正确识别的协议关键字数量占实际协议关键字总数的比例。$F1=\dfrac{2\rho\gamma}{\rho+\gamma}$，$F1$ 值综合衡量了准确率和召回率。需要说明：计算准确率和召回率时，在实验样本中没有出现过的协议关键字不做考虑，真实关键字指的是实验报文样本中出现的协议关键字。

实验结果的 $F1$ 值如图 4 所示，当 *Min_Sup* 为 0.8 与 0.9 时，本文方法对应的 $F1$ 比 Netzob，AutoEngine 与 Discoverer 方法的 $F1$ 值都高，这意味着本文方法的协议关键字识别结果好于其他对比方法。

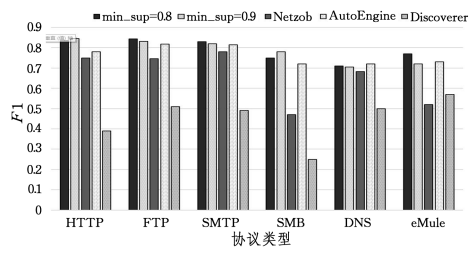


图 4　关键字识别效果的对比

Fig. 4　Effect comparison of keyword identification

设置 *Min_Sup* 为 0.8 时与 *Min_Sup* 为 0.9 时，方法的 $F1$ 值相差不大。主要原因是前者可能会错误识别关键字，导致对应的准确率 ρ 相对较低，而后者可能会漏掉关键字，召回率 γ 相对较低，但两者对应的 $F1$ 值相对稳定并且较高。由实验经验可知，当 *Min_Sup* 的取值范围为 0.7～0.9 时，可保证较高的关键字提取准确率与召回率。

Netzob 的 $F1$ 值比本文方法的 $F1$ 值低，因为 Netzob 识别关键字时是基于序列比对算法，该算法要求数据本身应该具有整齐性。Netzob 对多数协议进行分析时，可以输入分隔符。例如，HTTP 协议以“空格”作为分隔符，使用分隔符对报文分段，段与段之间采用多序列比对算法来识别关键字，但是 Data 字段长度变化大，算法无法准确地对齐字节；此外，对于无法确定分隔符的协议，算法将整条报文作为多序列比对算法的输入，同样存在无法准确对齐字节的问题。虽然 Netzob 方法的准确率 ρ 较高，但是其召回率 γ 较低，从而导致对应的 $F1$ 值相对较低。

在分析关键字在报文中出现的位置一致的协议时，AutoEngine 方法的 $F1$ 值较高；但是当存在位置变化较大的关键字时，例如位置可变的并列关系关键字，AutoEngine 对应的召回率 γ 较低，导致 $F1$ 值相对较低。Discoverer 递归地将 Token 聚类，然后在每一个子类中将频繁的 Token 作为协议关键字；但是存在 Token 被聚类后，在子类中变成频繁项的问题，此时过多的 Token 被错当成关键字，从而导致准确率 ρ 较低，对应的 $F1$ 值也较低。

本文的关键字识别方法中，分割阶段算法的计算开销远大于模式挖掘阶段算法的开销。分割阶段的时间复杂度为 $O(n)$，其中 $n=\sum_{i=1}^{|C_L|} len(L_i)$ 是报文组中报文的总长度，时间复杂度与报文序列的总长度线性相关；内存消耗方面，在整个挖掘过程中，无须将整个报文序列集加载到内存中，在任意时间只有一条报文位于内存中，这有助于内存的分配与释放，使方法可以满足较大规模报文样本的分析。

6.3　抗噪能力的评估

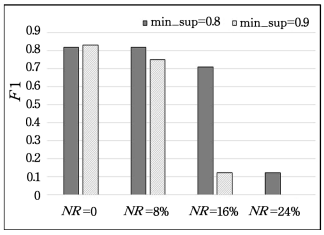
目前，基于网络流量的协议逆向分析很少考虑到噪音的影响，也没有公开的测试数据集。本实验在目标协议纯净的样本集中添加不同类型以及不同比例的噪音，以测试并分析方法的抗噪能力，样本中目的协议的报文数量与报文总数的比值称为噪音比例（Noise Ratio，NR）。

首先，在文本协议 HTTP 的数据集中随机选取 1000 条已聚类的报文序列作为纯净报文，然后从其他协议的数据集

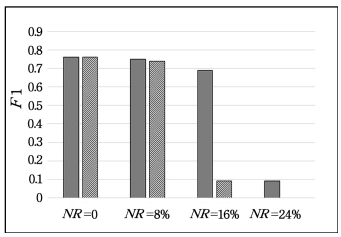
中随机选取 80 条、160 条、240 条报文作为噪音并替换对应条数的纯净报文，构造噪音比例分别是 8%，16%与 24%的报文样本。对二进制协议 eMule 采用同样的方法构造报文样本。

