软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(9):2791−2814 [doi: 10.13328/j.cnki.jos.005517]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

层次化反匿名联盟构建方法*

鲁 宁[1,2], 李 峰[1], 王尚广[2], 史闻博[1], 杨放春[2]

[1](东北大学 信息科学与工程学院,辽宁 沈阳 110819)

[2](网络与交换技术国家重点实验室(北京邮电大学),北京 100876)

通讯作者: 李峰, E-mail: lifeng@neuq.edu.cn

摘 要: IP 匿名是当前互联网协议中最具威胁的安全漏洞,它会引发一系列安全、管理和计费问题.基于对等过滤的域间源地址验证方法通过构建反匿名联盟,能够利用当前已广泛实现、轻量的 Egress Filtering 有选择性地将流向联盟成员的匿名包清理掉,在保证高效的同时兼具部署激励性.然而,现有方法存在以下问题:过于扁平化、单一化的联盟体系结构,使得其过滤器需求量和成员更新传播范围随联盟规模的扩张而急剧增大;过于随机的非成员判定方式和低效的数据处理方式,使得其过滤规则优化算法的计算开销和精度都有待完善.对此,提出了一种层次化的基于对等过滤的反匿名联盟构建方法.通过理论分析和基于大规模真实互联网拓扑的仿真实验结果表明:相比以往同类典型方案,该方法在继承其优势的同时改善了过滤器开销、通信开销、计算开销和优化精度.

关键词: IP 匿名;源地址验证;出边界过滤;对等过滤;层次化

中图法分类号: TP393

中文引用格式: 鲁宁,李峰,王尚广,史闻博,杨放春.层次化反匿名联盟构建方法.软件学报,2019,30(9):2791−2814. http://www.jos.org.cn/1000-9825/5517.htm

英文引用格式: Lu N, Li F, Wang SG, Shi WB, Yang FC. Hierarchical anti-spoofing alliance construction approach. Ruan Jian Xue Bao/Journal of Software, 2019,30(9):2791−2814 (in Chinese). http://www.jos.org.cn/1000-9825/5517.htm

Hierarchical Anti-Spoofing Alliance Construction Approach

LU Ning[1,2], LI Feng[1], WANG Shang-Guang[2], SHI Wen-Bo[1], YANG Fang-Chun[2]

[1](College of Information Science and Engineering, Northeastern University, Shenyang 110819, China)

[2](State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing 100876, China)

Abstract: IP spoofing, as one of the most threatening security flaws in the current Internet, can bring a series of issues about network management and telecommunications billing. For this reason, the researchers propose the mutual egress filtering based defense mechanism, which uses the best current anti-spoofing practice, i.e., egress filtering, to clean the anonymous packets with high-efficiency, and simultaneously increase the incentive deployment through constructing the anti-spoofing alliance. However, the existing work has the following disadvantages: the flat and plain architecture leads to the higher overhead on the filter and communication; the inefficient data processing and non-member identification leads to the higher computation overhead and the lower precision of filter optimization. Therefore, this study proposes a hierarchical anti-spoofing alliance construction approach based on mutual egress filtering. Extensive mathematical analysis and simulations are performed to evaluate the proposed approach. The results show that the proposed approach

* 基金项目: 国家自然科学基金(61601107, 61402094); 河北省自然科学基金(F2015501122, F2015501105); 辽宁省博士科研启动基金(F201501143)

Foundation item: National Natural Science Foundation of China (61601107, 61402094); the Natural Science Foundation of Hebei Province (F2015501122, F2015501105); the Doctoral Scientific Research Foundation of Liaoning Province (F201501143)

收稿时间: 2017-08-22; 修改时间: 2017-10-09; 采用时间: 2017-11-13; jos 在线出版时间: 2018-04-27

CNKI 网络优先出版: 2018-04-27 14:57:58, http://kns.cnki.net/kcms/detail/11.2560.TP.20180427.1457.002.html

significantly outperforms the prior approaches in terms of the filter overhead, communication overhead, computation overhead, and the precision of filter optimization.

Key words: IP anonymous; source-address validation; egress filtering; mutual egress filtering; hierarchical

当前,互联网转发数据包的依据是目的IP地址,通常不对源地址做真实性检查,这就使得网络中充斥了大量源地址虚假的匿名包,从而给网络安全运营带来了管理和计费问题.因此,如何在互联网上建立面向源IP地址的真实性验证机制,解决其安全性弱、可信度低问题,进而为用户提供高度可信的网络服务,就显得极其重要.

与互联网本身的分层结构相对应,源地址验证体系结构通常也划分为 3 个层次,自底向上分别为接入网络地址验证、域内网络地址验证和域间网络地址验证,其中:前两层都部署在自治域内,任务是阻止匿名包在域内网络传播;第 3 层通常部署在域边界路由器,主要任务是通过识别和过滤域间匿名流来实现自治域粒度的网络保护.由于自治域之间的关系要远比域内关系复杂,因此域间网络地址验证就成为整个验证体系结构中最为关键的一个层次.按照是否需要对已有地址和路由协议进行扩展和改革,域间源地址验证研究方法一般又可分为“革新型”和“改良型”两大类.其中,前者集中于设计一种新的可绑定身份安全机制的地址系统和路由协议,以达到从根本上解决源地址伪造问题的目的[1−6];而后者主要关注如何在保证现有地址体系和路由协议稳定的前提下,通过增加检测机制来弥补现有网络体系结构的缺陷[7−9],因此更适合在现有网络上部署.事实上,确实已有 3 种经典的“改良型”域间源地址验证方法被当前路由器市场所广泛接受,它们分别是反向路径转发(unicast reverse path forwarding,简称 uRPF)[7]、入边界过滤(ingress filtering,简称 I-filtering)[8]和出边界过滤(egress filtering,简称 E-filtering)[9].而本文主要关注其中过滤精度较高、操作开销较小的 E-filtering 方法.

虽然 E-filtering 因性能优势而倍受主流网络设备提供商的青睐,但是当前互联网至今仍有 38.5%的自治域未加入反匿名联盟,这就说明它并未被网络服务提供商(Internet service provider,简称 ISP)所广泛接受,其中的主要原因是它没有遵循“谁部署、谁受益”的激励原则,存在搭便车问题,阻碍了它的推广应用.针对该问题,已有研究者提出了一种基于对等过滤的域间源地址验证方法(mutual egress filtering,简称 MEF),其主要任务就是构建一种面向 Stub 域的反匿名联盟,将未部署 E-filtering 的 Stub 域从受益者列表中严格剥离,只有联盟成员才享有域间地址安全的权利[9].为了完成上述功能,该系统首先需要随时发现联盟成员状态(加入和退出)的变化,并将相关信息广播给其他成员,然后在联盟成员的 Stub 域边界路由器的访问控制列表(access control list,简称 ACL)上配置面向对等成员的 E-filtering 过滤规则.但是该方法存在可扩展性差、难以适应增量部署问题,原因如下:过于扁平化、单一化的联盟体系结构,使得过滤器需求量和成员更新传播范围随联盟规模的扩张而急剧增大;过于随机的非成员判定方式和低效的数据处理方式,使得过滤规则优化的计算开销和精度都有待改善.

针对上述挑战,本文提出一种层次化的基于对等过滤的反匿名联盟构建方法(hierarchical anti-spoofing alliance construction approach based on egress filtering,简称 EAGLE).与已有扁平化的反匿名联盟构建方法相比,本方法着力解决了以下 3 个问题:(1) 立足于实际自治域网络可分层的拓扑结构,通过对联盟成员分级划分,构建了新型的层次化的反匿名联盟体系结构,摒弃了传统方法中扁平的、单一的体系结构,避免成员之间建立不必要的全连接双向过滤关系,实现了过滤规则的精简,进而降低了过滤器需求量;(2) 通过引入 Transit 域对等过滤模块作为联盟边界,将每一层级联盟和外界成员隔离,使得不同层级联盟内部网络环境彼此互不可见,在确保域间高速通信的同时,排除了拓扑结构和成员变化带来的影响,有效控制了过滤规则信息更新的范围和频度,进而降低了系统通信开销;(3) 聚焦因联盟划分不均造成高层成员的过滤器资源不足问题,通过定量分析网络前缀与过滤规则之间的关系,结合联盟成员动态变化的特征,设计一种高精度、可增量更新的过滤规则优化算法,在加快求解速度的同时提高解的质量.

为了验证本文提出的 EAGLE 方法,首先对其高效性进行了理论分析;然后,在基于真实互联网拓扑的网络仿真环境下中对其进行了实验验证,并与其他经典方法进行了对比.结果表明:EAGLE 不仅延续了传统方法的优势,而且通过改善过滤器开销、通信开销、计算开销和优化精度来提高可扩展性.

本文第 1 节给出扁平化的基于对等过滤的反匿名联盟构建方法形式化描述.第 2 节介绍本文提出的 EAGLE 方法,其中,第 2.1 节介绍层次化反匿名联盟的基本概念,第 2.2 节给出联盟的构建方法,包括系统结构模

型、协同流程、安全策略、可靠性策略、过滤规则优化策略等实现细节.第 3 节对 EAGLE 的性能进行评估,其中,第 3.1 节给出理论评估,第 3.2 节则采用实验仿真手段对分析结果进行补充.第 4 节介绍相关工作.第 5 节总结全文并指出下一步的工作重点.

1 回顾扁平化的基于对等过滤的反匿名联盟构建方法

本节首先构建基于 C2P 商业关系的层次网络模型,说明上下层自治域的网络前缀应该存在包含关系;然后回顾基于对等过滤的反匿名联盟构建方法,证明它可以解决传统方法部署激励性缺乏的问题;最后,通过指出该方法存在可扩展性差等问题,表明本文研究动机.

1.1 基于C2P商业关系的层次网络模型

一方面,研究者通过分析自治域(automous system,简称 AS)路由策略,发现 AS 路径符合“无谷模型”的特征,即存在严格的层次结构,下层 AS 只有通过上层或同层 AS 才能将它的数据包路由转发出去.本文将这种由上层向下层提供穿越服务并根据流量收费的商业关系称为客户-提供商关系(customer-to-provider,简称 C2P),上层 AS 称为提供商,下层 AS 就是它的客户.另一方面,当前互联网为了解决因路由规模过大而造成路由器的处理能力和内存分配趋于饱和问题,广泛采用了更具层次化的无类别域间路由(classless inter-domain routing,简称 CIDR)技术来为 AS 分配前缀地址块,使得下层 AS 的网络前缀能够直接取自它的上层,进而完成下层到上次的路由聚合,减少上层网络中明细路由的数量.基于此,本文建立了一种基于 C2P 商业关系的层次网络模型.

如图 1 所示,该模型是典型的树形结构,其中,Stub AS 是叶子节点,负责生成数据流,它只能充当 Customer;而 Transit AS 是非叶子节点,负责转发数据流,既能充当 Customer,又能充当 Provider.此外,该模型具备以下特征:(1) 只包含单宿主自治域;(2) 客户地址块全部取自上层提供商;(3) 不包含对等体-对等体(peering-to-peering,简称 P2P)和同胞-同胞(sibling-to-sibling,简称 S2S)关系,只突出 C2P 商业关系.

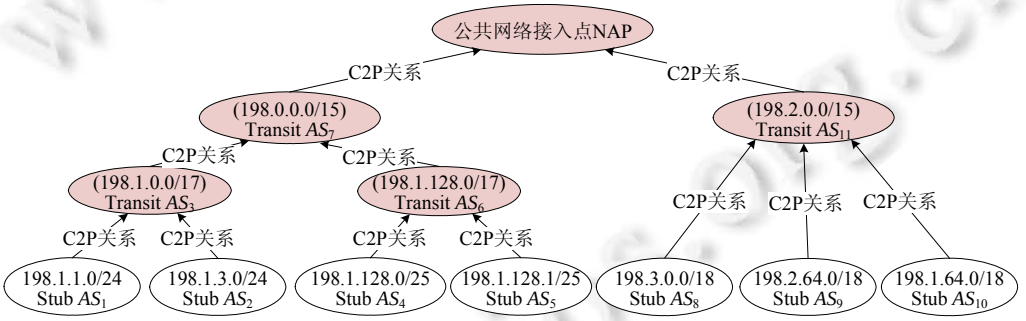

Fig.1 Hierarchical network model based on C2P business relationship

图 1 基于 C2P 商业关系的层次网络模型

提出上述假设的原因在于[10]:

(1) 互联网中虽然存在一个客户连接多个提供商的多宿主现象,但是这会引入大量无法聚合的明细路由.因而,绝大多数客户仍然在使用单宿主连接方式,只有当网络规模超过早期容量规划时或为了实现备用路由和可靠通信,才会购买不同提供商的地址块.然而,即使在多宿主连接下,我们也可按照地址块将客户逻辑划分为多个单宿主,而这种逻辑单宿主依然适合本文关注的出边界过滤技术,边界路由器就是先前客户与提供商的连接接口;

(2) 虽然存在因历史遗留或临时更换提供商而造成客户的地址块与提供商不相符的现象,但是这种连接方式非常少见,因为它相当于在提供商的地址空间钻了个洞,使得它们不得不针对新引入地址块设置明细路由.不过,即使客户采用这种连接,也只是妨碍局部网络的层次化,但并不影响系统的正常运行以及整个联盟的层次化.基于此,该模型未包含这种非常规连接;

(3) P2P 和 S2S 用于说明同层 AS 及其客户之间是否直接可达.就同层直连 AS 产生的匿名流来说,采用入边界过滤(ingress filtering,简称 I-Filtering)来防御更加合适,因为这种场景既不会引发 I-Filtering 的漏报率较高问题,也不会带来较大的操作开销.例如:假设 AS A 和 AS B 同层直连,那么在 A 和 B 的直连入接口上配置两条规则:1) permit AS_B AS_A;2) deny any any.其中:规则 1)允许所有源地址包含在 AS_B,且目的地址包含在 AS_A 的 IP 包进入自治域 A;规则 2)则指出,凡是不满足前一规则的数据包都被过滤.由于本文偏向出边界过滤,因此不包含 P2P 和 S2S.

定义 1. 基于 C2P 的层次网络模型定义为前缀树 $G=(V,E,A)$,其中,节点集 $V=\{v|v$ 是 G 中节点$\}$,边集 $E=\{(u,v)|u,v\in V$ 且 u,v 互为父子$\}$,前缀集 $A=\{a(v)|a(v)$是节点 v 的前缀地址$\}$,$\forall u,v\in V$ 且 $a(u)\subset a(v)$,$\exists u$-v 路径.

