计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20210654
58(11):2400-2415,2021

一种面向 IPv6 网络空间的特征水印生成与嵌入方案研究

陶 军[1,2,3] 朱珍超[1,2,3] 王昭悦[1] 李文强[1,2] 孙炜策[1,2]
[1](东南大学网络空间安全学院 南京 211189)
[2](计算机网络和信息集成教育部重点实验室(东南大学) 南京 211189)
[3](网络通信与安全紫金山实验室 南京 100084)
(wqli@seu.edu.cn)

A Feature Watermarking Generation and Embedding Scheme for IPv6 Network

Tao Jun[1,2,3], Zhu Zhenchao[1,2,3], Wang Zhaoyue[1], Li Wenqiang[1,2], and Sun Weice[1,2]
[1](*School of Cyber Science and Engineering*, *Southeast University*, *Nanjing* 211189)
[2](*Key Laboratory of Computer Network and Information Intergration* (*Southeast University*), *Ministry of Education*, *Nanjing* 211189)
[3](*Purple Mountain Laboratories for Network Communication and Security*, *Nanjing* 100084)

Abstract Under the limitation of space and time resources, researchers exploit the network covert channel, which based on a small amount of watermark information, to trace the attack flow and locate the real attack source. However, the self-similarity of the tracked traffic would appear because of the relatively fixed content and location of the watermark. What's more, the IPSec encryption protocol embedded in the IPv6 protocol limits the range of carrier choice, which may threaten the watermarking based on the single carrier. In this paper, Targeting at optimizing the watermark invisibility, combined with intermediate node not dividing the packet for IPv6 environment, considering the feature extraction limitation of intermittent transmission network and slow flow network, the feature watermarking sequence extraction strategy associated with the target stream is designed. Aiming at different network transmission scenarios, a packet-dependent watermarking generation based on mixed covert channel and a time-dependent watermarking generation based on mixed time slot are proposed. Extensive experiments show that the watermarking generation technology proposed in this paper can reduce the impact of watermarking embedding on the original traffic, decrease the probability of watermarking being recognized and attack, and improve the imperceptibility of watermark under the premise of certain accuracy.

Key words feature watermarking; covert channel; mixed interval; IPv6 protocol; flow correlation analysis

摘 要 在有限的时空资源条件下,研究人员使用网络隐蔽通道,基于少量的水印信息来追踪攻击流,定位真实攻击源.然而,水印内容和位置的相对固定会造成追踪的流量呈现出自相似性,并且 IPv6 协议内嵌的 IPsec 加密协议限制了载体的选择范围,基于单一载体的水印嵌入方案更容易被识别攻击.因此针对

收稿日期:2021-06-11;**修回日期**:2021-08-13
基金项目:国家重点研发计划项目(2018YFB1800205);中央高校基本科研业务费专项资金(2242021k30024);阿里云高校数字化创新专项(2021ALA03006);江苏省研究生研究与实践创新计划(KYCX180103)
This work was supported by the National Key Research and Development Program of China (2018YFB1800205), the Fundamental Research Funds for the Central Universities (2242021k30024), Alibaba Cloud College Digital Innovation Project (2021ALA03006), and the Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX180103).

水印隐蔽性的优化目标,结合 IPv6 报文中间节点不分片的特性,考虑间断性传输网络和流速较慢网络的特征提取限制,设计目标流关联的特征水印序列提取策略,针对不同的网络传输场景,制定了包依赖的基于混合隐蔽通道和时间依赖的基于混合时隙的水印嵌入方式.模拟实验表明:提出的水印生成与嵌入技术,能够在保证一定准确率的前提下,降低水印嵌入对原始流量的影响,减少水印被识别攻击的概率,提高水印的隐蔽性.

关键词　特征水印;隐蔽通道;混合时隙;IPv6 协议;流关联分析

中图法分类号　TP393

随着信息化的快速发展,网络早已融入了人们的日常生活之中.但是,网络是把双刃剑,在人们享受其所带来的便利时,个人利益也可能会受到损害.《2020 年上半年我国互联网网络安全监测数据分析报告》显示,半年时间内,恶意计算机程序的检测数量大约有 1 815 万个,涉及到约 11 000 个恶意程序家族,日均约 483 万次传播[1].仅仅在中国境内,被恶意程序攻击的主机数量就超过了 304 万台,同比增长 25.7%.其中 DDoS 攻击作为攻击者最常选择的攻击手段之一,在我国每天大约发生 220 起,其峰值流量超过 10 Gbps.在目前 IPv4 地址资源日趋耗尽的背景下,拥有更大地址空间范围的 IPv6 协议势必会逐渐取而代之,成为未来的主流网络层通信协议[2].在这种背景下,IPv6 环境下的安全问题也成为了新的挑战[3].

网络安全事故频繁发生,对金融、医疗、政府、教育、制造业等各个行业都造成了一定程度的威胁.早期对打击网络犯罪技术的研究更多着重于被动网络流分析,通过对网络流的特征分析获取到有效的信息.比如,使用统计方法或机器学习方法,定义应用协议类型,预测用户位置[4],区分正常流量和异常流量[5]以及识别恶意攻击[6].但被动网络流分析技术需要大量的样本来确保机器学习模型的有效性,在时间和空间的资源需求上非常严格,很难在可扩展性和准确性之间达到完美的平衡.另外,网络在传输过程中非常容易受到各种因素的干扰,特别是不法分子对网络流量恶意操纵后,原有的特征可能会发生改变.

为了解决这些问题,研究人员通过网络隐蔽通道来传输水印信息.隐蔽通道利用通信双方共用或已知的合法交互方式,经过再次约定形成隐蔽性强的额外通信渠道,进而实现在合法通道上传输额外信息的目的[7].对于普通的使用者而言,主动流水印技术是透明的,使用者很难察觉到水印的存在.主动流水印技术对时间和空间的需求远远小于被动网络流分析技术,因此基于主动流水印技术的流量追踪可以更好地实现有限资源条件下的精准定位.

目前,水印技术的研究主要集中在水印的嵌入方法、载体选择及识别等方面,并取得了一些研究成果.然而由于网络环境的瞬变性与网络业务的多样性,流水印技术依然面临着诸多挑战,例如,网络攻击者可以通过统计方法或人工智能方法来推断目标流中存在的非公开的水印,从而破坏或篡改水印信息.因此,水印本身的抗干扰性和可恢复性至关重要.目前的流水印方法中,水印信息通常是随机生成的二值序列,内容和位置的相对固定会造成追踪的流量呈现出自相似性.另外,目前大多数网络侧写工作都集中在 IPv4 协议上,与 IPv4 协议相比,IPv6 协议内嵌了标准化的 IPsec 协议,报文内容的不可见性使得基于传输层和应用层的水印嵌入方案难以实施.因此,针对 IPv6 网络空间的治理需求以及有限的时空资源条件限制,特征水印生成方案的研究具有一定的意义.

1　相关工作

以网络协议为基础构建的隐蔽通道称为网络隐蔽通道[8].据通信过程中隐蔽信息嵌入的方式可以将隐蔽通道分为两大类:隐蔽存储通道,即通过修改协议中的内容来实现隐蔽信息的传输;隐蔽定时通道,即通过改变数据包的发送时延、发送顺序来实现隐蔽信息的传输[9].虽然在 IPv6 协议标准制定之初,就已经充分地考虑到了数据传输的安全性,但由于部分字段的定义不够明确或严格,某些字段中仍存在保留位,以及对字段未定义的值采取忽视做法等问题,使得 IPv6 协议中的隐蔽通道构建成为了可能.

杨智丹等人分析了 IPv6 协议标准中存有漏洞的部分,提出了 5 类共 19 种隐蔽通道的构建方式,包括定义不完整字段、保留字段、转发时被中间节点忽视的字段及非关键字段等[10].在文献[11]中,以

IPv6 报文固定头部中的跳限制字段为载体,实现了基于数据包操作和位变换 2 种水印嵌入方式,并详细阐述了基于位变换的隐蔽通道构建方式.Lucena 等人分析了 IPv6 协议报文标准,基于 IPv6 协议的固定头部和 6 种扩展头部字段中的保留位,提出了 22 种隐蔽通道的构建方式[12].此外,IPv6 报文扩展头部中的目标选项头部也被用于构建隐蔽通道,Mavani 提出了针对该隐蔽通道的识别检测方法[13].在文献[14]中,研究人员没有提出新的隐蔽信息传输方法,而是关注于阻断隐蔽通道或限制其传输能力的方式.Wojciech 等人根据实际网络中的流量,通过对 IPv6 报文中可以用来构建隐蔽通道的字段取值分布的统计,以及安全设备对隐蔽信道的检测情况,分析在实际的应用场景中,寻找更加实用与有效的通道构建方法[15].

主动网络流水印技术是一种基于隐蔽定时通道的网络流量追踪技术,相较于隐蔽储存通道能够更好地适应 IPv6 的加密环境.根据水印嵌入的载体可将主动网络流水印方法分为基于流速、分组和间隔[16]3 种.例如,在 2003 年一种基于分组延迟的水印方法(WBIPD)被提出,该方案利用了时间上的扰动来传递水印信息[17].但是,基于分组延迟的水印方案鲁棒性不高,非常容易因为网络干扰和恶意攻击失去效用.

Houmansadr 等人提出了 RAINBOW 方案,该方案需要一个通信双方共享的 IPD 数据库,数据库中存放着目标流的原始 IPD 信息[18].为了减少通信过程中丢包、包重组和垃圾包注入对水印信息的影响,RAINBOW 方法使用了比其他基于分组延迟水印技术更低的延迟量来增加水印的不可见性,减少被攻击者检测与破坏的可能性.但是,为了进行包含 IPD 逐一匹配的流关联判断等操作,该方法需要更多的时间资源.为了解决水印嵌入过于复杂的问题,部分研究人员不再使用简单的二进制序列作为水印信息,而是使用 PN 码对原始的水印信息进行扩频,再通过调整分组延迟进行水印信息的嵌入,检测端可以通过解扩操作提取水印信息[19].