使用 $F1$ 值衡量关键字的识别效果，在 HTTP 与 eMule 协议的 5 种噪音比例情况下，最小支持度阈值 Min_Sup 分别为 0.8 与 0.9 时对应的 $F1$ 值直方图分布如图 5 所示。

实验证明，本文方法有较强的抗噪能力。从图 5 中可以看出，在混入不同比例噪音后，不同最小支持度阈值条件下关键字识别的 $F1$ 值不同。在噪音比例不大于 8%时，两种最小支持度阈值均以较高的准确度识别出关键字，具有一定的抗噪能力。



(a) HTTP Protocol



(b) eMule Protocol

图 5 抗噪能力的评估统计图

Fig. 5 Anti-noise property evaluation statistics

本文方法在最小支持度阈值 Min_Sup 为 0.8 时的抗噪能力比 Min_Sup 为 0.9 时的抗噪能力更强，表现在当报文 $NR=16\%$即为相对纯净的报文集时，$Min\_Sup=0.8$ 时的 $F1$ 值虽然略有下降，但是仍保持较高，关键字识别的效果依然可观；而当 $Min\_Sup=0.9$ 时，$F1$ 值大幅度下降。这是因为噪音协议已占报文样本总量的 16%，目标协议占报文样本总量的 84%。当 $Min\_Sup=0.8$ 时，关键字所在目标协议占报文总量大于 80%即可被识别。而 $Min\_Sup=0.9$ 时，因为需要关键字所在目标协议占报文样本总量的比例大于 90%，所以实验中关键字识别的 γ 值低，此时 $F1$ 值很低。尽管 Min_Sup 设置得越小抗噪能力越强，但是 Min_Sup 设置过小时关键字提取的准确率 ρ 也会过低，从而导致 $F1$ 值较低。因此，不同协议与不同噪音比例情况下的最小支持度阈值视情况而定。根据实验经验，对于实际网络环境中截取的报文，当 Min_Sup 的取值范围是 0.7～0.9 时 $F1$ 值较高，关键字识别的效果较好。

6.4 报文结构推断的准确率评估

报文结构推断的准确率以树编辑距离(Tree Edit Distance)[27]为指标，主要评估提取结果与真实报文结构的差异。推断的报文结构树与真实报文结构树之间的树编辑距离为通过删除、插入等操作将推断的报文结构树转化为真实的报文结构树所需的步骤数。设定最小置信度阈值 $Conf\_Sup$ 为 0.9，表 5 列出报文结构推断结果与真实结构之间的树编辑距离。

表 5 报文结构推断结果的统计

Table 5 Statistical table of messages structure inference results

报文类型	1000 条	2000 条	4000 条
HTTP	3.76	3.58	2.02
FTP	0.86	0.81	0.77
SMTP	0.93	0.82	0.80
SMB	9.6	7.93	7.07
DNS	0.12	0.11	0.11
eMule	2.88	2.72	2.70

从表 5 可以看出，本文方法推断结果的树编辑距离较小，说明其能够较为准确地推断报文结构信息；此外，报文样本规模越大，树编辑距离越小，推断结果更准确。需要指出的是，对于结构单一的 DNS 协议，树编辑距离明显小于复杂结构的 HTTP 协议或 eMule 协议的树编辑距离。由于在 HTTP 协议结构的推断过程中，提取 FD 字段关联的报文格式存在少量的冗余关键字，报文结构推断存在误差。eMule 协议有多级层次结构，在深层的结构推断中存在误差。对于 SMB 协议，由于关键字识别存在较高误差，因此结构推断存在误差。

结束语 本文提出一种基于闭合序列模式挖掘的关键字识别与报文结构推断方法，该方法不要求报文数据结构严格整齐，对存在位置浮动的关键字的报文样本也有较好的关键字识别效果；通过设计两阶段闭合序列挖掘策略，使闭合序列模式挖掘方法可适用于报文样本分析，具有较强的抗噪能力，并且能够准确提取报文结构。由实验结果可知，所提方法进行关键字识别时具有较高的 $F1$ 值，结构推断具有较小的树编辑距离，不论是理论分析还是实验结果均好于对比方法。下一步将研究如何基于关键字序列推断字段语义，以及推导协议的状态机，从而实现完整的协议逆向工程。

参 考 文 献

[1] 吴礼发，洪征，潘璠. 网络协议逆向分析及应用[M]. 北京：国防工业出版社，2016：10-13.

[2] DUCHÊNE J，GUERNIC C L，ALATA E，et al. State of the art of network protocol reverse engineering tools[J]. Journal of Computer Virology and Hacking Techniques，2017，14(2)：1-16.

[3] NARAYAN J，SHUKLA S K，CLANCY T C. A Survey of Automatic Protocol Reverse Engineering Tools[J]. Acm Computing Surveys，2015，48(3)：1-26.