定义 2. 给定层次网络模型 $G=(V,E,A)$,如果$\forall u\in V,\nexists v\in V$, s.t. $a(v)\subset a(u)$,那么 u 为 Stub 节点,记为 u_{stub},所有 Stub 节点构成 V_{stub} 集合,记为 V_{stub};反之,若存在 $v\in V$,使得条件成立,u 为 Transit 节点,记为 u_{trst},所有 Transit 节点构成 $V_{transit}$ 集合,$V_{stub}\cup V_{transit}=V$.

定义 3. 匿名流定义为三元组 $F=(s,i,d)$,其中,s 表示发送匿名流的自治域,i 表示匿名流所携带源地址的域前缀,d 表示被匿名流攻击的自治域.一般情况下,在正常网络环境中,给定数据流 F,都有 $a(F.s)=F.i$.然而在匿名网络环境下,$\exists$匿名流 $F=(s,i,d)$, s.t. $F.i\neq a(F.s)$和 $F.i\in\Omega_{stub}$,其中,Ω_{stub} 表示网络中 Stub 域网络前缀的全集.针对不同的网络协议漏洞,攻击者通过伪造源地址能够发起多种匿名攻击.根据源地址与攻击目标的联系,匿名攻击可划分为直接攻击和间接攻击:前者是由攻击者直接向受害主机发起匿名流,其中,匿名流的目的地址就是指向攻击目标,而源地址可设置为任何虚假 IP,常见类型有 SYN 风暴攻击;后者利用网络协议中请求和回复包数量不对称的特点,先由攻击者向中转服务器发起少量匿名请求,使它误以为该请求由受害主机发起,进而向受害主机发送大量回复信息,造成其瘫痪,常见类型有 DNS 放大攻击.基于上述分析,给定匿名流 $F=(s,i,d)$,如果 F 属于前者,那么 d 是指匿名流所携带目的地址的域前缀;如果 F 属于后者,那么 d 是指匿名流所携带源地址的域前缀.为了同时防御二者,反匿名联盟成员之间既不应该彼此发送匿名流,也不应该发送源地址涉及彼此网络前缀的匿名流,也就是说,凡是目的地址或源地址隶属于成员网络前缀的匿名流都应该被过滤.

1.2 扁平化的反匿名联盟构建方法FAGLE

当前,互联网协议通常不对源地址做真实性检查,因此,攻击者能够将其他自治域 IP 地址写入数据包的源地址字段,误导受害者.本文将这种源地址虚假的数据流称为匿名流,如定义 3 所描述.出边界过滤(egress filtering,简称 E-filtering)技术的主要任务就是过滤匿名流和净化域间网络,其基本原理正是利用第 1.1 节提到的数据流源地址与自治域网络前缀的隶属关系来判定数据流的合法性,具体过程是在边界路由器的访问控制列表(access control list,简称 ACL)上配置特定的包过滤规则,对每个流出数据包进行验证:如果发现其源地址不属于本网络范围,则丢弃;反之,正常转发.本文将这种配置 ACL 过滤规则的边界路由器称为边界过滤路由器(border filtering router,简称 BFR).以图 1 为例,假设 Stub 域 AS_2 的网络前缀是 198.1.2.0/24,那么对于所有从 AS_2 流出的数据包来说,其源地址必须隶属于该前缀,否则就过滤它.基于此,AS_2 的边界路由器需要配置以下两条规则:(1) permit AS_2 any; (2) deny any any.其中:规则(1)允许所有源地址包含在网络前缀 AS_2 的数据包正常转发;规则(2)指出凡是不满足规则(1)的数据包都被过滤,无论其源地址和目的地址是多少.本文将部署 E-Filtering 的 Stub 域记为 EStub,将具备匿名包过滤能力的网络称为反匿名网络(anti-spoofing network,简称 ASN).虽然 E-Filtering 因轻量、高效等优点在提出伊始就被标准化,然而近年的网络测量惊人地发现,该方法的部署率连续多年都维持在一个极低的水平.造成这种后果主要原因是,它缺乏部署激励性.在本文中,部署激励(incentive for deployment,简称 Inc)就是指网络服务提供商对一个反匿名方法的部署意愿程度,它通常表现在部署收益和非部署收益两方面:前者主要针对已部署反匿名机制的自治域,其部署收益越大,说明该方法的部署激励越大,反之越小;后者主要针对尚未部署反匿名机制的自治域,若它们总能不劳而获,那么收益越大,方法部署激励就越小,反之越大.基于此,本文借鉴文献[11]定义部署激励的思想,首先采用对偶方式来定义部署收益 f_1 和非部署收益 f_2,然后通过逐一累加反匿名网络中所有自治域的收益就可评估出方法的部署激励,如定义 4 所述.

定义 4. 部署激励可被定义为一个累加函数.

给定反匿名网络 $ASN=(G,R_{A\text{-}stub})$:

$$Inc(ASN)=\sum_{u_{Estub}\in R_{A\text{-}stub}} f_1(u_{Estub})+\sum_{u_{stub}\in R_{C\text{-}stub}} f_2(u_{stub}),$$

其中, $f_1(u_{Estub})=F_{Filter}^{u_{Estub}}/F_{Total}^{u_{Estub}}$ 表示 u_{Estub} 的部署收益, $f_2(u_{stub})=-(F_{Filter}^{u_{stub}}/F_{Total}^{u_{stub}})$ 表示 u_{stub} 的非部署收益, $F_{Total}^{u_{Estub}}$ 和 $F_{Total}^{u_{stub}}$ 分别表示在匿名网络上所有流向自治域 u_{Estub} 和 u_{stub} 的匿名流的数量, $F_{Filter}^{u_{Estub}}$ 和 $F_{Filter}^{u_{stub}}$ 分别表示在反匿名网络中所有流向自治域 u_{Estub} 和 u_{stub} 且被过滤的匿名流的数量.

为了强化 E-Filtering 的激励功能,Liu 等人借鉴社会学的互动关系理论,定义了一种基于对等过滤的反匿名联盟[9],其中,对等过滤(mutual egress filtering,简称 MEF)是指 Stub AS_i 能够阻止流向 Stub AS_j 匿名流的充分条件是 AS_j 也阻止流向 AS_i 的匿名流,而反匿名联盟(anti-spoofing alliance,简称 AA)就是由具备对等过滤关系 AS 组成的集合.虽然反匿名网络上可能存在多个 AA,甚至其成员彼此交叉,但通过定义 4 不难推断出:为了最大化激励效果,所有成员域都应该加入同一个 AA.因此,不失一般性,本文假设整个网络只有一个 AA.部署 MEF 方法的 Stub 域称为 MStub.在已有的方法中,反匿名联盟的所有成员都是 MStub,而且它们必须保持完全双向过滤关系,根据定义 5,它被称为扁平化的反匿名联盟(flat AA,简称 FAA).

定义 5. 扁平化的反匿名联盟定义为 $FAA=\{u_{MStub}\in R_{A\text{-}stub}|$具备对等过滤关系的 MStub 节点$\}$.∀数据流 $F=(s,i,d)$,已知 $F.s\in FAA,F$ 满足以下条件:(1) 如果 $F.d\in FAA$,那么 $F.i=a(u_{MStub})$;(2) 如果 $F.d\notin FAA$,那么 $F.i=a(\underline{F.s})$ 或 $F.i\in AP$,其中, $AP=\bigcup_{v\in FAA}a(v)$, $AP=\Omega_{stub}-AP$.

如图 2 所示:在 FAA 中,每个 MStub 的边界路由器都需配置 4 组 ACL 规则.

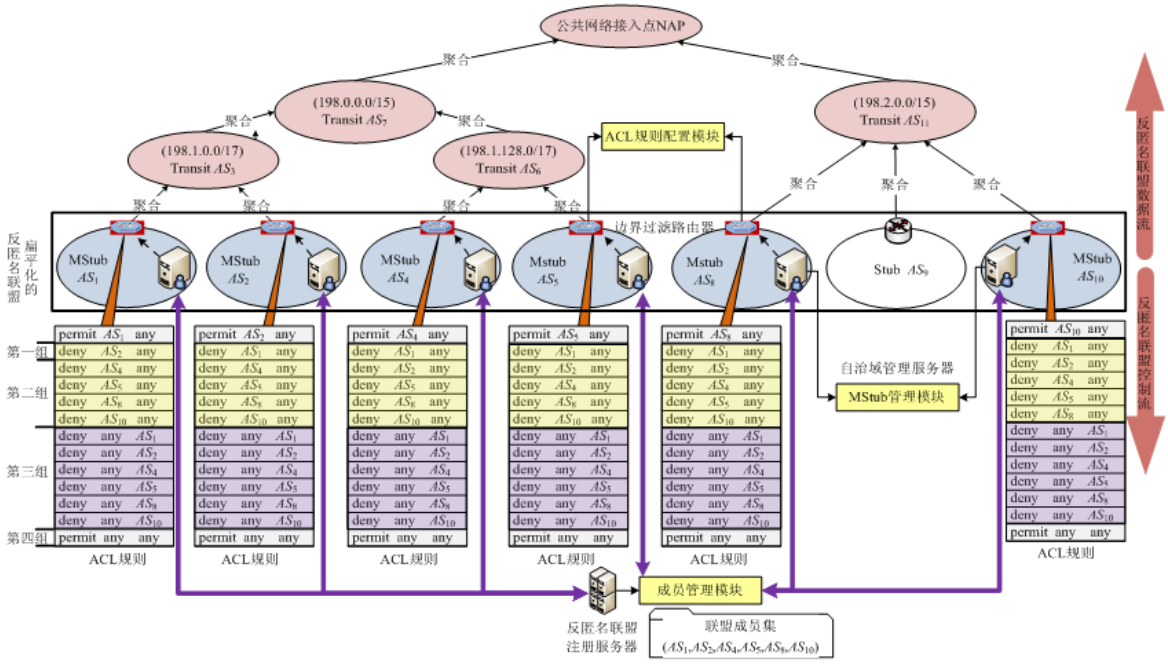

Fig.2 An example of FAA

图 2 扁平化的反匿名联盟举例

以 AS_1 为例,它的边界过滤路由器配置规则如下:(1) permit AS_1 any;(2) deny AS_2 & AS_4 & AS_5 & AS_8 & AS_{10} any;(3) deny any AS_1 & AS_2 & AS_4 & AS_5 & AS_8 & AS_{10};(4) permit any any.其中,

- 规则(1)允许通过源地址包含在 AS_1 网络前缀的数据包;
- 在前面规则不满足的条件下,规则(2)指出:无论目的地址是多少,所有源地址包含于成员域网络前缀的数据包都将被过滤;

- 在前面规则都不满足的条件下,规则(3)指出:无论源地址是多少,所有目的地址包含于成员域网络前缀的数据包也都将被过滤;
- 规则(4)则指出,凡是不满足规则(1)~规则(3)的数据包都将允许通过.

在基于 MEF 的反匿名网络 $ASN=(G,R_{A\text{-}stub})$中,$R_{A\text{-}stub}$ 等于反匿名联盟 AA.∀匿名流 $F=(s,i,d)$,如果 $d\in R_{A\text{-}stub}$,那么 $F.i=a(F.s)$;如果 $d\in R_{c\text{-}stub}$,那么 $F.i\neq a(F.s)$.与 E-Filtering 相比,MEF 能防止非部署者不劳而获,而仅让部署者受益,有效地解决了搭便车问题,使得非部署收益 f_2 恒等于 0.

1.3 研究动机

在扁平化的反匿名联盟中,

- 一方面,为确保对等过滤机制能够有效运转,MStub 域的边界路由器必须维护面向全部成员的过滤规则,这就产生以下问题:
  (1) 过滤器需求量过大.所谓过滤器是指访问控制列表 ACL 的表项,每个表项对应一条 ACL 规则.过滤器不足就意味着 ACL 过滤规则数量超过了其表项,例如:当联盟成员域数量为 N 时,每个域的边界过滤路由器都需要配置 $o(2N)$量级的 ACL 过滤规则.虽然当前路由器大都已提供 ACL 过滤平台,但是它们全都由昂贵的三态内容寻址存储器(ternary content addressable memory,简称 TCAM)来实现,为此,路由器设备商只能通过限制每台路由器的过滤器数量来降低成本[12,13].以常见的中端路由器 Cisco Catalyst4500 为例,它的 ACL 平台最多只有 64 000 个过滤器,而且同时还被 384 个接口共享;
  (2) 较大的通信开销.例如:当联盟成员域数量为 N 时,每当新成员加入或旧成员退出的时候,规则更新消息都将辐射到整个联盟范围,其代价达到 $o(N)$;
- 另一方面,当出现过滤规则数量超过过滤器需求的时候,MSM 模块会执行过滤规则优化算法.然而已有算法的效率较低,原因如下:非成员列表作为优化算法的重要输入参数,却采用随机性较大的排他法来判定,这会影响优化解的质量,使得过滤器利用率较差;面对不断变更的联盟成员,每次都只能将新旧数据重复处理,这会增加计算开销,影响系统响应时间.

简言之,以上问题终究会制约反匿名联盟的可扩展性.为此,本文希望在继承原有方法优势的前提下,构建一种层次化的体系结构,取代扁平化联盟的双向过滤关系,以达到降低通信开销和过滤器需求量的目的;设计一种可增量更新、高精度的过滤规则优化算法:一方面,利用已测量的拓扑数据来判定非成员域;另一方面,通过参考历史计算结果来处理新增数据,并将它们融合以求得最优解.

2 一种层次化的基于对等过滤的反匿名联盟构建方法

针对上述问题,本文提出了一种可自下而上分层、基于对等过滤的反匿名联盟构建方法(hierarchical anti-spoofing alliance construction approach based on egress filtering,简称 EAGLE).本节将详细阐述本方法的设计思想和具体实现机制.