2018 年 Lacovazzi 等人提出了一种新的网络流水印方法 Drop Wat,它通过模拟实际网络中的丢包行为,即删除流的几个选定的数据包,改变数据包之间的延迟来实现水印信息的嵌入[20].虽然该方法的隐蔽性较高,有效地减少了通信过程中的水印泄露几率,但是严重依赖于数据包的正常传输,如果传输过程中自然丢包率过高、网络抖动较大,将会出现比较严重的检测误差.

在网络干扰有限的情况下,基于分组延迟的水印技术能够对流量进行有效地追踪,具有比较高的鲁棒性.在此基础上,一种基于间隔的水印方法(interval based watermarking, IBW)被提出,该方法不再使用单个包间隔(inter-packet delag, IPD)作为载体,而是将目标流划分成固定长度的时间间隔,通过更改时间间隔内的数据包延迟,调整时间间隔内的数据包数量来嵌入水印信息[21].

2011 年 Houmansadr 等人为了解决多流追踪问题,提出了 SWIRL 方法,该方法将时间间隔重心分为基本间隔重心和标记间隔重心,水印的嵌入方式由基本间隔重心决定[22].2015 年 Lin 等人为了增强水印的抗干扰性以及抵御多流攻击的能力,提出了基于时间间隔重心匹配的水印模型,通过密钥动态选择水印的嵌入位置,改变时间间隔的重心来传递水印信息[23].2016 年一种更为均衡的基于时间间隔重心的水印方法被提出,该方法试图用更小的质心调制范围来达到较高的识别精度,从而提高跟踪简短流的有效性[24].2018 年 Zhai 等人使用近似正交的序列集来计算最小的质心移动距离,有效地减少了基于时间间隔质心的水印方案需要添加的人为延迟[25].

Yu 等人将无线通信中的 DSSS 机制融入到主动网络流水印技术中,提出了基于流速的新型水印方法[26].2015 年 Liu 等人针对不同虚拟机的出口网络流,在原型框架 OBSERVER 的基础上,提出了一种差异分析方法[27].该方法能够对 SDN 的状态进行动态配置,通过离散小波的多分辨率变换,对固定时间段的网络流进行分解,并使用 K-L 距离来衡量原始流和水印流之间的差异性.

2 基于流特征的混合载体水印生成方案

水印的生成包括 3 个步骤:1)初始化原始水印信息的二进制向量.2)利用预设的编码方式进行编码,并将水印嵌入到目标流中.3)在水印检测处获取目标网络流,并根据网络流水印的解码规则恢复原始水印信号.本文针对水印隐蔽性的优化目标,提出特征水印序列的生成方法,实现基于混合隐蔽通道和混合时隙的水印生成方案,降低水印对原始流量的影响.

2.1 特征水印序列生成

内容和位置相对固定的水印会造成追踪的流量出现自相似性,造成水印信息的隐蔽性大大下降.因此,

本文结合目标流的本身特性,提出具有意义的特征水印序列生成方案,降低水印信息的自相似性,减少水印被识别攻击的概率.

2.1.1　基于荷载的水印序列生成

面向间断性传输的特征提取场景,包依赖的特征具有更强的适应性.IPv4 协议中的 IP 分片是一个非常重要的特性,当数据包大小超过链路最大传输单元(maximum transmission unit, MTU)时,中间节点路由器可以将该数据包重新分片为满足当前 MTU 的多个数据包,因此 IPv4 流量的数据包荷载分布具有随机性.与 IPv4 协议不同的是 IPv6 协议只允许数据包在源节点分片,在目的节点重组,中间节点路由器只承担转发的任务,不允许对 IPv6 数据包进行分片重组.如果收到的数据包荷载超过了当前节点的最大传输单元,中间节点会丢弃该数据包,并给源节点发送 ICMPv6 消息,消息中携带了可以通过该中间节点的最大数据包荷载.因此,在 IPv6 环境下,数据包荷载和数量都是较为稳定的,尤其是在 TCP 连接中,数据包荷载值具有高度集中性,荷载序列呈现出了阶段的稳定性,所以可以通过荷载序列实现流特征的表示.图 1 展示了 2 次不同 TCP 通信中的荷载序列.

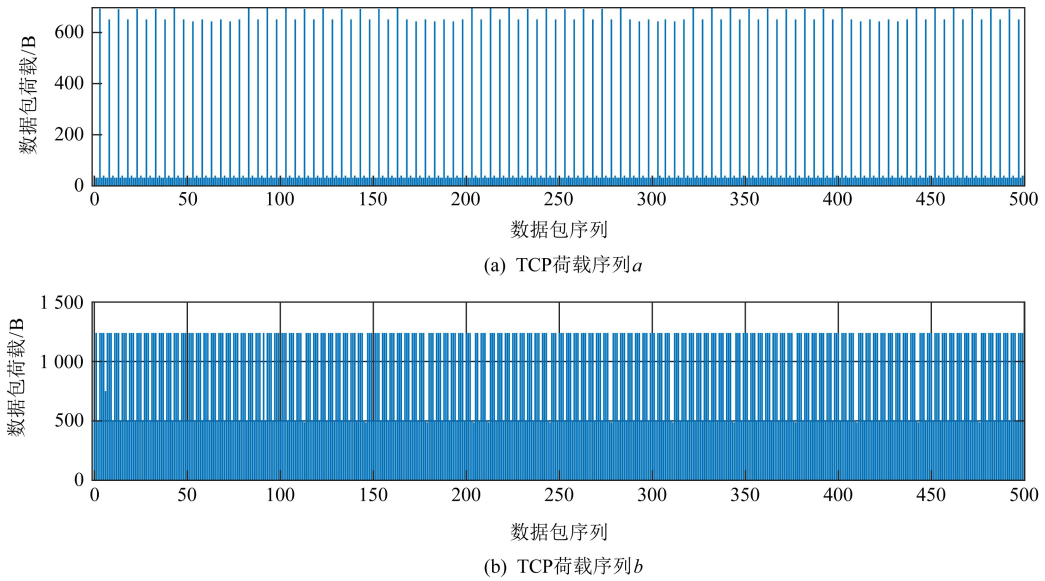


Fig. 1　Load sequence of TCP

图 1　TCP 荷载序列

数据包荷载无法直接作为水印信息进行传输,需要对荷载序列进行二值处理.如果直接使用定长的二进制编码,为了满足最大荷载的表示需求,单个数据包荷载的二进制码长度最少设置为 11 b,这样会造成有限长度的特征水印序列包含的有效信息过少.如果简单地使用中位数或均值进行二分类编码,虽然节约了编码长度,但是严重压缩了荷载信息,模糊了流本身的特性以及流与流之间的相异性.为了更好地通过荷载水印序列体现流特征,需要对荷载信息进行合理地分类,在没有标签的情况下,无监督聚类方法具有更好的适用性.

基于水印的追踪方案优势在于较低的时空开销,所以对水印信息长度有着较为严格的要求.数据包荷载序列能够反映出通信过程中的部分特征,为了区分不同的网络流,挖掘出最能够反映出当前通信特征的有限长度的荷载序列,需要尽量保证规定荷载序列中包含较多的数据包.因此,结合哈夫曼编码(Huffman Coding)的思想,针对出现概率较高的荷载类别,给予较短的二值编码,针对出现概率较低的荷载类别,给予较长的二值编码.

由于隐蔽通道带宽的限制,在经过哈夫曼编码后的数据包荷载序列仍然无法全部发送给接收方.所以,需要从完整的荷载序列中提取最能够代表通信特征的有限长度序列,即局部最优的子序列.通常情况下,使用出现频率最高的子序列来代表原序列,但是由于原序列中可能存在多个出现次数相近的频繁子序列,当网络环境出现波动时,子序列的提取会受到影响.并且基于哈夫曼树的二值编码使得不同的数据包荷载对应的码长不同,则子序列中的容量即序列中实际包含的数据包数量不同.将子序列的特征表现能力记为特征强度 CP_i,

$$CP_i=\frac{ap_i}{\sum_{k=1}^{n}ap_k}vl, \tag{1}$$

其中,ap_i 表示第 i 个序列出现的次数,vl 表示序列实际包含的数据包个数,所求特征强度最强的子序列即为局部最优子序列.

就 TCP 协议而言,可以将数据包分为 2 种类型:1)用来传输数据,实际荷载与上层的应用协议以及传输路径有关,呈现出流的相异性;2)应答包,包括 SYN 包、ACK 包等,根据应答包的不同功能,实际荷载也会有所不同,大多为 20,24,32 等.针对不同的通信状态,提取出的荷载特征序列也会有所不同.

图 2 是设备 V_1 与设备 V_2 通信过程中较为常见的 3 种荷载序列状态图.a 型荷载序列表现出的通信特征是,设备 V_1 持续稳定地给设备 V_2 传输数据;b 型荷载序列表现出的通信特征是,设备 V_1 持续地接收到设备 V_2 发送的信息,并给予回应;c 型荷载序列表现出的通信特征是,设备 V_1 和设备 V_2 互相之间都在进行数据传输.传输数据的大小、传输数据的速率以及网络干扰,例如超时导致的频繁重传等因素都会影响到荷载序列的提取,使得不同的通信状态之间具有相异性,从而有利于更好地进行流区分.另外,TCP 的传输具有稳定性,在一定时间段内的荷载序列有着较强的自相似性,在网络状态没有发生突变的情况下,能够在一定范围内抵抗丢包、垃圾包注入等因素的影响,有比较好的鲁棒性.

在间断性传输的特征提取场景中,包依赖的特征具有更强的适应性,同时利用 IPv6 协议以及 TCP 传输的稳定性,基于载荷的水印序列生成方式能够在很大程度上表现出较好的环境适应能力.