[4] LUO J Z，YU S Z. Position-based automatic reverse engineering of network protocols[J]. Journal of Network & Computer Applications，2013，36(3)：1070-1077.

[5] ZHANG Z，ZHANG Z，LEE P P C，et al. ProWord：An unsupervised approach to protocol feature word extraction[C]// INFOCOM，2014 Proceedings IEEE. IEEE，2014：1393-1401.

[6] CAI J，LUO J Z，LEI F. Analyzing network protocols of application layer using hidden semi-Markov model[J]. Mathematical Problems in Engineering，2016，2016：1-14.

[7] LUO J Z，YU S Z，CAI J. Method for determining the lengths of

protocol keywords based on maximum likelihood probability[J]. Journal of Software, 2016, 37(6): 119-128. (in Chinese)
罗建桢，余顺争，蔡君. 基于最大似然概率的协议关键词长度确定方法[J]. 通信学报，2016，37(6)：119-128.

[8] CUI W, KANNAN J, WANG H J. Discoverer: Automatic Protocol Reverse Engineering from Network Traces[C]// Proedings of the 16th USENIX Security Symposium. Berkeley: ACM, 2007: 1-14.

[9] BEDDOEM M. Protocol Information Project. [EB/OL]. (2004-10-5) [2018-01-20]. http://www. 4tphi. net/～awalters/PI/PI. html.

[10] BOSSERT G, HIET G, HENIN T. Modelling to simulate botnet command and control protocols for thee valuation of network intrusion detection systems[C]// 2011 Conference on Network and Information Systems Security (SAR-SSI). La Rochelle: IEEE, 2011: 1-8.

[11] KRUEGER T, KRAEMER N. PRISMA: Protocol Inspection and State Machine Analysis[J]. Journal of the American Chemical Society, 2015, 98(25): 8101-8107.

[12] LIN Z, JIANG X, XU D, et al. Automatic Protocol Format Reverse Engineering through Context-Aware Monitored Execution [C]// Network and Distributed System Security Symposium, NDSS 2008. San Diego, California, USA, DBLP, 2008.

[13] 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.

[14] LI M, YU Z S. Noise-Tolerant and Optimal Segmentation of Message Formats for Unknown Application-Layer Protocols [J]. Journal of Software, 2013, 24(3): 604-617. (in Chinese)
黎敏，余顺争. 抗噪的未知应用层协议报文格式最佳分段方法[J]. 软件学报，2013，24(3)：604-617.

[15] WANG Z, JIANG X, CUI W, et al. ReFormat: Automatic reverse engineering of encrypted messages[C]// Proc of the 14th European Conf on Research in Computer Security [S. l.]: Springer, 2010: 200-215.

[16] CROKER D, OVERELL P. Augmented BNF for syntax specifications: ABNF [R/OL]. http://tools. ietf. org/html/rfc4234.

[17] YAN X, HAN J, AFSHAR R. CloSpan: Mining Closed Sequential Patterns in Large Databases[C]// Siam International Conference on Data Mining. San Francisco, CA, USA, DBLP, 2003: 166-177.

[18] 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. (in Chinese)
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法[J]. 计算机学报，2011，34(2)：242-255.

[19] ZHANG J, WANG Y, YANG D. CCSpan: Mining closed contiguous sequential patterns[J]. Knowledge-Based Systems, 2015, 89: 1-13.

[20] BROWN P F, DESOUZA P V, MERCER R L, et al. Class-based n-gram models of natural language[J]. Computational Linguistics, 1990, 18(4): 467-479.

[21] WANG J, HAN J. BIDE: Efficient Mining of Frequent Closed Sequences[C]// International Conference on Data Engineering, 2004. IEEE, 2004: 79-90.

[22] ADAMO J M. Data Mining for Association Rules and Sequential Patterns[M]. Berlin: Springer, 2001.

[23] FOURNIER-VIGER P, GOMARIZ A, GUENICHE T, et al. SPMF: a Java open-source pattern mining library[J]. Journal of Machine Learning Research, 2014, 15(1): 3389-3393.

[24] HOLMES G, DONKIN A, WITTEN I H. WEKA: a machine learning workbench[C]// Proceedings of the 1994 Second Australian and New Zealand Conference on Intelligent Information Systems, 1994. IEEE, 2002: 357-361.

[25] NETRESEC. MACCDC traces[EB/OL]. [2017-10-16]. http://www. netresec. com/?page=MACCDC.

[26] MAHONEY M V, CHAN P K. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection[C]// Vinga G, ed. Proc. of the 6th Symp. on Recent Advances in Intrusion detection. Berlin, Heidelberg: Springer-Verlag, 2003: 220-237.

[27] KLEIN P N. Computing the edit-distance between unrooted ordered trees[C]// Proceedings of the 6th annual European Symposium on Algorithms. Berlin: Springer-Verlag, 1998: 91-102.