2.1 层次化反匿名联盟的基本概念

为了保证部署激励性,本方法依然采用对等过滤思想来建立反匿名联盟.随着越来越多的 Stub 域加入联盟,反匿名网络的局部区域可能出现以下情形:与提供商 Transit u_{trst} 直接关联的所有客户 Stub 都成为 MStub,例如,图 3 中 Transit AS_3 的两个客户 Stub 域 AS_1 和 AS_2 都是 MStub.本文将它们前缀的并集 $AS_3\cup AS_2\cup AS_1$ 称为内部前缀(inner prefix,简称 IN_p).除此之外,联盟其他成员前缀的并集统称为外部前缀(out prefix,简称 OUT_p).就 AS_1~AS_3 发送的匿名流来说,根据源地址 src 和目的地址 dst 的可能组合方式,它们分为以下 4 种:$src\in IN_p\wedge dst\in IN_p$,$src\in IN_p\wedge dst\in OUT_p$,$src\in OUT_p\wedge dst\in IN_p$,$src\in OUT_p\wedge dst\in OUT_p$.前 3 种都涉及区域内成员前缀,称为区域内匿名流(inner anonymous flow,简称 IN_f),第 4 种只涉及区域外成员,称为区域外匿名流(out anonymous flow,简称 OUT_f).鉴于 AS_1~AS_3 是客户-提供商关系,通过路由策略可推断出 OUT_f 必然经由提供商 AS_3 转发.利用这个特点,我们可

以实现层次化的对等过滤,基本原理就是由 AS_1 和 AS_2 来共同过滤 IN_f,而由 AS_3 来过滤 OUT_f.相应的 ACL 规则配置方法如下:首先,鉴于 AS_1 和 AS_2 的任务相同且关系对等,它们的规则配置方法也相同.因此,本文只介绍 AS_1 的规则配置情况.在 AS_1 的边界过滤路由器上,共需配置 4 组规则:(1) permit AS_1 any;(2) deny AS_2 & AS_3 any;(3) deny any AS_2 & AS_3;(4) permit any any.其中,第 1 组用于判断数据包是否真实,第 2 组和第 3 组将侵犯同层 MStub 域的匿名包全部过滤掉,第 4 组用于剪除搭便车的非 MStub 域.如果上述场景发生在第 1.1 节建立的层次网络中,利用提供商包含客户前缀的特点,第 2 组、第 3 组规则可简化为图 3 所示: (2) deny AS_3 any;(3) deny any AS_3.然后,AS_3 的配置规则为:(1) permit AS_1 & AS_2 & AS_3 any;(2) deny AS_4 & AS_5 & AS_6 & AS_7 any;(3) deny any AS_4 & AS_5 & AS_6 & AS_7;(4) permit any any.利用层次网络的前缀包含关系,第 1 组~第 3 组规则则可简化为图 3 所示:(1) permit AS_3 any;(2) deny AS_7 any;(3) deny any AS_7.到此为止,完成了由$\{AS_1,AS_2,AS_3\}$组成的局部区域的层次化过滤.从联盟其他成员来看,该区域可被看为一个规模较大、以 IN_p 为前缀的可信逻辑 Stub(trusted logical mef-stub,简称 TMStub),图 3 将它称为 $sub\text{-}TMStub_1$.随着联盟规模的进一步扩大,就有可能生成更高层级的 TMStub.例如,随着 $sub\text{-}TMStub_1=\{AS_1,AS_2,AS_3\}$和 $sub\text{-}TMStub_2=\{AS_4,AS_5,AS_6\}$的加入,即可生成 $TMStub_3=\{sub\text{-}TMStub_1,sub\text{-}TMStub_2,AS_7\}$.基于此,本文将 TMStub 表示为一个嵌套二元组(u_{Trst},IS_{MStub}),如定义 6 所示,其中,u_{Trst} 表示边界 Transit 域,IS_{MStub} 表示内部其他成员集合.通过层次网络的前缀包含关系,可推断出 TMStub 的前缀就是 u_{trst} 的前缀.综上所述,TMStub 所配置过滤规则的特点可归纳如下:虽然边界规则较为复杂(最高层边界需要与其他成员实现完全双向过滤),但是内部规则较为简单(只需 4 条)和稳定(既不因外部成员的变化而频繁更新,又不因联盟规则的增大而加大过滤器需求量).

本文将这种层次化的反匿名联盟(hierarchical anti-spoofing alliance,简称 HAA)表示为$\{MStub_1,\ldots, MStub_n, TMStub_1,\ldots,TMStub_m\}$,其中,$MStub_i$ 是 MStub 类型的成员域,$TMStub_i$ 是 TMStub 类型的成员域,如定义 7 所示.与扁平化联盟中单一双向过滤不同,HAA 的过滤场景需要被扩展成 3 类.

(1) 最高层级联盟成员之间过滤——在反匿名联盟中,最高层级成员之间互为过滤端,把侵犯彼此的匿名流全部过滤.在这类场景中,每个成员的 ACL 规则都要满足双向过滤,并且要实现物理 MStub 之间、TMStub 之间以及物理 MStub 与 TMStub 之间的匿名包过滤,例如图 3 中 $HAA=\{TMStub_3,Stub_8,Stub_{10}\}$,它们之间需要配置双向过滤规则,因此该层次的规则配置最为复杂,开销也最大;

(2) TMStub 内部过滤——TMStub 的内部成员域(其成员类型可能是 MStub 或 sub-TMStub)作为单向过滤端,把侵犯其他内部域的所有匿名流全部过滤.在这类场景中,内部成员域只需配置面向 TMStub 网络前缀的 ACL 规则,例如图 3 中 $TMStub_3=\{u_{Trst},IS_{MStub}\}$,其中,$u_{Trst}=AS_7$,$IS_{MStub}=\{sub\text{-}TMStub_1,sub\text{-}TMStub_2\}$.$u_{Trst}$ 是边界 Transit 域,鉴于 $TMStub_3$ 已属于最高层级成员,因此 u_{Trst} 需配置双向过滤规则;而 $sub\text{-}TMStub_1$ 和 $sub\text{-}TMStub_2$ 作为内部成员,分别配置面向 AS_7 的过滤规则即可.该场景的规则配置最简单,开销也较小;

(3) 反匿名联盟与非成员之间过滤——联盟成员不过滤任意流向非成员域的匿名包.在这类场景中,每个成员既不应该单独配置面向非成员域的 ACL 规则,也不应该配置可能涵盖非成员域地址块的 ACL 规则.例如图 3 中 $Stub_9$ 是非联盟成员,成员域不会配置任何涉及 $Stub_9$ 网络前缀的 ACL 规则,允许匿名流进入该域.

综上所述,虽然定理 1 已经证明 FAA 和 HAA 在过滤效果和激励效果方面理论上相同,但是后者通过引入 TMStub,使得 Mstub 之间不必保持完全双向过滤关系,降低了联盟建立开销:(1) 成员更新不会辐射到全联盟的 Mstub 域,只在最高层级联盟成员之间传播,无需通知 TMStub 内部成员,减少了通信开销;(2) 过滤规则配置数量与 Mstub 数量无关,只与最高层级联盟成员数量有关,缩减了过滤器需求量.很明显,在联盟规模一定的条件下,TMStub 成员数量越多,最高层级联盟成员数量就会越少,它的性能优势就越明显.但是,受限于自身策略、网络结构和经济、政治、军事利益等因素,某些 Transit 域可能无法或不愿意开启过滤功能,从而影响联盟中 TMStub 成员数量.本文将开启过滤功能的 Transit 域,记为 Mtransit.不过,即使只有少量 TMStub 成员,与 FAA 相比,HAA 在开销方面也具明显优势.这就意味着:我们可以在维持 HAA 优势的同时,采用松耦合的方式来生成 TMStub,

并不要求所有满足定义 6 的局部网络都建立 TMStub,从而能够更灵活的构建反匿名联盟.

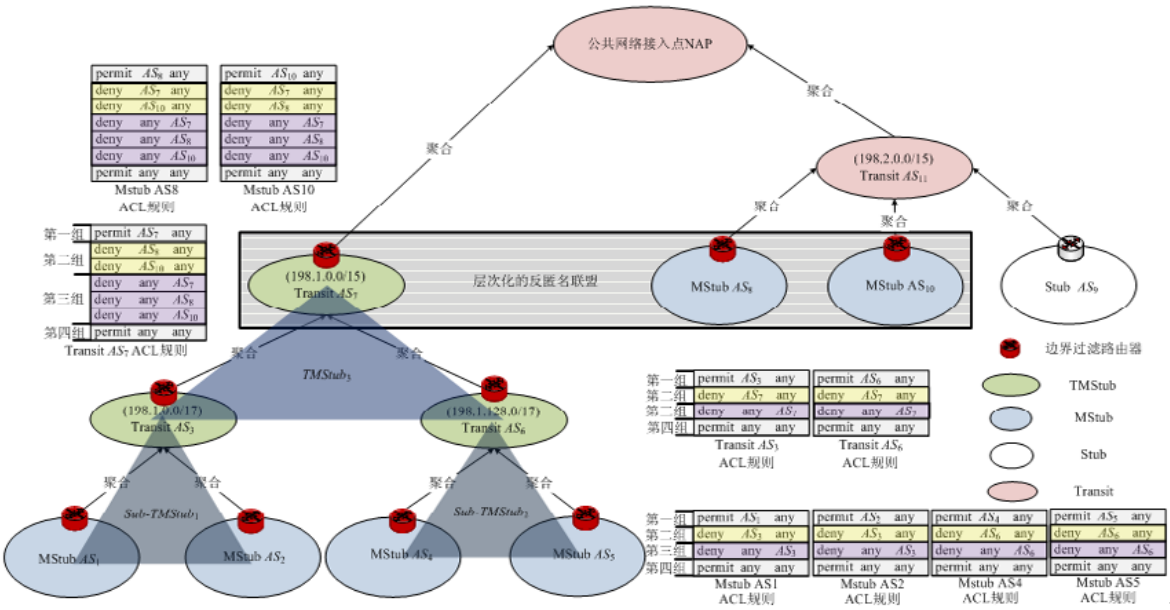

Fig.3 An example of HAA

图 3 层次化的反匿名联盟举例

定义 6. 逻辑可信 Stub 域定义为二元组 $TMStub=\{u_{Trst},IS_{Mstub}\}$,其中,$u_{Trst}$ 表示 TMStub 的边界 Transit 域,IS_{Mstub} 表示 TMStub 的内部 Mstub 域集合, $a(TMStub)=\bigcup_{u\in TMStub} a(u)$.∀数据流 $F=(s,I,d)$,已知 $a(F.s)\subset a(TMStub)$,F 满足以下条件:(1) 若 $a(F.d)\subset a(TMStub)$,则 $F.i=a(F.s)$;(2) 若 $F.d\not\subset a(TMStub)$,则 $F.i=a(F.s)$或 $F.i\in a(TMStub)^{-}$.

性质 1. 给定 TMStub,∀数据流 $F=(s,I,d)$,$a(F.s)\subset a(TMStub)$且 $a(F.d)\not\subset a(TMStub)$,如果 $F.i\subset a(TMStub)$,就有 $F.i=a(F.s)$.

证明:运用定义 6 即可证明. □

定义 7. 层次化的反匿名联盟定义为 $HAA=\{MStub_1,\ldots,MStub_n,TMStub_1,\ldots,TMStub_m\}$,其中,MStub 是物理 MStub 域,TMStub 是最高层的逻辑可信 Stub 域.给定 $u\in HAA$,∀匿名流 $F=(s,i,d)$,已知 $F.s=u$,F 需满足以下条件:

(1) 若$\exists v\in HAA$, s.t. $a(F.d)\subset a(F.v)$,那么 $F.i=a(u)$;

(2) 若$\forall v\in HAA$ 都不存在 $a(F.d)\subset a(F.v)$,那么 $F.i=\Omega_{stub}-\bigcup_{e\in HAA} a(e)$.

定理 1. 层次化的反匿名联盟 HAA 与扁平化的反匿名联盟 FAA 在过滤效果和激励效果方面理论上相同.

证明:首先,把 FAA 的 MStub 成员按照 HAA 中 TMStub 的组成结构划分为若干组.以图 3 的 $HAA=\{TMStub_3,AS_8,AS_{10}\}$为例,已知 $T_1=\{AS_1,AS_2,AS_4,AS_5\}$隶属于 $TMStub_3$,因此,图 2 中 FAA 可划分为两组,即 $FAA=T_1\cup T_2$,其中,$T_2=\{AS_8,AS_{10}\}$.然后,为了证明 FAA 和 HAA 的过滤效果相同,将 FAA 的匿名包过滤场景分为三类:组内成员互相发送匿名包、组内向组间发送匿名包、组间向组内发送匿名包,从而只需证明 3 个场景分别相同即可.

- 场景 1:∀匿名流 $F=(s,i,d)$,如果 $F.s\in T_1$ 且 $F.d\in T_1$,根据定义 5 可知 $F.i=a(F.s)$.已知$\forall u\in T_1$ 都有 $a(u)\subset a(TMStub_3)$,根据定义 6 可知 $F.i=a(F.s)$.这就意味着 FAA 和 HAA 在场景 1 下过滤效果相同;
- 场景 2:∀匿名流 $F=(s,i,d)$,如果 $F.s\in T_1$ 且 $F.d\in T_2$,根据定义 5 可知 $F.i=a(F.s)$.已知 $F.d\in HAA$ 且 $a(F.d)\not\subset a(TMStub_3)$,根据定义 7,可知 $F.i\subset a(TMStub_3)$.进一步通过性质 1 可得 $F.i=a(F.s)$.因此,FAA 和 HAA 在场景 2 下过滤效果相同;
- 场景 3:∀匿名流 $F=(s,i,d)$,如果 $F.s\in T_2$ 且 $F.d\in T_1$,根据定义 5 可知 $F.i=a(F.s)$.已知 $F.d\in HAA$ 且 $a(F.d)\subset$

$a(TMStub_3)$,根据定义 7,可知 $F.i=a(F.s)$.FAA 和 HAA 在场景 3 下过滤效果也相同.

最后,为了证明 FAA 和 HAA 的激励效果相同,只需说明它们都不存在搭便车自治域.也就是说:∀匿名流 $F=(s,i,d)$,如果 $F.d\notin AA$,那么不存在 $F.s\in AA$, s.t. $F.i=a(F.s)$.为此,将 FAA 的匿名包过滤场景分为两类:组内成员向非联盟成员发送匿名流和组外成员向非联盟成员发送匿名流.

场景 1:∀匿名流 $F=(s,i,d)$,如果 $F.d\notin FAA$ 且 $F.s\in T_1$,根据定义 5,可知 $F.i=\Omega_{stub}-\bigcup_{e\in HAA}a(e)$.如果$\exists u\in \mathrm{TMStub}_3$, s.t. $a(F.s)\subset a(u)$,且$\forall v\in \mathrm{TMStub}_3$, st. a_3, s.t. $a(F.d)\not\subset a(v)$,根据定义 7,可知 $F.i=\Omega_{stub}-\bigcup_{e\in HAA}a(e)$.因此,FAA 和 HAA 在场景 1 下不存在搭便车自治域;

场景 2 与场景 1 相同,也不存在搭便车自治域.

证毕. □

2.2 层次化反匿名联盟的构建方法

传统的 E-Filtering 通常是被各个自治域独立部署,然而层次化反匿名联盟 HAA 却要求部署 E-Filtering 的自治域之间既能够及时发现彼此,也能够互相交换网络前缀,因此它的构建必然依赖于一种分布式协同控制系统.鉴于 HAA 立足于实际网络体系结构,与网络控制系统相似,该协同控制系统应具备开放性、简单性、可扩展性、可靠性等特征.围绕实现这些特征,接下来我们主要阐述 HAA 分布式协同控制系统的结构模型、协同流程、安全策略、可靠性策略、过滤规则优化等.