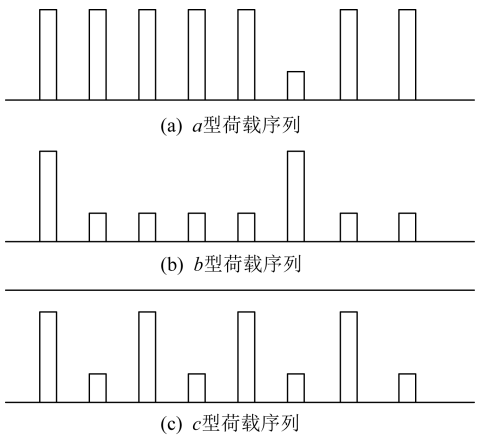


Fig. 2 State of load sequence

图 2 荷载序列状态

2.1.2 基于时隙质心的特征水印序列生成

面对流速较慢的特征提取场景,发送端需要缓存更长时间的特征流,才能够捕获到一定数量的数据包,而基于时隙质心的特征对数据包密度的依赖性远远低于荷载特征.在 IPv6 环境中,数据包的数量相对于流同样是独立同分布的,所以其时隙质心仍然符合泊松分布,图 3 展示了一段时间内的时隙质心分布情况.

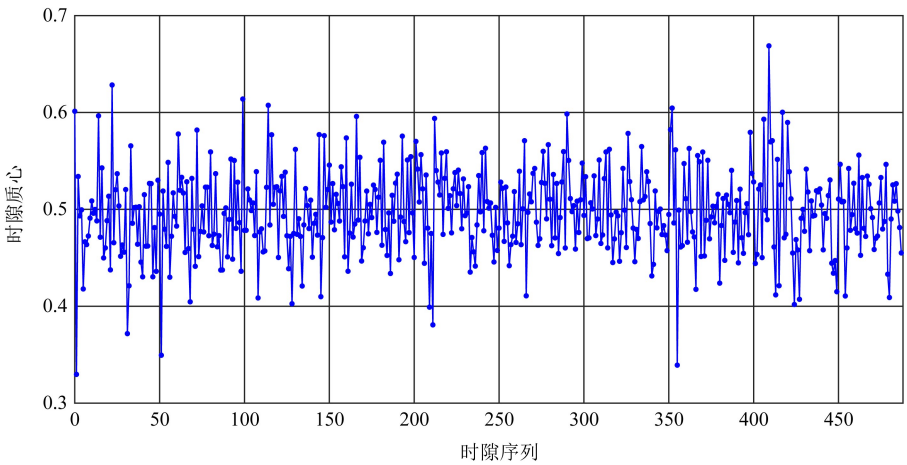


Fig. 3 Distribution of time-slot centroid

图 3 时隙质心分布

时隙质心符合泊松分布,即数据包到达时间偏移均匀分布在单个时隙内,所以可以估算出组内数据包到达间隔的时间偏移重心在时隙中心附近,进行水印信息的初步提取.但是在实际的网络流中,受到丢包重传、网络延时等不良因素的影响,时隙质心的分布不够稳定,容易因为网络的波动发生偏移.为了降低这些不良因素对质心分布的影响,本文提出了序列映射区的概念.

如图 4 所示,序列映射区以时隙中心为轴,并在单位映射区内向右偏移,偏移时长根据网络的实际

情况进行相应调整.如果初始时隙质心的位置不在序列映射区内,则需要通过人为添加延迟,使时隙质心落在映射区之内.

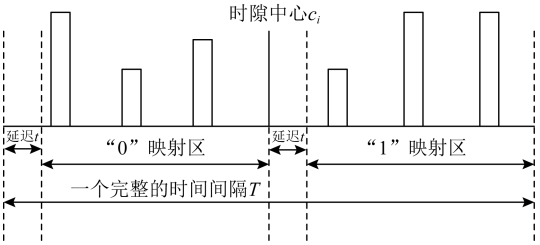


Fig. 4　Mapping zone of sequence

图 4　序列映射区

如图 5 所示,当前时隙的质心 C_i 可以表示为

$$C_i=\frac{1}{n}\sum_{i=1}^{n}\Delta t_i=d_1+\frac{n-1}{n}d_2+\frac{n-2}{n}d_3+\cdots+\frac{2}{n}d_{n-1}+\frac{1}{n}d_n,\quad(2)$$

其中,d_j 表示第 $j-1$ 个数据包和第 j 个数据包的到达时间差值,t 是人为添加的延迟,Δt 表示偏移时长.在单时隙中,人为添加延迟的数据包越靠近时隙的起点,时隙质心的增量越大.为了降低时隙质心移动对原始流量的影响,尽量减少时隙内数据包达到时间差的变化数量,最理想的状态是为当前时隙中的所有数据包人为地添加延迟 s,这样除了 d_1 以外,其他的 d_i 都能够保持不变.假设质心的移动距离为

$$C_i'=\frac{1}{X_i}\sum_{j=1}^{x_i}(p_{i,j}+s)-\frac{1}{X_i}\sum_{j=1}^{x_i}p_{i,j}=s,\quad(3)$$

由式(3)可知,s 最小值即为质心移动的最小距离.

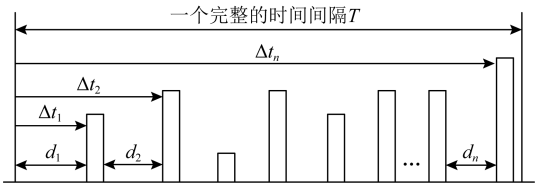


Fig. 5　Illustration of a complete time slot

图 5　完整单时隙

相较于包依赖水印序列生成,基于时隙质心提取的特征序列大小同样对数据包密度具有更小的依赖性,实际表现取决于数据包到达延迟.

在检测端提取时隙质心特征来获取原始时隙质心序列的时候,序列映射区的位置与发送端有所变化.在检测端进行时隙分割的时候,加入了时间偏移量,例如当发送端时间间隔 I_i 的起始时刻为 $t_{i,0}$,则检测端相匹配的时间间隔 I_i' 的起始时刻为 $t_{i,0+m}$.

因此,在网络流速较慢的特征提取场景中,相比于载荷特征,基于时隙质心的特征提取对数据包密度具有更低的依赖性.本文通过序列映射,减轻了网络波动对时隙质心分布的干扰,大大降低了水印对原始流量的影响,有效减少了特征水印被识别攻击的概率.

2.2　基于混合隐蔽通道的水印嵌入

面向间断性传输的网络传输场景,包依赖的水印载体能够更有效地保证水印连贯性.由于 IPv6 协议内嵌了 IPsec 协议,报文内容的不可见性限制了载体的选择范围,基于单一载体的水印嵌入方案的隐蔽性受到了威胁.此外,抵抗重组分片干扰能力较弱的 IPD 水印,在 IPv6 环境下有更强的可用性.

2.2.1　隐蔽通道库构建

根据使用场景的差异性及实时网络状态的随机性,可以更换不同的隐蔽通道来嵌入水印信息,通过不同的排列组合,增强水印的隐蔽性,本文选择了 IPv6 协议中的 4 条包依赖的隐蔽通道构建混合通道.另外,使用单个数据包作为水印载体对网络环境的稳定性有着极高的要求,为了提高水印信息的传输可靠性,本文将 m 个连续的数据包看作一个数据组 M_i.

1) 跳限制字段.由于网络的状况以及路由信息的变化是高频率触发事件,该字段的值在实际的传输过程中会发生改变,所以水印嵌入端可以根据预先约定的内容自行设定或更改跳限制字段值.但是,跳限制字段值在经过若干个路由器之后,其变化不受控制,可能会影响到水印信息的提取.如果使用固定的解码规则,就意味着将发送方和接收方固定在一个本地链路内或者固定了双方通信链路中的路由器的跳数,这显然是不符合实际网络环境的.为了能够识别跳限制字段中的水印信息,本文选择了基于组差的水印嵌入方式.将每个数据组按数据包个数平均分为前后 2 部分,分别记为 $M_{(i,1)}$ 和 $M_{(i,2)}$,将水印信息编码为 $M_{(i,1)}$ 和 $M_{(i,2)}$ 的跳限制字段平均值差值.当要嵌入的水印位为 0 时,增加 $M_{(i,1)}$ 的跳限制字段值,减少 $M_{(i,2)}$ 的跳限制字段值,当要嵌入的水印位为 1 时,减少 $M_{(i,1)}$ 的跳限制字段值,增加 $M_{(i,2)}$ 的跳限制字段值.

2) 通信流字段.根据对实际环境中网络流量的分析,使用该字段的前 3 位来传递水印信息,但是由于该字段可以被中间的节点改变,所以如果使用单个数据包隐藏水印信息可能会造成信息的丢失,

所以在使用该字段传递水印信息的时候,随机对整个数据组中一半的数据包进行重复性水印嵌入来提升水印信息传输的可靠性.

3) 流标签字段.由于流标签为 0 的数据包占据总数的大部分,为了减少对整体统计特性的影响,在每个数据组中只选择 2 个数据包进行水印信息的嵌入,每次嵌入 4 b 的水印信息.同时为了兼顾流标签的伪随机性,在不同的数据包中插入水印信息的位置不同.本文采用轮询插入法,假设当前数据组是第 i 个使用流标签字段隐蔽通道进行传输的水印信息,则其水印的嵌入起始位置为 $i\%16$.

4) 时间间隔 IPD.虽然 IPD 水印抵抗重组分片干扰的能力非常弱,但由于 IPv6 协议不允许中间节点对数据包分片重组,所以基于 IPD 的隐蔽通道在 IPv6 环境下可用性大大增加.将每个数据组按数据包个数平均分为前后 2 部分,分别记为 $M_{(i,1)}$ 和 $M_{(i,2)}$,将水印信息编码为 $M_{(i,1)}$ 和 $M_{(i,2)}$ 的 IPD 差值.当要嵌入的水印为 0 时,给数据组 $M_{(i,1)}$ 中的数据包添加额外的延迟,当要嵌入的水印为 1 时,给数据组 $M_{(i,2)}$ 中的数据包添加额外的延迟.

如图 6 所示,水印使用复合窗口的方式进行嵌入,每个复合窗口包含 2 个连续的数据组,由于短时间内的流量具有一定的稳定性,所以设定第 1 个数据组为观察组,用来决定使用的隐蔽通道;第 2 个数据组为嵌入组,用来嵌入水印信息.