2.2.1 系统结构模型

一般来说,按照分布式协同系统中各部分的位置、角色以及它们之间的关系,其结构模型可分为两类:对等结构和客户/服务器结构.若使用前者来构建 HAA,为了完成 TMStub 成员判别和对等过滤,每个成员域必须通过建立一对多的通信关系来收集其他成员信息,进而获取联盟成员列表.除此之外,鉴于联盟与成员之间宽松的绑定关系,每个成员域还需具备邻接成员资格检测功能,以便随时感知新成员的加入和旧成员的退出,进而更新成员列表并将其通知至整个联盟,保证元数据的一致性.简言之,这既会加大设计的复杂性,又会产生巨大的通信开销,进而影响系统的可扩展性.然而,若采用后者来构建 HAA,联盟成员列表的维护工作(包括成员的动态加入或退出)将全部由服务器来完成,而成员域作为客户端,只需提交加入或退出请求即可,成员域之间无需建立任何关系,这既能简化成员注册/退出流程,实现系统高效扩展,又能省去邻接成员资格检测,降低系统通信开销.不过,客户/服务器结构将所有操作都集中于服务器端,从而使它极易成为整个系统的瓶颈.为了解决该问题,一种直观的方法是利用 TMStub 分层嵌套结构的特点,将中心服务器的部分管理功能逐层下放到 MTransit 域中,由它们负责 TMStub 成员的维护以及其内部 ACL 规则参数的传递,而服务器只承担最高层级成员的管理工作.在这种可分层的客户/服务器结构模型中,下层自治域若提出注册/退出请求,正常情况下需要先与上层 MTransit 域建立通信连接,逐层向上转发请求信息,以便各层 MTransit 域随时判断下层成员是否已满足 TMStub 生成条件,直至中心服务器.此外,为了降低系统运行条件,使它更符合真实的网络环境,我们不能要求每个 Transit 域都成为 MTransit.在这种特殊情况下,因上层不存在 MTransit,下层自治域只能将注册/退出请求直接发送给中心服务器.基于此,上层 MTransit 的不确定性导致下层自治域必须具备关于它的检测功能,然后依据检测结果来选定请求消息的发送对象,进而再次引发因系统设计难度大和通信开销高的问题.因此,利用该模型来构建 HAA 并不合适.另一种方法是采用多副本存储技术,由多个配置完全一致、功能地位对等、所处地区不同的副本服务器共同组成中心服务器,而由地位完全相等的 MTransit 和 MStub 成员域组成客户端.由于该方法的上下层级的成员域之间无需执行主从式通信,因此不会引发可扩展性差的问题.综上所述,本文最终选定基于多副本均衡负载的客户/服务器系统结构模型来构建 HAA.

如图 4 所示,HAA 构建系统需要由联盟成员管理模块(alliance management module,简称 AMM)、Transit 对等过滤模块(mef-transit module,简称 MTM)、MStub 对等过滤模块(mef-stub module,简称 MSM)、ACL 规则配置模块(ACL rule configuration module,简称 RCM)协同工作来完成联盟成员列表的管理、ACL 规则的生成和匿名包的过滤.为了降低部署成本,本文借鉴了第 1.2 节提到的文献[9]中有关控制层面的部署方式,将上述模块依

次部署于联盟注册服务器(alliance registration,简称 AR)、网络管理服务器(network management,简称 NM)、边界过滤路由器,其中,AR 是由 Internet 号码分配局(Internet assigned number authority,简称 IANA)或地区注册处(regional Internet registries,简称 RIRs)负责维护的新增互联网实体设备,而 NM 是由网络服务提供商负责维护的已存在的网络实体设备.所不同的是:在 EAGLE 中,上述模块分别被赋予新的职能.MSM 作为核心功能模块,主要任务有:(1) 向 AMM 提交成员注册/退出请求,其内容须包含成员的 AS 编号和 AS 网络前缀等;(2) 接收 AMM 向它传递的 ACL 规则参数,以此为依据,结合 RCM 提供的过滤器数量,生成 ACL 规则;(3) 向同层 RCM 发起 ACL 规则配置请求.MTM 作为新添加模块,绝大部分任务都与 MSM 重合,唯一的区别是:MTM 向 AMM 提交的请求信息中,除了本自治域的 AS 编号和 AS 网络前缀,还应包含它所覆盖下层所有自治域的 AS 编号和 AS 前缀.RCM 作为过滤执行模块,一边向同层 MTM 或 MSM 上传边界过滤路由器所拥有的过滤器数量,一边接收和配置由它们下发的过滤规则.与基于路由的反匿名方法不同[14],本文的过滤规则表是基于 ACL 平台的,而不是路由转发表,因此它的下发过程不依赖于 BGP 扩展协议,而是 SSH,TELNET 等远程通信协议.鉴于联盟成员的动态性,本文的过滤规则表无法一次性完全生成,只能增量装载.为了降低开销、提升效率,本文建议通过组合当前路由器操作系统(例如 Cisco IOS v8.3)已支持的增删改操作来完成表增量,而不是先删除后重建.AMM 作为信息汇聚模块,主要任务有:

(1) 利用已有的拓扑测量方法和自治域商业关系推测方法来获取 AS 级网络结构数据,进而建立第 1.1 节提到的层次网络结构模型.在该模型中,Transit 节点配置两个属性:过滤状态位和 TMStub 状态位,前者记为 FS,用于声明相关自治域是否已注册为成员域;后者记为 TS,用于声明相关 Transit 域管辖区域能否构成 TMStub.而 Stub 节点只配置过滤状态位 FS;

(2) 接收和保存 MTransit 和 MStub 成员域的注册请求信息,修订其在层次网络结构模型中节点状态信息;

(3) 遍历拓扑结构模型,依据定义 6 来判别部分注册成员是否能生成 TMStub,在修改节点状态位的同时更新联盟成员列表(alliance member list,简称 M-List),以此为基础,结合第 2.1 节的过滤场景,给每个成员下发相应的 ACL 规则参数.

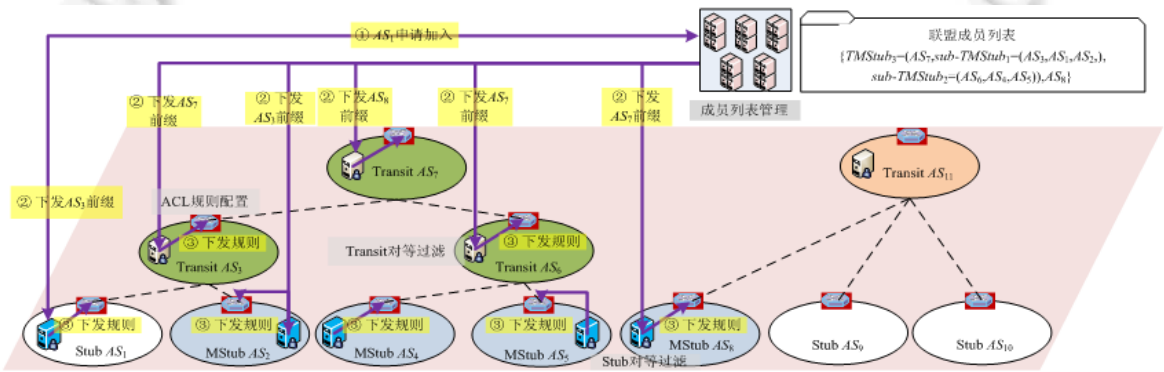

Fig.4 System structure overview for HAA (HAA from Fig.3)

图 4 HAA 构建系统的结构模型(联盟结构源于图 3)

2.2.2 协同流程

在层次化反匿名联盟中,数据包的过滤动作受联盟成员列表的控制,而成员列表又因 MTransit 和 MStub 域的加入或退出发生变化,原因是它们可能引发 TMStub 的生灭,进而造成列表成员的聚合和裂变.因此,MTransit 和 MStub 域的加入或退出是触发系统运行的主要事件.本文以图 4 为例来分别阐述当上述事件发生后,各模块之间是如何协同、完成列表更新的.假设 Stub AS_2,AS_4,AS_5,AS_7 都已注册为成员域,AMM 的网络拓扑结构模型如图 1 所示,成员列表 $M\text{-}List=\{AS_2,AS_4,AS_5,AS_7\}$.

(1) 非成员域若想加入联盟,先得向 AMM 提出注册申请.

AMM 在收到申请后,首先在拓扑结构模型中查找注册域对应节点,修改其过滤状态位,并将其孩子节点与申请消息中下层自治域逐一匹配,纠正拓扑模型的节点漏报;然后判定 TMStub 域的生成,依据判定结果,修改相关边界域的TMStub状态位,同时更新成员列表,最后向联盟成员(包括 AS_i)下发新的ACL规则参数;成员域收到参数后,依据过滤器数据,优化过滤规则,并将其下发到 RCM;RCM 通过配置规则,完成匿名包过滤.整个过程需AMM 完成定位、判定、更新这 3 个任务,接下来主要说明它们的实现方式.

a. 鉴于拓扑结构模型存在逐层汇聚的特点,本文将它定义为一种可用于快速检索的多叉树结构——前缀树(retrieval tree,简称 Trie),通过它的查找方法即能快速定位自治域;

b. 不同类型申请域的判定方法也不同.若 Transit 域 AS_i 申请加入,首先查验以 AS_i 为首的局部网络是否生成TMStub,然后采用向上就近查验原则,逐层判定高层级的TMStub.例如:若 AS_3 希望成为 MTransit,因为在其管辖的局部网络中 AS_1 尚未注册,所以不满足 TMStub 生成条件,判定结果为 MTransit.AS_6 在 AS_3 加盟后也提出注册请求,因其管辖的 AS_4 和 AS_5 都已加入联盟,生成 $sub\text{-}TMStub_2$.然后查验以 AS_6 为首、覆盖 $sub\text{-}TMStub_2$ 的局部网络,条件不再匹配,最终判定结果为 TMStub.若 Stub 域 AS_j 申请加入,直接采用向上就近查验原则即可.例如:继 AS_3 和 AS_6 后,AS_8 提出申请,因以 AS_{11} 为首、覆盖 AS_8 的局部网络存在非成员域,TMStub 判定结果为 MStub.此后,AS_1 选择加入,生成 $sub\text{-}TMStub_1$,然后查证以 AS_7 为首、覆盖 $sub\text{-}TMStub_1$ 的局部网络,生成 $TMStub_3$,最终判定结果为 TMStub;

c. 依据 TMStub 判定结果,选择成员列表更新方法.若结果为 Transit 域,无需更新.例如,加入 AS_3 既没有生成新 TMStub,也不影响对等过滤,因此 $M\text{-}List=\{AS_2,AS_4,AS_5,AS_7\}$.若结果为 TMStub 域,需要将成员列表的部分成员聚合为该自治域.例如:加入 AS_6 生成 $sub\text{-}TMStub_2$,先将该 TMStub 添加到成员列表,后将它覆盖的成员前缀 AS_4 和 AS_5 删除,$M\text{-}List=\{AS_2,AS_7,sub\text{-}TMStub_2\}$.若结果为 Stub 域,直接将它添加到成员列表.例如,加入 AS_8 虽然没有生成新 TMStub,但是为了不影响对等过滤效果,$M\text{-}List$ 更新为 $\{AS_2,AS_7,AS_8,sub\text{-}TMStub_2\}$.

(2) 成员域若想退出联盟,先得向 AMM 提出退出请求.

AMM 在收到请求后,首先在拓扑结构模型中修改相关节点的过滤状态位,然后将该节点及其祖先节点的TMStub 状态位都改为 False,最后更新成员列表,向联盟成员下发规则参数;成员域和 RCM 的处理过程与步骤(1)相同.值得说明的是,列表更新方法取决于退出成员类型及其前缀与成员列表的关系.若 MTransit AS_i 选择退出且它的前缀等于或包含于最高层级的 TMStub 成员,首先分裂与 AS_i 相关的所有 TMStub,然后将剩余 TMStub 和 MStub 添加到成员列表,例如:当 $M\text{-}List=\{sub\text{-}TMStub_3,AS_8\}$运行一段时间后,MTransit AS_3 因经济或政治原因选择退出,希望重新变为 Transit 域,与 AS_3 相关的 TMStub 域(包括 $sub\text{-}TMStub_1$ 和 $sub\text{-}TMStub_3$)全部发生裂变,最终剩余 MStub AS_1,AS_2 和 $sub\text{-}TMStub_2$,从而将 $M\text{-}List$ 更新为$\{AS_1,AS_2,sub\text{-}TMStub_2,AS_8\}$;若 AS_i 的前缀与任何成员不发生关系,无需更新,例如:继 AS_3 后,AS_7 也选择退出,它就与 $M\text{-}List$ 成员无交集;若 MStub AS_j 选择退出且它的前缀等于最高层级成员,就直接将它从列表中删除,例如:AS_8 退出不会引发 TMStub 分类,只需更新 $M\text{-}List=\{AS_1,AS_2,sub\text{-}TMStub_2\}$;若 AS_j 前缀包含于某 TMStub 成员,其处理过程与 MTransit 退出相同,例如:AS_4 退出造成 $sub\text{-}TMStub_2$ 的裂变,使 $M\text{-}List$ 更新为$\{AS_1,AS_2,AS_5\}$.

2.2.3 安全和可靠性策略

就安全性而言,HAA 构建系统的开放性允许任何陌生自治域都能向 AMM 提出成员注册、退出请求,这为匿名、中间人等恶意攻击的发起提供了可能,究其原因在于:(1) AMM 无法通过成员域的网络前缀来识别其身份;(2) 成员域 MTM/MSM 无法确认 AMM 发送的消息中 ACL 规则参数是否被篡改.基于此,系统模块之间的通信只有嵌入身份认证和消息验证才能预防上述攻击.为此,本文以节约部署成本、保证执行效率为出发点,先使用资源公共密钥基础架构(resource public key infrastructure,简称 RPKI)来保障 MTM/MSM→AMM 的通信安全,后使用 RSA 数字签名来完成 AMM→MTM/MSM 的安全通信,最终实现它们之间安全的双向通信.

鉴于核心管理模块 AMM 的负载较重,本文采用多副本存储技术来构建 AMM,这虽然提升了 HAA 系统的可靠性和可用性,但是也带来管理难题,包括如何放置副本、如何均衡副本负载、如何保持副本之间的成员数

据一致性、如何处理副本故障等.本文以高效、实用为出发点,分别选用如下技术来解决上述问题.

- 为了降低通信时延、缩短响应时间,引入数据分区技术,将 AMM 按照 RIRs 的组织方式划分为 5 份,每个注册地区至少放置一个 AMM 副本,除了少量的跨分区通信外,大部分的分区都可独自地处理服务请求;
- 若所属注册地区存在多个副本,客户端通过循环域名服务技术(round-robin DNS,简称 RR)来选定与它通信的副本服务器,从而保证各副本之间的负载平衡;
- 通过对 AMM 业务特点进行分析,选定通信开销较小的最终一致性模型来完成副本数据之间的同步,通过牺牲一致性来换取高可靠性和可用性;
- 采用自适应检查点机制和 Merkle 哈希树来解决临时故障和永久性故障.

2.2.4 过滤规则优化

在 HAA 构建初期,联盟中 MStub 成员类型较多而 TMStub 成员类型较少,进而造成绝大部分成员位于联盟最高层.根据过滤场景 1 可知,最高层成员之间需要维持双向过滤关系,因此这会产生过多过滤规则,从而使边界过滤路由器的过滤器资源发生短缺.为此,MTM 和 MSM 模块如何在不损害激励机制的前提下优化过滤规则、降低过滤规则对过滤器的需求,成为决定 HAA 激励性能的关键问题.

2.2.4.1 过滤规则优化问题形式化描述

如图 3 所示,按照功能不同,位于联盟最高层成员的 BFR 所配置的过滤规则可划分为 4 组,其中:第 1 组和第 4 组分别只含一条规则,而第 2 组和第 3 组所含规则较复杂.因此,BFR 通常能够容纳第 1 组、第 4 组规则,却无法容纳其余两组.基于此,本文主要集中于优化第 2 组、第 3 组规则.鉴于这两组规则都具备一维特性(其规则参数仅局限于源地址或目的地址且二者只能取其一),过滤规则优化可归为同一前缀压缩问题(same prefixes compression,简称 SPC).给定一组去本地化且彼此不重叠的成员列表 *M-List*、一组彼此不重叠的非成员列表 *W-List*、过滤器投入量 $R_{\max}$ 以及一组归一化的地址权重集 *W*.为了覆盖所有 *M-List* 成员,同时最小化搭便车率(free riding ratio,简称 FRR),SPC 可被形式化为如下方程:

$$minimize\ \ FRR=\sum\nolimits_{p\in L}[x_p\cdot(g=\sum\nolimits_{ip\in p\wedge w_{ip}\in W}w_{ip})] \tag{1}$$

$$\text{s.t. } \sum x_p\leqslant R_{\max} \tag{2}$$

$$\sum\nolimits_{p\in L\wedge ip\in p}x_p=1,\forall ip\in_{sub} M\text{-}List \tag{3}$$

$$x_p\in\{0,1\},\forall p\subseteq\Omega_p \tag{4}$$

其中,公式(1)表明优化目标,g 是 *FRR* 的决定因子;公式(2)表明过滤器投入数量不高于 $R_{\max}$;公式(3)表明每个成员前缀有且只能被过滤一次,否则会浪费过滤器;公式(4)表明决策变量的取值范围.

定义 8. 成员列表 $M\text{-}List=\{P_{AS_1},P_{AS_2},...,P_{AS_n}\}$,其中,$P_{AS_i}$ 表示 HAA 中最高层成员 AS_i 的网络前缀(若是 TMStub,则 AS_i 就是 TMStub 边界域).*M-List* 的成员具备两个性质.

(1) $\forall P_{AS_i},P_{AS_j}\in M\text{-}List,P_{AS_i}\cap P_{AS_j}=\varnothing$;

(2) $\forall P_{AS_i},P_{AS_j}\in M\text{-}List,\exists P\in\Omega_p$, s.t. $P_{AS_i}\subset P$且$P_{AS_j}\subset P$.

其中,Ω_p 表示 IP 前缀全集.

定义 9. 亚属于$\in_{sub}$:$ip\in_{sub}List$,当且仅当 $\exists P_{AS_i}\in List$, s.t., $ip\in P_{AS_i}$,其中,*List* 表示 *M-List* 或 *W-List*,*ip* 表示 IP 地址.

定义 10. 非成员列表 $W\text{-}List=\{P_{AS_1},P_{AS_2},...,P_{AS_m}\}$,其中,$P_{AS_i}$ 表示未加入联盟的 Stub AS_i 网络前缀.*W-List* 具备以下性质:

(1) $\forall P_{AS_i},P_{AS_j}\in W\text{-}List,P_{AS_i}\cap P_{AS_j}=\varnothing$;

(2) $\forall ip\in_{sub} M\text{-}List$,不存在$P_{AS_k}\in W\text{-}List$, s.t., $ip\in P_{AS_k}$.

定义 11. 决策变量 $x_p\in\{0,1\}$,当 $p\in L$ 时,x_p=1;当 $P\notin L$ 时,x_p=0.其中,*L* 表示规则优化算法输出的网络前缀集.$R_{\max}$ 是指过滤器最大投入数,也就是说$|L|\leqslant R_{\max}$.

定义 12. 地址权重 $w_{ip}\in[0,1]$,表示免费保护地址 ip 可能带来多大程度的不良影响.若 $ip\in_{sub}M\text{-}List, w_{ip}=0$;若 $ip\in_{sub}W\text{-}List, w_{ip}>0$.

定义 13. 地址权重集 $W=\{w_{ip}\mid_{ip\in sub\Omega_p}\}$,为了描述简单,对权重进行归一化处理,使得 $\sum_{ip\in sub\Omega_p}(w_{ip})=1$.给定 W,$\forall$网络前缀 $P\in L$,它的权重 $g_p=\sum_{ip\in p\sqcap w_{ip}\in W}(w_{ip})$.

定义 14. 搭便车率 $FRR=\sum_{p\in L}(g_p\cdot x_p)$ 是指得到免费保护的网络前缀的总权重.

2.2.4.2 过滤规则优化算法

文献[13]指出,SPC 问题可归为多维背包问题(multidimensional knapsack problem,简称 dKP).众所周知,dKP 属于 NP-hard 范畴,若使用常规解法,例如分支界限法和割平面法,虽然能够求得精确解,但是运行时间随着反匿名联盟规模的增大会呈指数增长,从而使得各成员域的 MTM 或 MSM 模块无法快速求得最优解.其次,联盟规模的逐渐扩张要求过滤规则优化算法必须具备增量更新的特点,否则,若新旧过滤规则合并后再优化,那么这一方面会因数据集过大而消耗太多的时间和存储空间,另一方面则会因全局优化使得原有过滤规则也需重新配置,产生额外处理开销.Soldo[13]曾经提出一种基于最长公共前缀树(longest common prefix tree,简称 LCP)、可增量更新的动态规划算法,用来解决 IP 地址优化问题.经过碎片化处理,虽然能够将本文所关注的前缀优化也转化为上述的 IP 地址优化,但是这会造成 IP 数量激增,进而带来庞大的计算开销.为此,Liu[9]提出了基于扩展 LCP 树(extended LCP,简称 ELCP)的动态规划算法,它在保证解质量的同时提高了求解速度.然而该算法存在两点不足.

(1) 采用排他法来建立非成员列表 *W-List*,即$\forall ip, ip\in M\text{-}List\Leftrightarrow ip\notin W\text{-}List$.由于 *W-List* 是构建 ELCP 树的重要数据源,这既会生成大量碎片 IP,进而造成 ELCP 树的规模过大,产生较大内存开销,又会因多余的 *W-List* 成员而影响 ELCP 树非叶子节点中误报率属性的计算结果,进而降低了算法优化精度.原因如下:假设两个自治域执行聚合操作,判断其结果是否引入误报的依据应该在于它有没有夹杂其他真实自治域的网络前缀,而与这两个自治域的网络前缀是否连续没有直接关系.也就是说:即使它们不连续,只要聚合结果不包含真实自治域,它也就不会引发误报.然而在上述方法中,鉴于 *W-List* 包含大量非真实自治域 IP,这会造成 ELCP 树包含许多无用白色叶子节点,进而影响非叶子节点中误报率属性的计算,最终误导算法的优化方向,降低它的求解精度;

(2) 不再支持增量更新,降低了算法重新求解的效率.

针对上述两个问题,本文提出了一种可增量的高效过滤规则优化算法.

从聚合角度来看,SPC 问题就是在指定数量的条件下搜索一组误报率最小的 *M-List* 成员聚合前缀.基于此,可增量的过滤规则优化过程可划分为 3 步:第 1 步,以最小化聚合误报为目标,推测 *M-List* 成员最优聚合次序,提取它们的聚合前缀,组成最优前缀集 I-Set;第 2 步,从 I-Set 中挑选数量不超过$|R_{\max}|$个、彼此不相交、累积聚合误报最小的前缀,组成最终结果 F-Set;第 3 步是在成员加入或退出的时候,及时更新 I-Set,以便重新计算 F-Set.每一步的实现方式如下所述.

(1) 给定 $M\text{-}List=\{P_{AS_1},P_{AS_2},...,P_{AS_n}\}$,计算所有成员的聚合前缀 $P_{AS_1}\infty P_{AS_2}\infty...\infty P_{AS_n}$.

根据定义 15 可知:聚合操作属于二元运算,且满足交换律和结合律,因此任何加括号的方法都会得到相同的计算结果.例如 $M\text{-}List=\{P_{AS_1},P_{AS_2},P_{AS_3}\}$,则共有 3 种加括号方式:$((P_{AS_1}\infty P_{AS_2})\infty P_{AS_3}),(P_{AS_1}\infty(P_{AS_2}\infty P_{AS_3})),(P_{AS_1}\infty(P_{AS_1}\infty P_{AS_3}))$,它们的计算结果完全相等.然而不同的加括号方法通常会产生不同的累积聚合误报,例如:根据定义 16,$\rho(((P_{AS_1},P_{AS_2}),P_{AS_3}))=\rho(P_{AS_1},P_{AS_2})+\rho(P_{AS_{1,}AS_2},P_{AS_3}),\rho((P_{AS_1},(P_{AS_2},P_{AS_3})))=\rho(P_{AS_1},P_{AS_{2,}AS_3})+\rho(P_{AS_2},P_{AS_3})$,$\rho(P_{AS_2},(P_{AS_1},P_{AS_3}))=\rho(P_{AS_2},P_{AS_{1,}AS_3})+\rho(P_{AS_1},P_{AS_3})$,而且 $\rho(P_{AS_{1,}AS_2},P_{AS_3})=\rho(P_{AS_1},P_{AS_{2,}AS_3})=\rho(P_{AS_2},(P_{AS_1},P_{AS_3}))$.此时,如果 $\rho(P_{AS_1},P_{AS_2})>\rho(P_{AS_2},P_{AS_3})>\rho(P_{AS_2},P_{AS_3})$,那么不难推断 $\rho(((P_{AS_1},P_{AS_2}),P_{AS_3}))>\rho((P_{AS_1},(P_{AS_2},P_{AS_3})))>\rho(P_{AS_2},(P_{AS_1},P_{AS_3}))$.因此,如何优化聚合次序(即加括号),使 *M-List* 成员聚合操作产生最小的累积误报率,成为首先需要解决的关键问题.

在提出解决方案之前,本文对聚合优化问题做了如下分析.

- 首先,给定一个规模为 n 的 *M-List*,令 $P(n)$表示可供选择的括号化方案的数量,不难推出 $P(n)=C_n^2\times$

$C_{n-1}^2\times...\times C_2^2$,这意味着括号化方案数量与 n 呈指数关系,因而穷举法的效率会非常差;

- 其次,根据定理 2,聚合优化问题具备最优子结构和重叠子问题等基本要素,因此我们可以利用基于自底向上的动态规划方法来对它求解,即,通过子问题的最优解来逐层向上递归构造出原问题最优解.整个过程可看做是在构建一颗二叉树:首先,从 *M-List* 中挑选聚合误报最小的两个成员作为参数;然后建立一颗以参数为叶子、聚合前缀为根的二叉子树;最后,将该子树插入 *M-List*,同时删除 *M-List* 中与子树相关的成员.逐层向上重复上述动作,最终生成一颗前缀聚合二叉树,树中所有非叶子节点就可组成最优聚合前缀集 I-Set.

通过分析,动态规划方法供选择的括号化方案 $P(n)=C_n^2\times C_{n-1}^2\times...\times C_2^2$. 很明显,与穷举法相比,它能将运行时间降低为多项式阶.但是当 n 较大时,其效率依然较差.主要原因在于子问题的求解过程采用了暴力搜索方法,即:只有将所有二元组合的聚合误报都进行比较,才能计算出误报最小的组合.然后,令 *CM-List* 表示 *M-List* 中每个成员在最长公共前缀取最大值时所对应的二元组合,*CM-List*=$\{\langle P_i,P_j\rangle|\forall P_k\in$*M-List*, 都有 $LCP(P_i,P_j)\geqslant LCP(P_i,P_k)$,其中,$P_i,P_j\in$*M-List*,*LCP* 表示最长公共前缀$\}$.根据性质 2,误报最小的二元组合必然包含在 *CM-List* 中,这就意味着利用该信息能够将子问题求解范围缩小到 *CM-List*.基于此,本文提出一种网络前缀序列空间,通过定义一组面向网络前缀的全序关系操作集,如定义 17 所示,使得 *M-List* 成员按照彼此所拥有的最长公共前缀长度进行排列(即 *LCP* 越长,成员之间距离越近;反之越远),进而获得 *M-List* 成员序列.以该序列集为基础,通过顺序提取成员对就可计算出 *CM-List*.然而,由于 *M-List* 成员在向上递归过程中会不断更新,因此 *CM-List* 也需要被反复计算,这又会造成较大的时间开销.最后,本文借鉴时空权衡理论,提出了一种面向网络前缀的二叉线索树 BT,如定义 18 所示.它利用 IP 地址空间结构稳定且可分层的特点,将聚合操作与网络前缀二进制代码巧妙结合,使树节点既能保存前一阶段 *M-List* 成员前缀序列,又能利用它们对当前 *M-List* 成员前缀进行快速排序,极大地减少了重复排序的次数,节省运行时间.此外,为了快速、准确地计算出各子问题的聚合误报,可以将 *W-List* 成员逐一插入到二叉树中,同时把该成员所对应的地址权重添加到树中已存在的相关节点,各节点通过累加地址权重就能得到它所对应的聚合误报.

基于上述分析,因为二叉线索树能够标识出哪些节点对应最优聚合前缀,所以通过遍历就能获取最优聚合前缀集 I-Set.此外,由于 *M-List* 和 *W-List* 成员因联盟规模变化而不断更新(加入或退出),因此树的建立过程除了在 BT 中标识最优聚合前缀和计算聚合误报两个阶段,还应包括更新阶段.更新操作由成员变化类型而决定.例如:当 *M-List* 加入新成员 m 时,如果 m 之前属于 *W-list*,说明树中已包含 m;如果 m 不属于 *W-List*,先遍历已存在节点,不做任何操作,后插入新节点.当 *W-List* 加入新成员 n 时,操作过程与第二阶段相同.退出 *M-List* 相当于是向 *W-List* 添加新成员,因此该操作与上一过程相同.