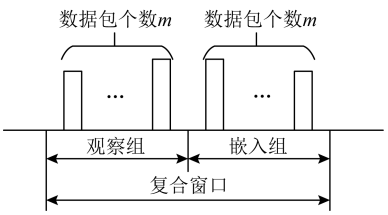


Fig. 6 Watermarking embedded in the composite window

图 6 水印嵌入复合窗口

2.2.2 隐蔽通道选择

为了能够对混合通道中的各类隐蔽通道的隐蔽性进行量化计算,本文通过流量的信息熵值、均值偏差、全局统计特性进行多维度的衡量.在观察组中,假设序号为 p 的隐蔽通道的原始特征值序列为 $(z_1,z_2,\cdots,z_m)$,记为 $S_{(k,p)}$,其中 k 表示第 k 个复合窗口,观察组对应的数据组序列号为 $2k-1$.在加入水印信息后,隐蔽通道的特征值序列发生了改变,变成 $(z'_1,z'_2,\cdots,z'_m)$,记为 $S'_{(k,p)}$.

1) 熵值差异

信息熵是香农从热力学借用的概念,可以体现出数据在某个特征上的集中程度,对于网络流量而言,当数据的特征值越分散,它的特征熵就越大,数据的特征越集中,特征熵就越小.

$$H(S_{(k,p)})=-\sum_{n=1}^{m}p(z_n)\log p(z_n), \tag{4}$$

$$Hdr_{k,p}=\frac{|H(S_{(k,p)})-H'(S_{(k,p)})|}{H(S_{(k,p)})}, \tag{5}$$

其中,$H(S_{(k,p)})$代表第 k 个复合窗口的观察组中序号为 p 的隐蔽通道的特征值信息熵,$H'(S_{(k,p)})$代表加入水印信息后该隐蔽通道的特征值信息熵,$Hdr_{k,p}$代表第 k 个复合窗口的观察组中序号为 p 的隐蔽通道的信息熵差异值.差异值越大,则表示该隐蔽通道对原始流量的影响越大.

2) 均值偏差

通过比较原始流量和加入水印信息后的调制流量的均值偏差,选择隐蔽通道.

$$Adr_{k,p}=\frac{1}{Ave_{k,p}}\sum_{n=1}^{m}|z_n-Ave_{k,p}|, \tag{6}$$

其中,$Ave_{k,p}$代表第 k 个复合窗口的观察组中序号为 p 的隐蔽通道的特征平均值,$Adr_{k,p}$代表第 k 个复合窗口的观察组中序号为 p 的隐蔽通道的均值偏差.均值偏差越大,则表示该隐蔽通道对原始流量的影响越大.

通过对信息熵差异值和均值偏差 2 个维度的计算,量化当前复合窗口的各个隐蔽性.

$$C_{k,p}=z\times Hdr_{k,p}+(1-z)\times Adr_{k,p}, \tag{7}$$

其中,z 为自定义的偏重参数,取值范围为[0,1],用来衡量隐蔽性计算中对不同维度的偏重程度,$C_{k,p}$代表第 k 个复合窗口的观察组中序号为 p 的隐蔽通道的隐蔽性值,$C_{k,p}$的值越小,则表明该类型的隐蔽通道的隐蔽性越好.

隐蔽性的量化面向的是局部的复合窗口,在嵌入水印信息的时候,全局的统计特性也要纳入考虑范围之内.因为单个复合窗口携带的水印信息越多,对原始流量的影响越大,而在大部分情况下,并不需要最大限度地使用隐蔽通道的传输带宽.所以本文在单个复合窗口中只使用一种隐蔽通道,进一步降低水印信息的嵌入对原始流量的影响.

设 WL 表示水印信息的长度,同时也代表了单冗余度下需要选取的复合窗口的数量.因为通信流字段和流标签字段在实际网络流量中有明显的全局统计特征,需要优先考虑.剩余的部分则采用跳限制

字段及 IPD 的隐蔽通道进行传输.因此,第 k 个复合窗口中序号为 p 的隐蔽通道 $Ch_{k,p}$ 可以用 3 个维度的特征值来描述:

$$Ch_{k,p}=\{wl_p,c_{k,p},n_p\}, \tag{8}$$

其中,wl_p 表示使用序号为 p 的隐蔽通道可以携带的水印信息长度,n_p 表示序号为 p 的隐蔽通道的剩余可用次数.在顺序嵌入水印信息的过程中,载体选择当前复合窗口隐蔽性最好的通道,该过程如算法 1 所描述.

算法 1. 隐蔽通道选择算法.

输入:数据包序列 P、通道使用次数 n_p、水印信息长度 L;

输出:水印通道序列 S.

① WHILE $L>0$ DO
② FOR $j=1:4$ DO
③ calculate 窗口隐蔽性值集合 C_j;
④ END FOR
⑤ choose minist C_j;
⑥ IF $n_p>0$ THEN
⑦ add 通道 m to S;
⑧ update n_p and L;
⑨ ELSE
⑩ remove C_m out of C_j;
⑪ goto line 5;
⑫ END IF
⑬ END WHILE

2.3　基于混合时隙的水印嵌入

面向流速较慢的网络传输场景,时间依赖的水印载体能够更好地节约水印嵌入时长,降低对原始流量的影响,并且时间依赖的水印载体有更强的抗丢包干扰能力.由于混合水印载体拥有更好的隐蔽性和传输带宽,本文设计了一种基于混合时隙的水印(mixed interval based watermarking, MIBW)生成方法,结合时间间隔重心和时间间隔 2 种载体实现地基水印和内部水印的嵌入.

2.3.1　基于时间间隔重心的地基水印嵌入方法

给目标流 f 设定一个随机时间偏移 o,经过时间 o 后对持续时长为 T_{in} 的目标流嵌入地基水印,设地基水印开始点的时间戳为 t_0.

对于长度为 WL_p 的地基水印信息 W_p,将 T_{in} 分成 $2n$ 个长度为 T 的间隔 I_i,每个 I_i 含有 X_i 个连续的数据包.数据包 $P_{i,j}(1\leqslant i\leqslant 2n,1\leqslant j\leqslant X_i)$ 的发送时间戳为 $t_{i,j}$,它相对于其所在间隔 I_i 开始点的时间偏移为 $\Delta t_{i,j}$.

在 $2n$ 个间隔 I_i 中随机选择 n 个间隔组成 A 组间隔 $I(A)_k(1\leqslant k\leqslant n)$,剩下 n 个间隔组成 B 组间隔 $I(B)_k$.分别将组 A 和组 B 的间隔随机分配,使得每 $2r_p$ 个间隔用来编码一位水印位,其中 r_p 表示地基水印信息的冗余度.

本文中 $2n$ 个间隔的分配策略为:设 x 为间隔号$(1\leqslant x\leqslant 2n)$,将第 $i,i+WL_p,i+2WL_p,\cdots,i+(2r_p-1)WL_p$ 号间隔用来编码第 i 位地基水印.其中,当 $\dfrac{x-i}{WL_p}\%2=i\%2$ 时,将第 x 号间隔分配为 A 组间隔;否则,将第 x 号间隔分配为 B 组间隔.

$I(A)_{i,j}$ 和 $I(B)_{,i,j}$ 分别为组 A 和组 B 中用作编码第 i 位地基水印的第 j 个间隔.$X(A)_{i,j}$ 和 $X(B)_{i,j}$ 分别为间隔 $I(A)_{i,j}$ 和 $I(B)_{i,j}$ 中的数据包数量,$X(A)_i$ 和 $X(B)_i$ 表示编码第 i 位地基水印的数据包的总数量,计算方法为

$$X(A)_i=\sum_{j=0}^{r_p-1}X(A)_{i,j}, \tag{9}$$

$$X(B)_i=\sum_{j=0}^{r_p-1}X(B)_{i,j}. \tag{10}$$

间隔 $I(A)_{i,j}$ 和 $I(B)_{,i,j}$ 中第 k 个数据包 $P_{i,j,k}$ 的时间偏移为 $\Delta t(A)_{i,j,k}$ 和 $\Delta t(B)_{i,j,k}$,分别聚合组 A 和组 B 中 r_p 个间隔的时间戳,计算组 A 和组 B 的数据包的整体时间间隔偏移重心为

$$A_i=\frac{1}{X(A)_i}\sum_{j=0}^{r_p=1}\sum_{k=0}^{X(A)_{i,j}-1}\Delta t(A)_{i,j,k}, \tag{11}$$

$$B_i=\frac{1}{X(B)_i}\sum_{j=0}^{r_p=1}\sum_{k=0}^{X(B)_{i,j}-1}\Delta t(B)_{i,j,k}. \tag{12}$$

将每个地基水印编码为 A_i 和 B_i 的时间偏移重心差 Y_i.当要编码的地基水印信息是 1 时,通过增加 A_i 使 Y_i 的分布向右平移,即在 r_p 个间隔 $I(A)_{i,j}$ 中的数据包 $P_{i,j,k}$ 发送前人为添加额外延迟,且间隔 $I(A)_{i,j}$ 即为内部水印的嵌入位置;当要编码的地基水印信息是 0 时,通过增加 B_i 使 Y_i 的分布向左平移,即在 r_p 个间隔 $I(B)_{i,j}$ 中的数据包 $P_{i,j,k}$ 发送前人为添加额外延迟.且间隔 $I(B)_{i,j}$ 即为内部水印的嵌入位置.

2.3.2　基于时间间隔的内部水印嵌入方法

地基水印信息嵌入时,需要在间隔中的数据包发送前人为添加额外延迟来改变重心的分布.在重心移动的同时,数据包数量的分布也发生了改变,为了更好地利用时间隐蔽通道,提升水印信息的传输带宽,本文利用了重心移动的间隔实现基于时间间隔的内部水印嵌入.

内部水印的嵌入位置和嵌入长度都由地基水印确定.每一位地基水印信息确定一位内部水印信息的嵌入位置.内部水印的冗余度 r_{in} 由内部水印信息长度 WL_{in} 和地基水印决定.