定理 2. 聚合优化问题能够采用动态规划方法来求解.

证明:众所周知,只有具备最优子结构和子问题重叠特征的最优化问题才适合使用动态规划方法来求解.基于此,我们首先发掘聚合优化问题的最优子结构,该证明过程通常遵循如下模式.

a. 如果对 *M-List* 所有成员执行聚合操作,那么必先将 *M-List* 划分为两个子集 sub_1 和 sub_2,聚合误报 ρ(*M-List*)=$\rho(sub_1)+\rho(sub_2)+\Delta_{M\text{-}List}$,其中,$\Delta$表示包含于 *M-List* 的聚合前缀、与 sub_1 和 sub_2 无交集的聚合误报;

b. 假设 sub_1 和 sub_2 是 *M-List* 的最优分割子集,那么对子集 sub_1 进行聚合优化,直接采用独立求解该子集时所得的最优方案即可.这样做的原因可采用“剪切-粘贴”技术证明:如果不采用独立求解 sub_1 所得的最优分割方案来对它进行聚合操作,那么可以将此最优方案带入 *M-List* 的最优方案中,代替原来对子集 sub_1 进行分割的方案,显然,这样得到的方案比 *M-List* 原来最优方案的误报更低,这与之前假设相矛盾.对子集 sub_2 来说,可得到相似的结论:在原问题 *M-List* 的最优分割方案中,对子集 sub_2 进行分割的方法,就是它自身的最优分割方案.然后,证明聚合优化问题的的递归算法会反复地求解相同的子问题,而不是一直生成新的子问题,这也就证明它具备子问题重叠性质.

假设 $m[A_n]$表示在聚合基数为 n 的前缀集 A_n 时所生成的最小误报,根据最优子结构可知 $m[A_n]=m[A_i]+$

$m[A_{n-i}]+\Delta_{A_n}$.考虑到子集 A_i 与 A_{n-i} 彼此关系对等,A_n 的分割方案有 $1/2(C_n^1+C_n^2+\ldots+C_n^{n-1})$ 种.由于最优分割方案必在其中,我们只需检查所有可能情况,找到最优者即可.因此,A_n 最优分割方案的递归求解公式变为

$$m[A_n]=\begin{cases}0, & n=1\\ \min\limits_{1\leqslant i\leqslant n}\{m[A_i]+m[A_{n-i}]+\Delta_{A_n}\}, & n>1\end{cases}.$$

通过观察该函数的递归调用树不难发现:在求解高层的子问题时,底层子问题的解会被反复调用,这也就满足了子问题重叠的特征. □

定义 15. 聚合∞:$P_{i,j}=P_i\infty P_j$,当且仅当$(P_i\subseteq P_{i,j})\wedge(P_j\subseteq P_{i,j})$,且$\forall P'\in\Omega_p$, s.t., $(P_i\subseteq P')\wedge(P_j\subseteq P')$,都有$(P_{i,j}\subseteq P')$.

定义 16. 聚合误报$\rho(P_i,P_j)$,当且仅当 $\rho(P_i,P_j)=\sum_{ip\in P_{i,j}\sqcap w_{ip}\in W}(w_{ip})$,其中 $P_{i,j}=P_i\infty P_j$,也可记为$\rho(P_{i,j})$.

性质 2. 给定 *M-List*=$\{P_1,P_2,\ldots,P_n\}$,如果 $LCP(P_i,P_j)\geqslant LCP(P_i,P_k)$,其中,$P_i,P_j,P_k\in$*M-List*,*LCP* 表示最长公共前缀,那么$\rho(P_i,P_j)\leqslant\rho(P_i,P_k)$.

证明:鉴于最长公共前缀具备网络前缀的性质,已知 $LCP(P_i,P_j)\geqslant LCP(P_i,P_k)$,可推出 $P_{i,j}\subseteq P_{i,k}$,其中,$P_{i,j}=P_i\infty P_j$,$P_{i,k}=P_i\infty P_k$,根据定义 16,就可得到$\rho(P_i,P_j)\leqslant\rho(P_i,P_k)$. □

定义 17. 网络前缀序列空间被定义为一种三元组 $H=(I,S,P)$,其中,

- 前缀变量集 I:$I=\{P_1,P_2,\ldots,P_m\}$为所涉及网络前缀的定义范围;
- 前缀序列集 S: $S=(P_{x_1},P_{x_2},\ldots,P_{x_n})$,其中, P_{x_i} (i=1,2,…,n)是定义在 I 上按照从小到大排列的序列,本文用 32 位二进制字符串记法来表示网络前缀 P_{AS_i} 或 x_i,$x_i::=\{\langle$网络号:net-id$\rangle,\langle$零字符:host-id$\rangle\}=\{x_i^1,\ldots,x_i^{32}\}$;
- 操作 P:关于 S 中的前缀变量的操作集.

由于变量集实际上是由二进制字符串构成,因此,传统的绝大部分字符串操作对它仍然是适用的,例如属于(∈)、包含(⊆)、交(∩).然而,对于网络前缀集格来说,仅有这些操作是不够的,还需要扩展它的操作能力.首先,遵循全序关系的自反性、反对称性和传递性原则,给定 $p_i,p_j\in I$,网络前缀之间可能存在的“=”、“>”和“<”关系重新定义如下:(1) 若 $p_i=p_j$,那么$\forall m\in[1,32]$, $x_i^m=x_j^m$;(2) 若 $p_i>p_j$,那么$\exists m\in[1,32]$,使得 $x_i^m>x_j^m$ 且$\forall k<m$, $x_i^k=x_j^k$;(3) 若 $p_i<p_j$,那么$\exists m\in[1,32]$,使得 $x_i^m<x_j^m$ 且$\forall k<m$, $x_i^k=x_j^k$.

以图 5 为例,已知 *M-List*=$\{AS_1,AS_4,AS_8,AS_9\}$,*W-List*=$\{AS_2,AS_5,AS_{10}\}$和 $W=\{W_{AS_2}=0.3,W_{AS_5}=0.3,W_{AS_{10}}=0.4\}$ 全部源自图 1.树的构建分为两个阶段:*M-List* 成员插入和 *W-List* 成员插入.每个节点都有两个属性:最优聚合前缀和误报量.前一阶段负责在尚未插入 *W-List* 的二叉树中标识最优聚合前缀,即将叶子和度为 2 的节点的最优聚合前缀设置为 True,如图 5(1)所示;后一阶段负责在原有树的基础上标识误报量,如图 5(2)所示.

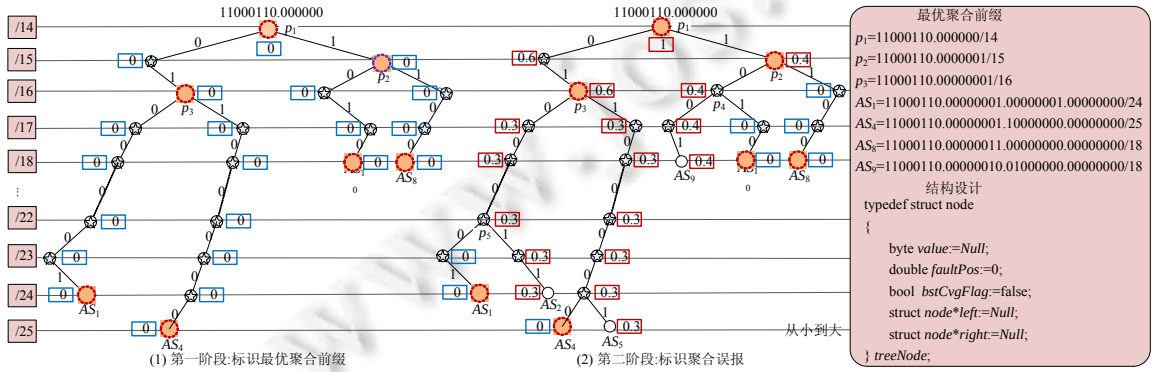

Fig.5 An example of BT

图 5 二叉字典树(BT)举例

定义 18. 二叉线索树(binary trie tree,简称 BT)定义为 *BT*(*H.I*),其中,*H.I*=*M-List*∪*W-List* 表示无序的前缀变量集.BT 具备以下几个特征.

1) 每条从节点到根的路径都对应一个网络前缀的网络号;
2) 非叶子节点的左孩子取值为 0,右孩子取值为 1,这能使得树节点满足序列空间所定义的前缀比较方法从左到右依次增大;
3) 以最优聚合前缀为判别依据,将树节点分为白、黑两种:若某节点与最优聚合前缀(包括 *M-List* 成员前缀和它们之间的最长公共前缀)对应,那么该叶子为黑色;否则,为白色;
4) 每个节点前缀所累计的聚合误报用矩形框来标识到它的附近.

(2) 给定二叉线索树 BT,从中搜索$|R_{\max}|$个、彼此不相交且标识最优聚合前缀的树节点.

通过分析不难发现:不同的节点组合会累积不同的聚合误报.以图 5 为例,假设$|R_{\max}|=3$,满足上述条件的树节点组合有两种:$A=\{AS_1,AS_4,p_2\}$和$B=\{AS_{10},AS_8,p_3\}$,其中,$\rho_A=\rho_{AS_1}+\rho_{AS_2}+\rho_{p_2}=0.4,\rho_B=\rho_{AS_{10}}+\rho_{AS_8}+\rho_{p_3}=0.6$,也就是说 *A* 比 *B* 更高效.基于此,如何获取累积聚合误报最小的节点组合就成为第 2 个需要解决的问题.

最优前缀选择问题可描述如下:给定$|R_{\max}|$个过滤器序列,通过遍历二叉线索树中拥有最优聚合前缀的节点来求取一种分配方案,使得累积聚合误报最小.首先,令 $L_p(k)$表示当使用 *k* 个过滤器来覆盖最优聚合前缀 *p* 时可供选择的分配方案.当 $k=1$ 时,只有一种分配方案;当 $p=leaf$ 时,也只有一种分配方案;当 $k\geqslant 2$ 时,分配方案可描述为两个子方案相乘的形式,而两个子方案的划分点在第 *k* 个过滤器和第 $k+1$ 个过滤器之间,*k* 为 $1,2,\ldots,|R_{\max}|-1$ 中的任意一个值.基于此,可以得到如下递归公式:

$$L_p(n)=\begin{cases}1, & n=1\\ 1, & p=leaf\\ \sum_{k=1}^{n-1}L_{s_l}(k)L_{s_r}(n-k), & n\geqslant 2\end{cases},$$

其中,*sl* 和 *sr* 分别表示在 *p* 的左子树和右子树中离 *p* 最近的最优聚合前缀节点,而 $L_{sl}(k)$和 $L_{sr}(n-k)$则表示关于 *sl* 和 *sr* 的分配方案.该公式与矩阵链相乘问题的递归公式相似,它的序列增长速度接近$\Omega(2^n)$,也就是说分配方案数量与 *n* 呈指数关系.因此,通过暴力搜索所有分配方案来寻求最优组合并不高效.然后,最优前缀选择问题也能够使用基于自底向上的动态规划方法来对它求解,其证明过程与定理 2 相似.

(3) 假设二叉线索树 BT 已成功建立,每当有新成员加入或旧成员退出时,BT 都需要被更新.

例如:自治域 *s* 准备加入联盟,如果 *s* 包含于已选定的过滤器,那么只需将它插入到 BT 即可;反之,除了插入操作,它还需要重新计算最优前缀.不过,它的计算开销与第(2)步相比减少了很多,原因如下:本次只需计算 *s* 的上游节点,其他结果可直接从之前的结果中提取.当 *s* 选择退出联盟时,首先从 BT 中删除一个成员 *s*,然后重新计算位于 *s* 上游的最优前缀.简言之,利用第(2)步已保存的所有中间结果,更新算法只需重新计算与新增节点相关的最优解.

3 性能评价

为了验证提出的 EAGLE 方法,本文进行了理论分析和仿真实验,其中,第 3.1 节将使用理论分析来证明方法的高效性,第 3.2 节则通过基于真实网络拓扑的实验仿真来补充分析结果.

3.1 理论分析

通过与经典的反匿名联盟构建方法(包括 E-Filtering,FAGLE 等)进行比较,本节将使用数学方法来分析 EAGLE 方法的性能,涉及的指标包括部署激励(deployment incentive,简称 Inc)、过滤器需求(filter demanded,简称 FD)、通信开销(system communication overhead,简称 SC)、存储开销(system storage overhead,简称 SS)、规则优化误报(false positive of filter optimization,简称 FP)、规则优化的时间复杂度(time overhead for filter optimization,简称 TF)和规则优化的空间复杂度(space overhead for filter optimization,简称 SF).

3.1.1 部署激励

部署激励指网络服务提供商 ISP 对反匿名方法的部署意愿程度.鉴于 ISP 部署新技术的动力是追求潜在的经济利益,Inc 的大小取决于部署收益和非部署收益两方面:前者是当一个自治域部署了反匿名方法后,其受到

的伪造攻击中匿名报文减少比率的期望增量;后者是就一个尚未部署反匿名方法的自治域而言,其获得匿名报文的减少比率.很明显,Inc 与部署收益成正比,而与非部署收益成反比.基于此,本文采用对偶方式来定义部署收益 $f_1>0$ 和非部署收益 $f_2<0$,而将 Inc 定义为二者之和 f_1+f_2,见定义 5.在基于 E-Filtering 的反匿名网络中,各自治域除了部署收益,还存在非部署收益,因此 $Inc_{E\text{-}Filtering}<f_1$;在基于对等过滤的反匿名网络中,各自治域只具备部署收益而没有非部署收益,因此 $Inc_{MEF}=f_1$.此外,根据定理 2 可知,FAGLE 和 EAGLE 的过滤效果在理想情况下是相同的,进而推出 $Inc_{FAGLE}=Inc_{EAGLE}$.然而在过滤器短缺和规则优化精度等因素影响下,上述两个方法都会产生聚合误报 ρ_{FAGLE} 和 ρ_{EAGLE},从而使得部分自治域再次获得非部署收益.在 *M-List* 和 $F_{\max}$ 给定的条件下,一方面,鉴于聚合操作是在过滤器需求量 FD 超过过滤器供应量的时候发生的,FD 越缺乏,聚合执行越频繁,误报量也越大,可得 ρ 与 FD 成正比关系;另一方面,鉴于聚合结果会受规则优化算法求解质量的影响,优化解的误报越高,聚合误报也越高,可得 ρ 与规则优化误报 FP 也成正比关系.基于此,本文将 ρ 定义为一种以 FD 和 FP 为变量的单调递增函数.

根据第 3.1.2 节和第 3.1.5 节可知,$FD_{FAGLE}\geqslant FD_{EAGLE}$ 和 $FP_{FAGLE}\geqslant FP_{EAGLE}$,因此,$\rho_{FAGLE}(FD_{FAGLE})\geqslant\rho_{EAGLE}(FD_{EAGLE})$,进而得到 $Inc_{E\text{-}Filtering}<<Inc_{FAGLE}\leqslant Inc_{EAGLE}$.

3.1.2 过滤器需求

过滤器需求是指在不执行规则优化的前提下,每个 *AS* 的边界过滤路由器平均配置过滤规则的数量,它会间接影响构建系统的部署激励性.给定一个反匿名网络 $ASN=(G,R_{A\text{-}stub})$,不失一般性,假设每个 *AS* 只有一个网络前缀,通过第 1.3 节和第 1.4 节所述,不难推出:E-Filtering 的过滤器开销 $FD_{E\text{-}Filtering}=2$;FAGLE 的过滤器开销 $FD_{FAGLE}=2|R_{A\text{-}stub}|+1$;在 EAGLE 中,最高层成员分为 TMStub 和 MStub 两类,其中,TMStub 内部成员(包括 MStub 和 MTransit)的过滤器开销 $FD_{EAGLE\text{-}Intra}=4$,最高层成员的过滤器开销 $FD_{EAGLE\text{-}Inter}=2\times$最高层成员数量+1.在 $|R_{A\text{-}stub}|$ 为常数的情况下,如果 $R_{A\text{-}stub}$ 的成员压缩比 $r=$最高层成员数量/所有 MStub 成员数量$|R_{A\text{-}stub}|$,那么 EAGLE 的过滤器需求 FD_{EAGLE} 可定义为一元整数分段函数,如下所示:

$$FD_{EAGLE}(r)=\begin{cases}4, & TMStub\text{内部成员}\\ 2\,|\,R_{A\text{-}Stub}\,|\times r+1, & \text{最高层成员}\end{cases} \tag{5}$$

很明显,r 越大,FD_{EAGLE} 越大,当 $r=1$ 时,$FD_{EAGLE}=FD_M$;r 越小,FD_{EAGLE} 越小.需要说明的是,当 $r=1/|R_{A\text{-}stub}|$ 时,相当于最高层只有一个成员,FD_{EAGLE} 等于 2.因此,$FD_{EAGLE}<<FD_M$.这就意味着:即使只有极少量 MStub 聚集成为 TMStub,与 FAGLE 相比,EAGLE 在过滤器需求方面也会有明显优势.综上所述,$FD_{E\text{-}Filtering}\leqslant FD_{EAGLE}\leqslant FD_{FAGLE}$.

3.1.3 通信开销

通信开销是指因构建反匿名网络而产生的数据通信量,它能反映构建系统对网络带宽的影响.在基于 E-Filtering 的反匿名网络中,每个 EStub 独立过滤匿名流,因此通信开销 $SC_{E\text{-}Filtering}=0$.然而,在基于对等过滤的反匿名网络中,因为 MStub 之间需要通过交换彼此前缀来实现对等过滤,所以 FAGLE 或 EAGLE 会因成员加入和退出而产生通信开销(主要是 AMM 模块与 MStub 和 MTransit 的交互次数).鉴于成员更新不会频繁发生,无论是 FAGLE 还是 EAGLE 构建系统,其通信开销都不会很大.即使这样,若能进一步降低通信开销,也会非常有意义.给定一个反匿名网络 $ASN=(G,R_{A\text{-}stub})$,鉴于 FAGLE 扁平化的体系结构,每次加入或退出产生的通信开销都为 $|R_{A\text{-}stub}|-1$,因此,它的通信开销 $SC_{FAGLE}=n\times(|R_{A\text{-}stub}|-1)$,其中,$n$ 表示成员更新发生次数.鉴于 EAGLE 层次化的体系结构,每次成员加入或退出都有两种情况发生.

(1) 如果生成新的 TMStub 或分裂已有的 TMStub,那么 AMM 不仅需要通知最高层成员,还有 TMStub 的内部成员.假设当前 $R_{A\text{-}stub}$ 的成员压缩比为 $r\in[0,1]$,且相关 TMStub 由 m_1 个成员组成(包括 Mstub, Mtransit 和 sub-TMStub),前者的最大通信开销都为 $(r\times|R_{A\text{-}stub}|-1)$,后者的最大通信开销为 m_1.基于此,它的最大通信开销为 $SC_{EAGLE}=n\times[(r\times|R_{A\text{-}stub}|-1)+m_1]$.在联盟构建初期,有时会生成规模较大的 TMStub,特别是该 TMStub 所含成员数量超过构建联盟 MStub 数量,即 $m_1>|R_{A\text{-}stub}|$,此时,SC_{EAGLE} 略大于 SC_{FAGLE};

(2) 如果没有生成新的 TMStub 或分裂已有的 TMStub,那么仅需通知最高层,此时,$SC_{E\text{-}Filtering}<<SC_{EAGLE}=$

$n\times[(r\times|R_{A\text{-}stub}|-1)]<<SC_{FAGLE}$.

证毕. □

3.1.4 存储开销

存储开销是指因记录联盟成员信息而带来的内存开销,它能反映构建系统的可扩展性.在基于 E-Filtering 的反匿名网络中,成员之间互相独立,无需记录信息,因此 $SS_{E\text{-}Filtering}$=0.在基于对等过滤的反匿名网络中,构建系统需要收集和管理成员信息,因此 FAGLE 或 EAGLE 都会产生存储开销.

- 前者的构建系统包括 3 个模块,分别为 RCM,MSM 和 AMM.其中,RCM 只记录 ACL 规则参数,存储开销可忽略不计,$SS_{F\text{-}R}$=0;MSM 需要记录所有成员的前缀,每个前缀占 5B,那么 $SS_{F\text{-}M}=5|R_{A\text{-}stub}|$B;AMM 除了成员前缀,还记录 *AS* 号与前缀的映射表,假设每个 *AS* 号占 4B 且每个 *AS* 只有一个网络前缀,那么 $SS_{F\text{-}A}=(5|R_{A\text{-}stub}|+9|R_{A\text{-}stub}|)$B;
- 后者的构建系统除了上述 3 个模块,还增加了 MTM 模块.只有 RCM 的开销没有变化:$SS_{E\text{-}R}$=0,其余模块都有不同.
  - 就 MSM 来说,如果它位于联盟最高层且成员压缩比为 r,那么 $SS_{E\text{-}MSM}=(5r\times|R_{A\text{-}stub}|)$B;反之,位于 TMStub 内部的 MSM 只需记录其上层提供商的前缀,因此 $SS_{E\text{-}MSM}$=5B;
  - MTM 除了上述开销,还需记录下层客户前缀.假设每个提供商的平均客户数量为 m,那么 $SS_{E\text{-}MTM}=(5r\times|R_{A\text{-}stub}|+5m)$B 或者 $SS_{E\text{-}MTM}=(5+5m)$B;
  - AMM 需要建立面向全网的层次结构模型(如图 1 所示),模型中,树节点的数据结构描述为{过滤状态位,TMStub 状态位,网络前缀,*AS* 号,孩子(客户)指针},共占(11+4m)B.假设全网的自治域数量为 z,那么 $SS_{E\text{-}A}=z\times(11+4m)$B.通过统计可知 z 约为 45k[31],进而可推出 $SS_{E\text{-}A}$ 是在可接受范围内.

综上所述,EAGLE 的 MSM 和 MTM 存储开销要小于 FAGLE,而 AMM 存储开销要大于 FAGLE.不过,当所有 *AS* 都加入联盟时,它们的 AMM 开销相差并不大.

3.1.5 规则优化误报

规则优化误报是指因运行过滤规则优化算法而带来的误报,这会间接影响构建系统的部署激励性.基于 E-Filtering 的反匿名网络对匿名包进行无差别过滤,不存在过滤器短缺问题,不需要规则优化,因此 $FP_{E\text{-}Filtering}$=0;基于对等过滤的反匿名网络对匿名包进行选择性过滤,过滤器需求量较大,需要规则优化.在 *M-List* 和 $F_{\max}$ 给定的条件下,影响规则优化误报的主要因素包括地址权重的准确度σ和优化解的质量.鉴于 EAGLE 和 FAGLE 方法都能求得最优解,误报率的计算公式可简化为 $FP=f(\sigma)$,σ越准确,*FP* 越小;反之,则越大.根据定义 12 可知,地址权重源于 *W-List*,这也就是说,它的准确度σ取决于 *W-List* 是否客观.已知 EAGLE 中的 *W-List* 成员是真实自治域,而 FAGLE 采用排他法来推测 *W-List* 成员,后者增加了随意性,不难推出$\sigma_{EAGLE}\geqslant\sigma_{FAGLE}$,因此 $FP_{EAGLE}\leqslant FP_{FAGLE}$.

3.1.6 规则优化的时间复杂度

规则优化的时间复杂度是指因运行规则优化算法而产生的的时间开销,这会间接影响构建系统的响应时间.基于 E-Filtering 的反匿名网络不存在过滤器缺乏问题,因此 $TF_{E\text{-}Filtering}$=0.给定 *M-List*,*W-List* 和 $F_{\max}$,FAGLE 的规则优化算法主要包括建立树(build tree,简称 BT)、裁剪树(prune tree,简称 PT)和压缩树(compress tree,简称 CT),其中,BT 需要进行(|*M-List*|+|*W-List*|)次节点插入操作,而每次插入最多执行 H 次比较,H 表示树的高度,最大值为 32,时间开销为 $o(32\times(|M\text{-}List|+|W\text{-}List|))$;PT 需要对树至少执行 5 次遍历,已知树节点数量最多为 $((|M\text{-}List|+|W\text{-}List|)\times(|M\text{-}List|+|W\text{-}List|-1)/2)$,它的开销为 $o(5\times(|M\text{-}List|+|W\text{-}List|)\times(|M\text{-}List|+|W\text{-}List|-1)/2)$;CT 需要为每个树节点计算过滤器分配方案,最大开销为 $o(|M\text{-}List|\times F_{\max}^2)$.基于此,$TF'_{FAGLE}=o(BT)+o(CT)+o(PT)$.需要注意的是:因为 FAGLE 的规则优化算法不支持增量更新,所以每当有成员加入或退出,必须重新运行一次算法,也就是说假设先后有 s 个成员需更新,那么时间开销为 $TF_{FAGLE}=s\times TF'_{FAGLE}$.EAGLE 的规则优化算法主要包括建立树 BT、前缀选择(prefix selection,简称 PS)和成员更新(member update,简称 MU),其中:BT 需要进行(|*M-List*|+|*W-List*|)次节点插入操作,而每次插入最多执行 32 次比较,因此它的开销为 $o(32\times(|M\text{-}List|+|W\text{-}List|))$;PS 需计算每个最优聚合前缀节点的过滤器分配方案,它的开销为 $o(N\times F_{\max}\times(F_{\max}-1))$,$N$ 表示树中最优聚合前缀节点数量,在

最坏的情况下(即文献[9]提高的非平衡树,它的每个叶子节点几乎都对应一个最优聚合前缀),$N=|M\text{-}List|-1$;MU只需重新计算更新节点上游的过滤器分配方案,而且可重用已有分支的分配方案,因此它的开销为 $o(H\times F_{\max}\times(F_{\max}-1))$,最坏情况下,$H=F_{\max}-1$.基于此,假设先后有 s 个更新成员,EAGLE 的时间开销为 $TF_{EAGLE}=o(BT)+o(PS)+s\times o(MU)$.通过比较不难发现,$TF_{E\text{-}Filtering}<TF_{EAGLE}<<TF_{FAGLE}$,联盟成员更新越频繁,这种优势越明显.

3.1.7 规则优化的空间复杂度

规则优化的空间复杂度是指因运行规则优化算法而产生的内存开销,这会间接影响构建系统的可扩展性.基于 E-Filtering 的反匿名网络不存在过滤器缺乏问题,因此 $SF_{E\text{-}Filtering}=0$.FAGLE 的规则优化算法会先后引入 3 种数据结构:ELCP 树 T、误报率数组 Z 和过滤器分配数组 R,其中:T 的最大节点数量为$(|M\text{-}List|+|W\text{-}List|)\times(|M\text{-}List|+|W\text{-}List|-1)/2$,内存开销约为 $o(1/2\times(|M\text{-}List|+|W\text{-}List|)^2)$;$Z$ 和 R 的最大长度为 $H\times|M\text{-}List|$,H 表示树的最大高度,它们的内存开销为 $o(32\times|M\text{-}List|)$.EAGLE 的规则优化算法先后引入 3 种结构:ELCP 树 T、误报率数组 Z 和过滤器分配数组 R,其中:T 的最大节点数量为$(|M\text{-}List|+|W\text{-}List|)\times(|M\text{-}List|+|W\text{-}List|-1)/2$,它的内存开销约为 $o(1/2\times(|M\text{-}List|+|W\text{-}List|)^2)$;$Z$ 和 R 的最大长度为 $H\times|M\text{-}List|$,H 表示树的最大高度,内存开销为 $o(32\times|M\text{-}List|)$.基于此,$SF_{FAGLE}=o(1/2\times(|M\text{-}List|+|W\text{-}List|)^2+64\times|M\text{-}List|)$.与 FAGLE 相似,EAGLE 的规则优化算法也引入 3 种结构,其中:BT 树的最大节点数量为 $32\times|M\text{-}List|$,最优解集 Sec 和误报率集 f 的最大长度也为 $H\times|M\text{-}List|$,内存开销为 $o(32\times|M\text{-}List|)$.基于此,$SF_{EAGLE}=o(96\times|M\text{-}List|)$.通过比较不难发现,$SF_{E\text{-}Filtering}<<SF_{EAGLE}\leqslant SF_{FAGLE}$.不过,FAGLE 中的 T 和 Z 都是中间数据,一旦运行结束,就会释放空间;而 EAGLE 为了实现增量更新,上述结构都不会被中途释放.简言之,就是用空间换取时间.

3.2 仿真实验

鉴于第 3.1 节的理论分析都是基于层次结构的网络拓扑,本节的仿真实验将在结构特征更加多样的真实网络拓扑上运行,以便对上述分析结果进行补充.考虑到真实环境的复杂性和实现的困难性,出于简化实验的目的,与文献[9]相同,本文也是通过搭建面向 BGP 的网络仿真平台来模拟域间网络的 IP 包活动情况.平台搭建过程分两步.