在传统的基于时隙的水印方法(例如 IBW)中,往往需要借助下一个时隙来改变数据包数量差,但本文的地基水印使用的是连续的时间间隔,如果将数据包推入下一个时间间隔,可能会影响到地基水印的嵌入.为了保证内部水印信息的嵌入能够在一个单独的时间间隔中完成,将长度为 T 的间隔 I_i 平均分为 3 个小间隔,记为 $Sub_i(1\leqslant i\leqslant 3)$,在 $\langle Sub_{i,j,1},Sub_{i,j,2},Sub_{i,j,3}\rangle(1\leqslant i\leqslant WL_{in},1\leqslant j\leqslant r_{in})$ 中,$\langle Sub_{i,j,1},Sub_{i,j,2}\rangle$ 用作编码第 i 位内部水印的第 j 组嵌入的小间隔对,$Sub_{i,j,3}$ 为辅助小间隔,用于对内部水印进行辅助判断.每个小间隔中含有 $X(Sub)_{i,j,k}$ 个连续数据包.

在不添加人为干扰的情况下,数据包的到达时间在每个小间隔中是均匀分布的,因此每个小间隔中所含有的包数量的期望 u 是相同的.将每个内部水印编码为 $Sub_{i,j,1}$ 和 $Sub_{i,j,2}$ 的数据包数量差 $Y(Sub)_{i,j}$ 为

$$Y(Sub)_{i,j}=X(Sub)_{i,j,2}-X(Sub)_{i,j,1},\quad(13)$$

即可计算出用来编码第 i 位水印信息的 r_{in} 个 $Y(Sub)_{i,j}$ 的平均值为

$$\overline{Y(Sub)}_{i,r_{in}}=\frac{1}{r_{in}}\sum_{j=0}^{r_{in}-1}Y(Sub)_{i,j}.\quad(14)$$

已知每个小间隔中数据包的数量期望为 u,并且数据包数量 $X(Sub)_{i,j,k}$ 相对于网络流是独立同分布的,可算出 $\overline{Y(Sub)}_{i,r_{in}}$ 的期望值为 0.因此可以通过增加或减少 $\overline{Y(Sub)}_{i,r_{in}}$ 来编码内部水印信息 1 或 0,即通过调整 $Sub_{i,j,1}$ 和 $Sub_{i,j,2}$ 内的数据包数量 $X(Sub)_{i,j,1}$ 和 $X(Sub)_{i,j,2}$,使得 $\overline{Y(Sub)}_{i,r_{in}}$ 的分布向右平移或者向左平移.

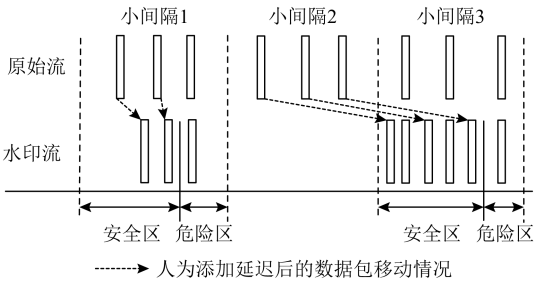


Fig. 7 Embedding with internal watermarking information 0

图 7 嵌入内部水印信息为 0

由于中间节点不允许对 IPv6 报文进行分片重组,因此在 IPv6 通信环境中,数据包数量是较为稳定的特征,内部水印更多地遭受到了丢包以及网络延迟的威胁.为了增强内部水印的鲁棒性,将小间隔划分为安全区和危险区.如图 7 所示,当要嵌入的内部水印信息为 0 时,给 $Sub_{i,j,2}$ 中所有的数据包添加延迟,使其落入 $Sub_{i,j,3}$ 中的安全区,$Sub_{i,j,1}$ 为时间间隔重心辅助调整区;如图 8 所示,当要嵌入的内部水印信息为 1 时,给 $Sub_{i,j,1}$ 中所有的数据包添加延迟,使其落入 $Sub_{i,j,2}$ 中的安全区,$Sub_{i,j,3}$ 为时间间隔重心辅助调整区.

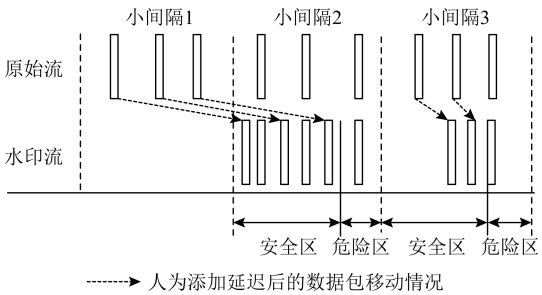


Fig. 8 Embedding with internal watermarking information 1

图 8 嵌入内部水印信息为 1

2.3.3 水印检测

针对地基水印信息,通过记录每一间隔内的所有数据包的到达时间偏移,由式(11)和式(12)计算组 A_i 和组 B_i 的数据包到达时间偏移重心,得到时间偏移重心差 Y_i.当 $Y_i>0$ 时,判定地基水印信息为 1;否则,判定地基水印信息为 0.

针对内部水印信息,记录每组小间隔对 $\langle Sub_{i,j,1},Sub_{i,j,2},Sub_{i,j,3}\rangle$ 的数据包数量,计算 $Sub_{i,j,1}$ 和 $Sub_{i,j,2}$ 的数据包数量差均值 $\overline{Y(Sub)}_{i,r_{in}}$ 及小间隔 $Sub_{i,j,3}$ 数据包数量和 $X(Sub)_{i,3}$.当 $\overline{Y(Sub)}_{i,r_{in}}>0.5$ 时,判定内部水印信息为 1;当 $\overline{Y(Sub)}_{i,r_{in}}<-0.5$ 时,判定内部水印信息为 0.若 $\overline{Y(Sub)}_{i,r_{in}}$ 在 $-0.5\sim0.5$ 之间时,则需要借助辅助小间隔 $Sub_{i,j,3}$ 进行二次判断.当嵌入的内部水印信息为 0 时,需要移动部分数据包到小间隔 $Sub_{i,j,3}$ 中,因此内部水印信息为 0 对应的 $X(Sub)_{i,3}$ 平均值要比内部水印信息为 1 对应的 $X(Sub)_{i,3}$ 平均值大.所以可以根据数据包数量总和的平均值 $\overline{X(Sub)}$ 来辅助判断,如果 $X(Sub)_{i,3}<\overline{X(Sub)}$,判定内部水印信息为 1,否则为 0.在内部水印提取的时候,并不需要特别考虑水印的嵌入位置,因为在没有人为调制和干扰的情况下,数据包的

到达时间在每个小间隔中是均匀分布的,即每个小间隔中所含有的数据包数量的期望 u 是相同的.所以,针对非内部水印嵌入位置的时间间隔,小间隔对 $\langle Sub_{i,j,1}, Sub_{i,j,2}\rangle$ 的数据包数量差值均值为 0,对水印的提取没有影响.

基于混合时隙的水印信息在增强了隐蔽性的同时也导致了更高的误差概率.因此,如何控制水印的差错就显得十分重要,纠错码的核心设计思想是增加冗余信息. 低密度奇偶校验码(low-density parity-check codes, LDPC)是一种线性分组码,可以将长度为 L 的特征序列,通过编码器添加 $WL-L$ 位校验码元,变成长度为 WL 的水印信息,L 位特征序列与 $WL-L$ 位校验码元呈线性关系.则水印信息可以用(WL, L)来表示.

LDPC 码的编码方法可以简单地表示为特征序列与生成矩阵 $\boldsymbol{G}$ 相乘的结果,但是由于通过生成矩阵 $\boldsymbol{G}$ 直接编码,运算的复杂度比较高,所以本文选择对校验矩阵 $\boldsymbol{H}$ 高斯消元进行编码,如果在高斯消元的过程中进行了列交换,则需要对水印信息进行相应位交换得到水印传输信息再进行嵌入工作.

收到经过 LDPC 编码的水印传输信息后,将水印传输信息与校验矩阵 $\boldsymbol{H}$ 相乘,如果结果是 $\boldsymbol{0}$ 矩阵,则表明收到的水印传输信息是正确的.反之,则表示收到的水印传输信息有误,需要根据相乘的结果进一步纠错解码.因为水印传输信息属于短码,所以本文使用了实现相对简单的硬判决译码算法比特翻转(bit flipping, BF)译码算法.在使用 BF 译码算法时,需要设置一个阈值,当水印传输信息位不满足校验关系式的方程个数超过这个阈值时,则表示该水印传输信息位译码出现了错误,需要将该水印传输信息位翻转后再次进行译码工作,直到译码正确或者译码的迭代次数超过了预设的值.最后,根据校验矩阵 $\boldsymbol{H}$ 高斯消元过程中的列交换情况,将水印传输信息还原成水印信息,并提取出特征水印序列.

3　模拟实验

3.1　实验环境

本文在如图 9 的实验环境中进行水印信息传输测试,其中,发送端的 IP 地址为 2001:da8:1002:6004:1:45fa,接收端的 IP 地址为 2001:da8:1002:315:91de:3992:c2a2:6170,发送端与接收端都位于江苏省南京市教育网内,操作系统均为 64 位 Ubuntu20.10.本实验使用了 WIDE Project 项目的 IPv6 数据集进行测试,数据采集自日本到美国的某条骨干网络的网络流量.从数据集中选取 10 条不同的流作为测试集进行测试,每条流的数据包数量不少于 8 000 个.在发送端对测试集进行流量重放,并通过水印嵌入机进行水印信息的嵌入.

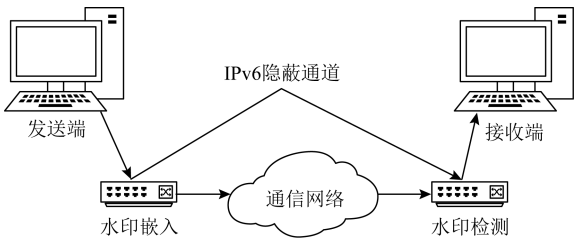


Fig. 9　The architecture of concealed channel system

图 9　隐蔽通道系统结构

3.2　鲁棒性分析

3.2.1　基于混合隐蔽通道的水印方法

在实际的通信过程中,携带水印信息的网络流可能会受到时延、丢包等因素的干扰.在干扰环境下正确传输水印信息的能力可称作隐蔽通道的鲁棒性.从理论上分析,基于字段修改的 3 种隐蔽通道对数据包本身有更强的依赖性,而基于 IPD 的隐蔽通道则会对时间更加敏感.由于发送端和接收端在没有经过中间跳板主机的情况下直接进行通信,网络环境较好,所以需要人为地添加一些干扰来进行测试.本次实验中,选择时隙特征序列作为水印信息,隐蔽性偏重参数为 0.5,4 种隐蔽通道的使用比例为 4:1:1:4,水印信息长度为 16 b,冗余度为 2,误差阈值为 3.

图 10 描述了不同延迟对混合隐蔽通道检测率的影响.随着最大抖动延迟的增加,混合通道的检测率在不断地下降,但是可以通过增加数据组的容量

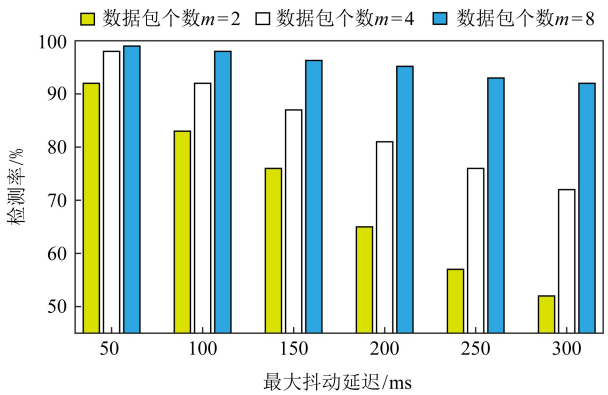


Fig. 10　Effect of delay on the detection rate of Mixed Channels

图 10　延迟对混合通道检测率的影响

来提升检测率.这是由于基于头部字段修改的隐蔽通道和基于 IPD 的隐蔽通道都非常依赖分组的到达顺序,随着数据组容量的增加,发生错位的数据包对于整个数据组而言只是非常小的一部分,不会影响到数据组的整体特征.当数据组容量达到 8 个数据包时,在 300 ms 最大抖动延迟条件下已经可以达到 90%的检测率.

图 11 描述了不同丢包率对混合隐蔽通道检测率的影响.由图 11 可知,丢包对混合隐蔽通道的检测率有着较大的干扰性,在丢包率较小的情况下,增大数据组能够提升检测率,但是随着丢包率的增加,过大的数据组反而成了拖累,小容量的数据组表现出更好的检测率.造成这种现象主要有 2 个原因:1)本文使用的 4 种隐蔽通道对于丢包的抗干扰性都很弱,基于字段修改的隐蔽通道依赖字段值提取水印信息,如果数据包大量丢失,就很难从中恢复出正确的信息.基于 IPD 的隐蔽通道依赖相邻数据包的到达时间差,数据包的丢失使得原本不相邻的数据包变成了相邻数据包,提取出的 IPD 值可能为原来多个 IPD 值的和.2)虽然通过数据组的方式增加了单个水印位的容错率,随着丢失的数据包数量的积累,复合窗口的选取误差越来越大,读取水印信息的窗口与嵌入水印信息的窗口发生了位移,导致水印信息无法提取恢复.

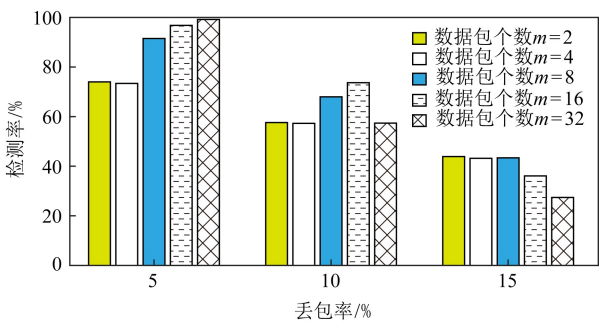


Fig. 11 Effect of packet loss on the detection rate of mixed channels

图 11 丢包对混合通道检测率的影响

3.2.2 基于混合时隙的水印方法

单个时间间隔内的数据包数量对检测率有很大的影响.设地基水印和内部水印的长度都为 48 b,其中特征水印序列 32 b,校验位 16 b,时间偏移 o=20 000 ms,地基水印冗余度为 3,地基水印和内部水印的误差阈值都为 3.本文通过调整单个时间间隔内的数据包数量,对网络流进行多次实验,测试单个时间间隔内的数据包数量对水印检测率的影响.

由图 12 可知,随着单个时间间隔内的数据包数量增加,2 种水印的检测率都在不断提高.在时间间隔内数据包数量较少时,由于内部水印需要对当前间隔再次分割,使得小间隔中的数据包数量极其不稳定,影响到内部水印的嵌入.例如当数据包数量为 3 时,小间隔中可能会存在没有数据包的情况,致使无法正确地嵌入内部水印,因此相较于地基水印检测率更低.但是随着时间间隔内数据包数量的增加,内部水印的检测率提升幅度比地基水印更快.当单个时间间隔内的数据包数量为 12 时,在设定的误差阈值下,内部水印和地基水印都能够达到 95%的检测率.

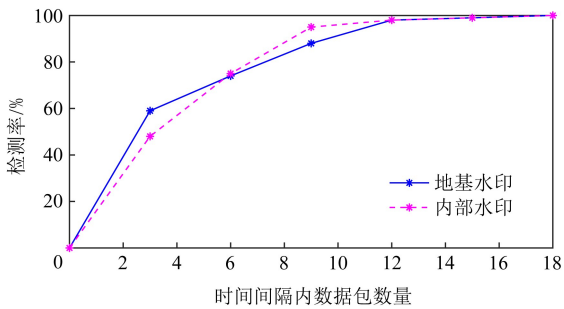


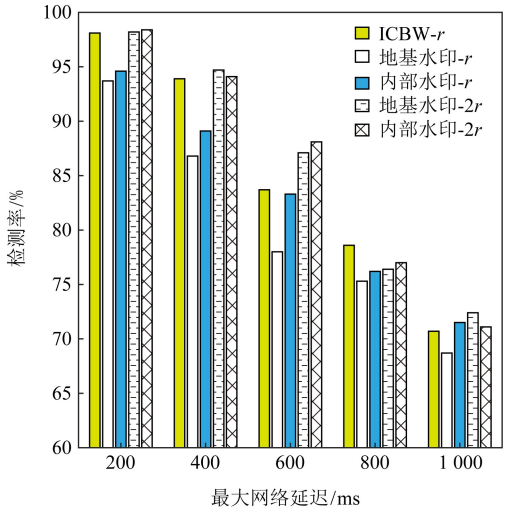
Fig. 12 Effect of packet numbers in a time interval on the detection rate

图 12 时间间隔数据包数量对检测率的影响

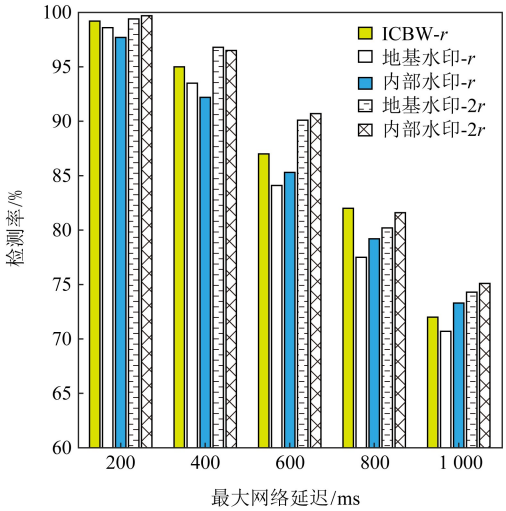
为了验证延迟抖动对检测率的影响,使用了 Linux 操作系统中的流量控制器 TC 对发送的数据添加延迟.本次实验,对比了不同的最大网络延迟下 ICBW 方案及本文提出的 MIBW 方案针对原始特征水印序列和 LDPC 码水印信息的检测率.其中,特征水印序列都为 16 b,水印信息都为 24 b. ICBW 方案的水印为地基水印和内部水印的集合.时间偏移 o=12 000 ms,冗余度 r=6,时间间隔长度 T=800 ms,时间间隔内的平均数据包数量为 12.7. ICBW 方案的误差阈值为 6,地基水印和内部水印的误差阈值为 3.

由图 13 可知,随着网络延迟的增加,2 种方案的检测率都在不断下降.通过比较图 13(a)和图 13(b)可得,使用 LDPC 码水印能够提升不同网络延迟下的水印检测率,尤其是对于低冗余度下的 MIBW 方案提升较大.其中,低延迟场景下的地基水印收益最为明显,这是由于地基水印受到了内部水印的限制,时间间隔重心的移动范围变小,导致地基水印在嵌入时就产生了错误.实验表明通过增加冗

余或使用 LDPC 码水印,可以有效降低嵌入错误的影响.当延迟过高时,水印的误码率也随之提升,受到 LDPC 码纠错能力的限制,检测率的提升十分有限.在使用 LDPC 码水印时,内部水印由于间隔长度远小于 ICBW 和地基水印,所以在低延迟下,检测率相对较低.但在延迟较高的情况下,地基水印的表现反而不如内部水印,除了因为地基水印的质心移动距离有限以外,时隙质心特征序列受到网络延迟的影响更大.所以,MIBW 方案可以在损失有限检测精度的条件下,实现更高带宽的水印信息传输;在同样的水印传输带宽下,MIBW 方案表现出了更好的检测率.



(a) 特征水印序列

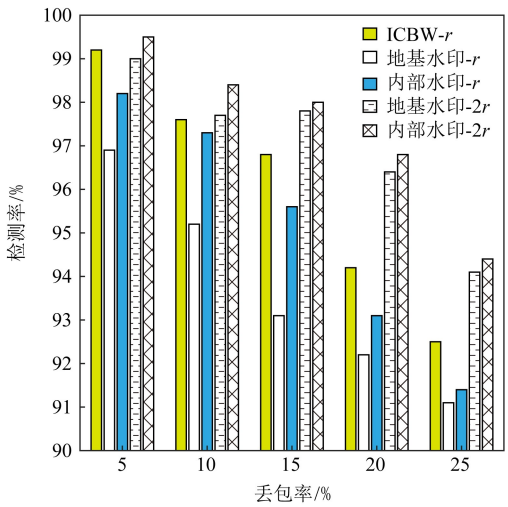


(b) LDPC码水印

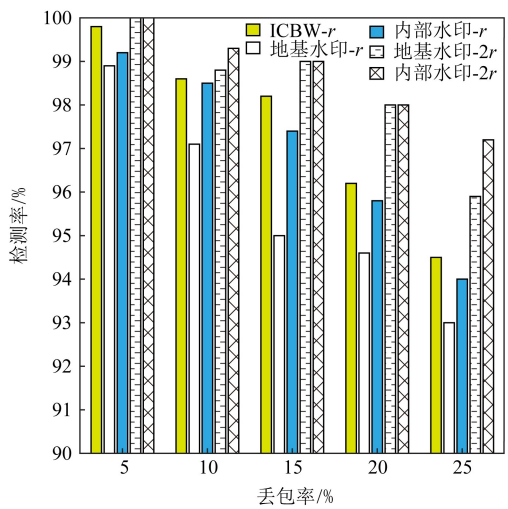
Fig. 13　Effect of delay on the detection rate

图 13　网络延迟对检测率的影响

为了测试丢包对检测率的影响,使用了 Linux 操作系统中的流量控制器 TC 对发送的数据随机丢包.由图 14 可知,随着丢包率的增加,2 种方案的检测率都在不断下降.通过比较图 14(a)和图 14(b)可得,由于整体的检测率较高,使用 LDPC 码水印对于不同丢包率下的水印检测率的提升幅度较为平均,但仍然对低冗余度下的地基水印提升较大.在相同的冗余度下,MIBW 方案的检测率略低于 ICBW 方案,但其嵌入水印的时间间隔数只有 ICBW 方案的一半,这表明 MIBW 方案能够在较高的水印传输带宽下保持稳定的检测率.在水印信息带宽相同时,MIBW 方案由于冗余度的增加,表现出了更优秀的检测率.在低丢包率场景下,内部水印相较于地基水印有更强的抗丢包干扰能力,这是由于内部水印可以通过辅助间隔进行二次判断.当丢包率较高时,部分荷载水印序列的提取出现了误差,致使内部水印的检测率有所下降.



(a) 特征水印序列



(b) LDPC码水印

Fig. 14　Effect of packet loss on the detection rate

图 14　丢包率对检测率的影响

3.3 隐蔽性分析

3.3.1 基于混合隐蔽通道的水印方法

本节使用了基于孤立森林的水印信息检测方法.孤立森林算法是一种常见的无监督异常检测算法,由于带有水印信息的流量满足 2 个异常数据所具有的特性:1)嵌入了水印信息的流量与原始流量不完全一致;2)该部分流量在整体流量中占比较小.所以本文将原始流量看作正常数据,将带有水印信息的流量看作异常数据.与传统的单一统计学检测方法相比,孤立森林的准确度更高.实验对比了使用混合隐蔽通道和使用单一通道的异常检测情况.异常识别率表示被正确划分为异常数据的样本个数占隐蔽流量样本总个数的比例,异常识别率越低,说明带有水印信息的隐蔽流量越接近原始的流量,通道的隐蔽性越高.

由图 15 可知,使用单一隐蔽通道传输水印信息时,通信流类别字段和流标签字段可以被准确识别,而对跳限制字段及 IPD 隐蔽通道的识别率则相对较低,证明后者的隐蔽性略强一些.在使用混合隐蔽通道传输水印信息后,不同隐蔽通道的异常识别率都有所下降,但通信流类别字段的异常识别率仍然居高不下,这是由于该字段的值存在高度集中性,少量水印信息的嵌入也会对通道造成较大的影响.同时,高度集中的分布特性也造成了该字段的误检率非常高,即原始流量被错误地识别为异常流量,这从另一方面弥补了一部分通道的隐蔽性缺失,因为攻击者很难分辨这些异常数据中哪些是带有水印信息的,哪些是原始流量,这也增加了对水印信息的识别及攻击难度.

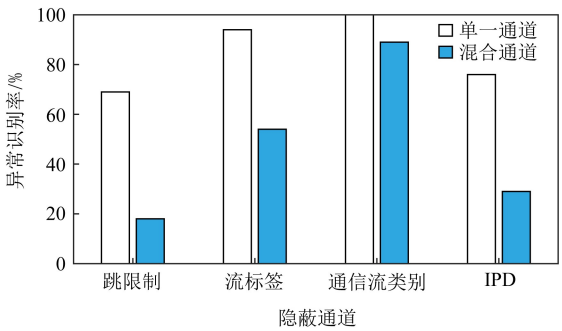


Fig. 15 Accuracy of anomaly detection in the covert channel

图 15 隐蔽通道异常检测识别率

基于混合隐蔽通道的水印生成方法是基于 IPv6 流量特征的全局统计特性来设计的,所以进行水印嵌入的数据包数量对实验结果有很大的影响.根据使用混合隐蔽通道和使用单一通道的异常检测情况,动态地调整不同隐蔽通道的使用比例,更多地使用异常识别率较低的跳限制字段和 IPD 来进行水印信息的嵌入.虽然复合窗口的大小也会对实验结果产生影响,但异常检测更关注的是比例而非具体的数值,所以本实验以水印嵌入度,即嵌入水印信息的数据包占总的数据包数量的比例为变量,测试不同情况下使用混合隐蔽通道的异常识别率.

由图 16 可知,随着水印嵌入度的降低,异常识别率也随着降低,当水印嵌入度降至 0.1 左右时,异常识别率趋于稳定.所以,可以将 0.1 作为水印嵌入度的阈值.同时,实验也比较了不同数据组容量下的异常识别率,当数据组容量增加时,异常识别率有所下降,但其降幅不大,相较于嵌入水印所需的数据包数量的增量而言,收益不高,并且在保证较低异常识别率的前提下,随着数据组容量的增加,水印信息的容量不断被压缩,水印传输的鲁棒性也会受到影响.

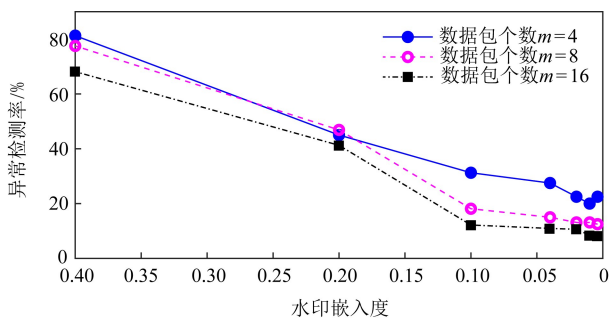


Fig. 16 Effect of watermarking embedding degree on abnormal recognition rate

图 16 水印嵌入度对异常识别率的影响

3.3.2 基于混合时隙的水印方法

网络流水印的隐蔽性主要是为了给水印信息提供安全保障,避免信息在网络传输的过程中被第三方识别或破坏.嵌入水印信息的网络流与原始目标流的相似程度决定了水印的隐蔽性.目前针对水印的检测方法主要包括基于特征的测试和基于规律的测试.基于特征的测试主要使用如方差、均值、分布等一阶的统计量.基于规律的测试通常使用多维的统计数据,机器学习算法也开始被用于水印的发现和检测.本部分将通过 2 种基于特征的测试方法来检验网络流水印的隐蔽性:1)K-L 测试,从相对熵的角度评判水印流和原始流之间的差异性;2)K-S 测试,通过计算水印流和原始流之间的 IPD 的经验累积分布函数的最大值,判断水印的隐蔽性.

如图 17 所示,当观察的时间间隔数量增多后,*K-L* 散度值和 *K-S* 值都趋于稳定,MIBW 方案的 *K-L* 散度值稳定在 0.18 左右,*K-S* 值稳定在 0.1 左右.虽然 MIBW 方案与 IBW 方案、ICBW 方案都是基于时间间隔的方案,但是比起这 2 种方案,其隐蔽性会更优一些.1)因为内部水印的嵌入对原有的间隔进行了二次划分,细化了时间间隔,这就意味着在水印嵌入时减少了人为添加的数据包延迟,*K-L* 散度值会变小;2)由于地基水印间隔对的使用以及内部水印位置的随机性,降低了短时间内无数据包到达间隔的出现几率,使得 *K-S* 值有所下降;另外 MIBW 方案提高了水印信息传输的带宽,这也意味着传输同样长度的水印信息,需要依赖的时间间隔减少,对原始流量的影响也会减少.

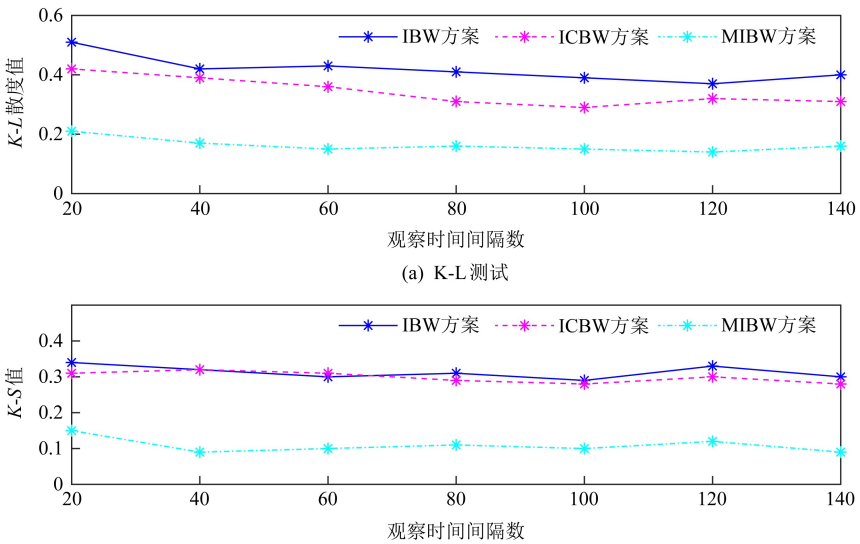


Fig. 17　Test on K-L and K-S

图 17　K-L 及 K-S 测试

针对基于时间间隔的水印方法,多流攻击(MFA)能够发现水印信息的存在,甚至能够通过分析得到水印中的部分参数,给水印的隐蔽性和安全性带来极大的隐患.MFA 测试依赖于收集多条带有水印标记的流,并将这些标记流合并成一个单独的流,通过检查数据包之间的时间间隔来判断水印是否存在.为了验证 MIBW 方案可以比较好地抵御多流攻击,本文将其与另外 2 种基于时间间隔的方法进行比较.在本次测试中,每条混合流包含 5 条单流,测试时长为 10 s,其中 IBW 方案和 ICBW 方案嵌入的水印信息都为固定水印信息,其他的参数都保持一致.

如图 18 所示,MIBW 方案相较于 IBW 方案和 ICBW 方案并没有出现明显的数据包空窗区,即没有出现一定时间内无数据包到达的情况,这证明了 MIBW 方案能够有效地抵御多流攻击.1)由于 MIBW 方案嵌入的水印信息并不是固定的二进制序列,而是根据目标流生成的特征水印序列,这样可以保证在多条流中嵌入的水印信息不同.2)基于地基水印进行内部水印的二次嵌入,随机化了内部水印的嵌入位置,降低了水印的规律性.此外,时间间隔的二次划分减少了人为添加的延迟量,能够降低出现数据包空窗区的概率.

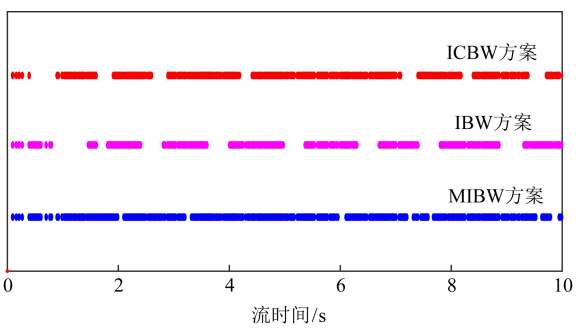


Fig. 18　Packet distribution in the multi-stream attack

图 18　多流攻击数据包分布

4　结束语

IPv6 协议作为未来的主流网络层通信协议,近年来得到了广泛地关注.本文对面向 IPv6 环境的水印生成技术进行了重点研究,结合 IPv6 协议的报文

标准及中间节点不分片的机制,基于目标流关联的特征水印序列,针对不同的网络传输场景,制定了包依赖和时间依赖的水印嵌入方式,在保证一定准确率的前提下,降低水印嵌入对原始流量的整体影响,提高水印的隐蔽性,为追踪 IPv6 网络空间威胁提供手段.

致谢 本文由国家网络空间国际治理研究基地(东南大学)以及 CERNET 华东(北)地区网络中心提供实验环境支撑.

参 考 文 献

[1] National Computer Network Emergency Response Technical Team/Coordination Center of China. China's Internet network security monitoring data analysis report in the first half of 2020 [R]. Beijing: National Computer Network Emergency Response Technical Team/Coordination Center of China, 2020 (in Chinese)
(国家互联网应急中心. 2020 年上半年我国互联网网络安全监测数据分析报告[R]. 北京: 国家互联网应急中心, 2020)

[2] Lucas T, Ferreira M, Plachta R, et al. Non-fragmented network flow design analysis: Comparison IPv4 with IPv6 using path MTU discovery [J/OL]. Computers, 2020, 9(2) [2021-06-01]. https://doi.org/10.3390/computers9020054

[3] Zhao Xuqi, Sun Liang, Wang Yijun, et al. Covert channel contruction method based on IPv6 protocol [J]. Communications Technology, 2021, 54(1): 158-163

[4] Cheng C, Wang T, Huang Y. Indoor positioning system using artificial neural network with swarm intelligence [J]. IEEE Access, 2020, 8: 84248-84257

[5] Dib J, Sirlantzis K, Howells G. A review on negative road anomaly detection methods [J]. IEEE Access, 2020, DOI: 10.1109/ACCESS.2020.2982220

[6] Chen Shaojie, Lang Bo, Liu Hongyu, et al. DNS covert channel detection method using the LSTM model [J]. Computers & Security, 2021, 104: No.102095

[7] Guo Zhaozhong, Shi Liucheng, Xu Maozhi, et al. MRCC: A practical covert channel over monero with provable security [J]. IEEE Access, 2021, DOI:10.1109/ACCESS.2021.3060285

[8] Hua Jingyu, Zhou Zidong, Zhong Sheng. Flow misleading: Worm-Hole attack in software-defined networking via building in-band covert channel [J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 1029-1043

[9] Xie Jinpu, Chen Yonghong, Wang Linfan, et al. A network covert timing channel detection method based on threshold secret sharing [J/OL]. Trans Emerg Telecommun Technol, 31(2) [2021-06-07]. https://doi.org/10.1002/ett.3781

[10] Yang Zhidan, Liu Kesheng, Li Li. Research on network-based covert channels in IPv6 [J]. Journal of Southeast University: Natural Science Edition, 2007, (S1): 141-148 (in Chinese)
(杨智丹, 刘克胜, 李丽. IPv6 中的网络隐蔽通道技术研究[J]. 东南大学学报: 自然科学版, 2007, (增 1): 141-148)

[11] Guo Haoran, Wang Zhenxing, Yu Chong, et al. Analysis and preservation of covert channel based on IPv6 header [J]. Computer Engineering, 2009, 35(14): 160-162, 165 (in Chinese)
(郭浩然, 王振兴, 余冲, 等. 基于 IPv6 报头的隐蔽通道分析与防范[J]. 计算机工程, 2009, 35(14): 160-162, 165)

[12] Tian Jing, Xiong Gang, Li Zhen, et al. A survey of key technologies for constructing network covert channel [J]. Secur Commun Networks, 2020: No.8892896

[13] Mavani M, Ragha L. Covert channel in IPv6 destination option extension header [C] //Proc of 2014 Int Conf on Circuits, Systems, Communication and Information Technology Applications (CSCITA). Piscataway, NJ: IEEE, 2014: 219-224

[14] Bernhards B, Mauno P, Markus K, et al. Creating and detecting IPv6 transition mechanism-based information exfiltration covert channels [C] //Proc of the 21st Nordic Conf on Secure IT Systems. Oulu: NordSec, 2016: 85-100

[15] Wojciech M, Krystian P, Luca C. IPv6 covert channels in the wild [C] //Proc of the the 3rd Central European Cybersecurity Conf. Munich: CECC, 2019: 1-6

[16] Guo Xiaojun, Cheng Guang, Zhu Chengang, et al. Progress in research on active network flow watermark [J]. Journal on Communications, 2014, 35(7): 178-192 (in Chinese)
(郭晓军, 程光, 朱琛刚, 等. 主动网络流水印技术研究进展[J]. 通信学报, 2014, 35(7): 178-192)

[17] Wang Xinyuan, Reeves D. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays [C] //Proc of the 10th ACM Conf on Computer and Communications Security. Washington DC: U.S. Department of Commerce, 2003: 20-29

[18] Houmansadr A, Kiyavashyn N, Borisov N. Rainbow: A robust and invisible non-blind watermark for network flows [C] //Proc of the 16th Network and Distributed System Security Symposium. San Diego: University of California, 2009: 224-236

[19] Huang Junwei, Pan Xian, Fu Xinwen, et al. Long PN code based DSSS watermarking [C] //Proc of IEEE INFOCOM. Piscataway, NJ: IEEE, 2011: 2426-2434

[20] Lacovazzi A, Sarda S, Elovici Y. INFLOW: Inverse network flow watermarking for detecting hidden servers [C] //Proc of IEEE INFOCOM. Piscataway, NJ: IEEE, 2018: 747-755

[21] Pyun Y, Park Y, Wang Xinyuan, et al. Tracing traffic through intermediate hosts that repacketize flows [C] //Proc of IEEE INFOCOM. Piscataway, NJ: IEEE, 2007: 634-642

[22] Houmansadr A, Borisov N. Swirl: A scalable watermark to detect correlated network flows [C] //Proc of the Network and Distributed System Security Symposium. San Diego, California: NDSS, 2011: 1-15

[23] Lin Mao, Liu Guangjie, Liu Weiwei, et al. Network flow watermarking method based on centroid matching of interval group [C] //Proc of the 3rd IEEE Int Conf on Progress in Informatics & Computing. Piscataway, NJ: IEEE, 2015: 628-632

[24] Xu Xiaoqiang, Zhang Jing, Li Qianmu. Equalized interval centroid based watermarking scheme for stepping stone traceback [C] //Proc of the 1st IEE Int Conf on Data Science in Cyberspace. Piscataway, NJ: IEEE, 2016: 109-117

[25] Liu Weiwei, Liu Guangjie, Xia Yang, et al. Using insider swapping of time intervals to perform highly invisible network flow watermarking [J]. Security and Communication Networks, 2018, 2018(4): 1-16.

[26] Yu Wei, Fu Xinwen, Graham S, et al. DSSS-based flow marking technique for invisible traceback [C] //Proc of the 2007 IEEE Symp on Security and Privacy. Piscataway, NJ: IEEE, 2007: 18-32

[27] Liu Anyi, Chen J, Wrahsler H, et al. Real-time timing channel detection in a software-defined networking virtual environment [J]. Intelligent Information Management, 2015, 7(6): 283-302

Tao Jun, born in 1975. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include cyber security, Internet of things.

陶　军,1975 年生.博士,教授,博士生导师.CCF 高级会员.主要研究方向为网络安全、物联网技术等.

Zhu Zhenchao, born in 1982. PhD, associate professor. Member of CCF. His main research interests include IoT security, privacy preserving.

朱珍超,1982 年生.博士,副教授.CCF 会员.主要研究方向为物联网安全、隐私保护.

Wang Zhaoyue, born in 1996. Master. Her main research interest is Internet measurement.

王昭悦,1996 年生.硕士.主要研究方向为网络测量.

Li Wenqiang, born in 1996. Master candidate. His main research interest is Internet measurement.

李文强,1996 年生.硕士研究生.主要研究方向为网络测量.

Sun Weice, born in 1993. PhD candidate. His main research interests include matrix/tensor theory, network measurement and traffic analysis.

孙炜策,1993 年生.博士研究生.主要研究方向为矩阵/张量理论、网络测量和流量分析.