(1) 对美国 UCLA 大学收集到的自治域级拓扑数据集进行清理以及二次开发,获取可支持 C-BGP 的网络拓扑[15];

(2) 以真实拓扑作为输入参数,通过搭建面向网络模拟器 C-BGP 的开发环境来构建自治域级网络[16].

该仿真平台运行在一台 PC 虚拟机上(Intel 4core 2.2GHz processor,16GB of RAM, Parallels Desktop 12, Ubuntu 16.04).除了仿真任务,还需要提取网络特征、收集过滤器需求量以及优化过滤规则,上述计算任务都在另一台 PC 机(Windows XP SP3,Intel 2core 2.40GHz processor,2.0GB of RAM,VS 2008)上运行.

与 FAGLE 相比,EAGLE 的先进性在于提高了联盟成员的部署激励性,而这又取决于反匿名联盟的层次化程度.基于此,本节实验都将围绕能够反映联盟层次化的网络拓扑特征和影响部署激励性的因素(包括过滤器需求量、规则优化准确度)而展开.首先,从真实网络中提取与层次化相关的网络拓扑特征;然后,比较过滤器需求量的分布情况;最后,比较因规则优化而产生误报率的分布情况.

此外,E-Filtering 部署激励性差是因非部署收益,而与网络拓扑无关,因此它不必参与实验比较.

3.2.1 网络结构特征

根据第 2.1 节的描述可知,反匿名联盟的层次化程度源于网络结构的层次化,而后者又取决于以下拓扑因素:*AS* 之间是否存在大量 C2P 关系;*AS* 的前缀数量;上下层 *AS* 前缀之间的包含情况;*AS* 的路由备份情况;上下层 *AS* 的累积高度.基于此,本节将运行以下 5 组实验来反映真实网络的层次化程度.

(1) 第 1 组实验分别统计真实网络中 Stub 域、Transit 域、C2P、P2P 占 *AS* 总量或 *AS* 商业关系的比重,结果见表 1.在真实网络中,一方面 C2P 关系所占比例明显要高于 P2P 关系,这为联盟层次化奠定了基础;另一方面,Stub 域所占比例要远高于 Transit 域,这又为大规模逻辑域的建立提供了可能;

Table 1 Basic parameters of topological structure (%)

表 1 拓扑基本参数 (%)

Stub 比重	Transit 比重	C2P 比重	P2P 比重
83.4	16.6	65.9	34.1

(2) 第 2 组实验用来计算真实网络中拥有不同前缀数量的 *AS* 占总数量的比例,进而统计它的补充累积分布函数,结果如图 6 所示.虽然网络中存在少量前缀数量较大的 *AS*,但是绝大部分 *AS* 的前缀数量都只有 1 或 2 个,这与第 3.1 节的假设基本相同;

(3) 第 3 组实验用来计算上下层前缀存在不同包含关系的 C2P 数量占总数量的比例,进而统计它的补充累积分布函数,结果如图 7 所示.在真实网络中,上下层前缀之间存在包含关系的 C2P 数量只占了 10%左右,其中只有不足一半 C2P 的前缀包含比例超过 50%,这与第 3.1 节的假设(供应商前缀包含客户前缀)略不相同.原因可能是:虽然设备商已提供 CIDR 技术,但是由于 *AS* 管理协商困难或历史等原因,该技术并没有被广泛推广.但即使这样,依据 EAGLE 的松耦合性,相比于 FAGLE,它依然会带来少量收益.更何况在某些属权较为统一且 CIDR 技术使用率较高的专用网络(例如教育网)上,C2P 关系的前缀包含比例必然会大幅度提升.这也就意味着,本文方法更适合教育网等专属网络;

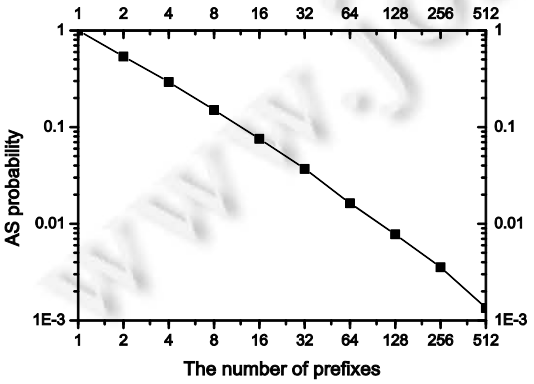

Fig.6 Number of prefixes on different *AS*

图 6 不同自治域的前缀数量变化情况

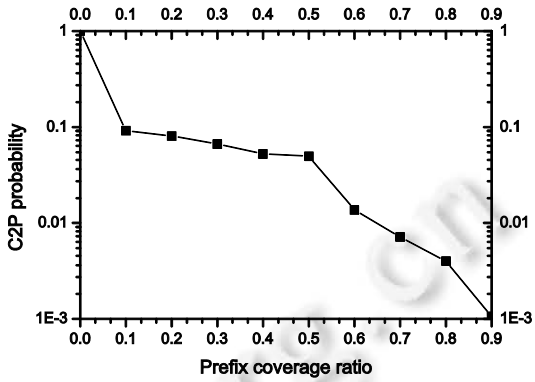

Fig.7 Prefix coverage ratio in different C2P

图 7 不同 C2P 关系的前缀包含比情况

(4) 第 4 组实验通过统计拥有多个提供商的客户数量占总数量的比例以及相关补充累积分布函数来说明单宿主或多宿主连接概率,结果如图 8 所示.无论客户是 Stub 域或 Transit 域,单宿主连接概率都会接近 50%及以上,这也就证明第 3.1 节的假设基本合理.至于剩余的那些采用多宿主连接的客户,无论它们是因为网络前缀太多或路由备份,都可以在不影响 EAGLE 性能的前提下,通过逻辑划分单宿主的方式来满足第 3.1 节的假设条件.

(5) 第 5 组实验通过统计逻辑域中 Stub 域数量占总数量的比例来说明不同高度逻辑域的生成概率,结果如图 9 所示.在真实网络中,一方面,高度大于 6 的逻辑域所占 Stub 域的比例不到 5%,而且大多是因为少量非常规的首尾相接的 C2P 关系而造成的;另一方面,接近 95%的 Stub 域位于高度不足 5 的逻辑域中,其中,3 层逻辑域就占了一半以上.基于此,本文在第 3.2.2 节选择 3 层逻辑域作为实验拓扑.

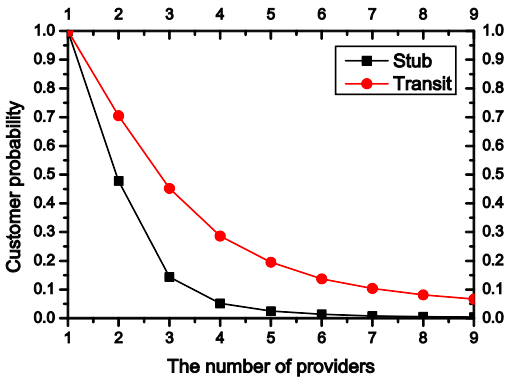

Fig.8 Number of providers on different multi-homed *ASs*

图 8 不同多宿主 Stub *AS* 的提供商数量变化情况

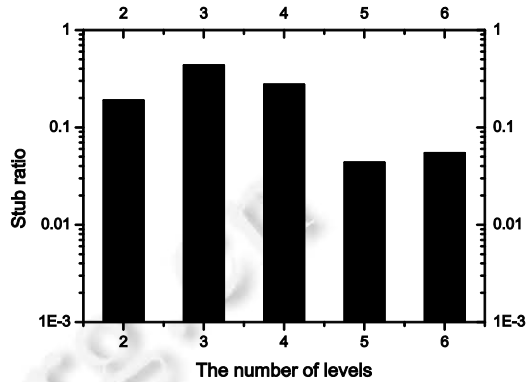

Fig.9 Number of stub *ASs* on different levels

图 9 不同层次中 Stub *AS* 数量变化情况

3.2.2 部署激励性

根据第 3.1 节可知,过滤器需求量和规则优化误报会间接影响反匿名联盟构建方法的部署激励性.为此,本实验主要关注在相同规模的反匿名联盟条件下 EAGLE 和 FAGLE 方法中过滤器需求量和规则优化误报的变化情况.为了简化实验步骤、突出比较结果,依据第 3.2.1 节已推测出的网络结构特征,本实验选则所有 3 层逻辑域作为实验拓扑,而且将所有多宿主 *AS* 简化为单宿主 *AS*.基于此,本节运行的两组实验如下.

(1) 第 1 组实验通过搜集不同联盟成员的过滤器需求量来统计它的补充累积分布函数,结果如图 10 所示.鉴于 EAGLE 方法的最高层成员过滤器需求量 FD_{Inter} 与 TMStub 内部成员过滤器需求量 FD_{Intra} 不同,需要分层搜集;而 FAGLE 只包含一种过滤场景,无需分层.首先,在 EAGLE 中,联盟成员的过滤器需求量从底层到高层逐层增加,且增幅较大.主要原因是为了完成对等过滤,除了本层前缀,上层的过滤规则还必须包含下层前缀.然后,从总体上看,与 FAGLE 相比,EAGLE 的过滤器需求量要降低很多;而且随着层数增多,优势会更明显.然而,实验结果与第 3.1.2 节理论分析结果略有不同,原因在于真实网络中上层前缀并不能完全覆盖下层前缀,进而无法简化过滤规则.最后,EAGLE 中最高层的过滤器需求量与 FALGE 的几乎相同,因为它们的过滤原理完全相同;

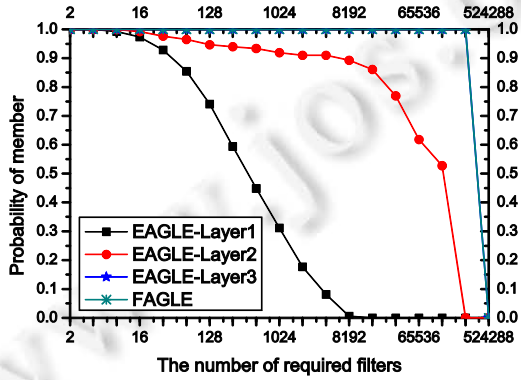

Fig.10 Number of required filters on different members

图 10 不同联盟成员的过滤器需求量变化情况

(2) 根据第 3.1.5 节可知:鉴于规则优化误报取决于地址权重准确度,第 2 组实验通过搜集所有 BT 树节点权重的绝对误差来统计它的补充累积分布函数,结果如图 11 所示.已知 EAGLE 地址权重的计算方法是客观的,而 FAGLE 带有不确定性,为了评估后者与前者的差距,本文提出了地址权重绝对误差

$d_{ip}=|w_{ip}^{EAGLE}-w_{ip}^{FAGLE}|$.为了突显实验的客观性,一方面假设所有 Stub *AS* 全部成为联盟成员,因此 $\forall w_{ip}^{EAGLE}=0\Rightarrow d_{ip}=|w_{ip}^{FAGLE}|$;另一方面,假设所有 Transit *AS* 都不是联盟成员,因为权重计算方法与联盟层次化无关.首先,从总体上看,EAGLE 与 FAGLE 的地址权重之间确实存在绝对误差,根附近节点的权重误差甚至达到 0.5 以上;然后,虽然绝大部分位于叶子和底部节点的地址权重误差都非常小,接近于 0,但是仍存在少量误差较大的节点,这些节点最终会影响规则优化算法;最后,部分节点的地址权重误差可达到 0,主要原因是下层自治域前缀之间不存在间隙,这意味着只有规划非常合理,FAGLE 与 EAGLE 的规则优化误报才可消除.

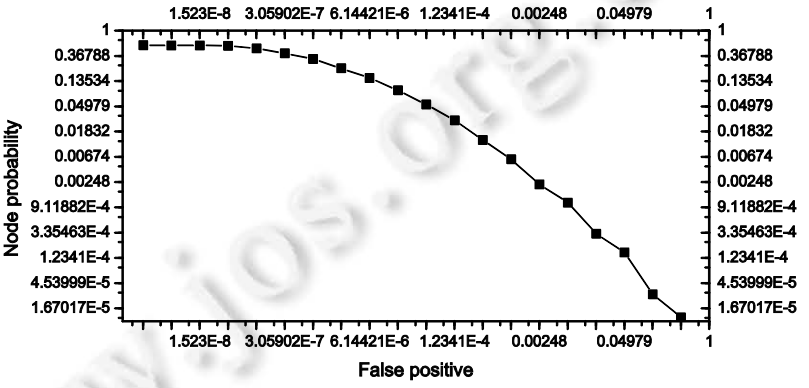

Fig.11 Rate of false positive on different tree nodes

图 11 不同树节点的过滤误报率变化情况

4 相关工作介绍

按照动作发生的先后顺序,反匿名技术可分为源地址验证和 IP 溯源两大类:前者实现事前预防,后者实现事后追责[17,18].二者在功能上互为补充,缺一不可,本文主则要关注前者.学术界已经提出大量的源地址验证方法,经典的有 DPF[1],IDPF[2],SAVE[3],SPM[4],Hidasav[5]和 Passport[6]等.DPF 方法利用核心网络大都依据最短路由转发 IP 包的特点,将凡是不应出现在当前路由上的 IP 包当作匿名包,并过滤.然而为实现该功能,部分边界路由器必须掌握全网路由,这并不现实.为此,Hai 等人提出了 IDPF 方法,通过配置 BGP 路由策略,将原方法的强假设——最短路由弱化为可行性路由,增强了可部署性.然而,这又造成较高的过滤漏报率.SAVE 方法提出了一种新的通信协议,通过在网络路由节点上建立源地址和入接口的映射关系,彻底实现了基于路由的匿名包过滤.不过,该协议没有考虑路由系统的层次结构,源地址或路由变化引发的消息更新范围达到整个网络,这既会产生较大通信开销,又会影响过滤的准确性.SPM 是一种基于加密认证的源地址验证方案,每对通信自治域都需共享一个密钥,其中源自治域负责写入,而目的自治域负责检查.然而,该方法过于扁平化和单一化的体系结构会造成较大的存储和通信开销.为此,吴建平等人提出一种层次化的域间真实源地址验证方法,简称 Hidasav,通过合理规划联盟层次和聚类整合,即使部署在大规模网络上,仍能保证验证的简单、轻权、有效.Passport 是一种基于对等密钥 MAC 的端到端匿名包过滤方法,通过计算上游 *AS* 号对应的 MAC 值并将它标记到转发包中,使得下游 Transit 域能够对该包进行验证和过滤.通过分析上述方法不难发现:它们都是对现有通信协议进行革新,进而要求彻底升级当前路由器.然而,本文关注的出边界过滤方法 E-Filtering 既不需要升级路由器,又不需执行额外操作,因此拥有更小的部署和计算开销.

尽管 E-Filtering 具备上述优点并已在路由器广泛实现,但是它的部署率依然较低,其中一个重要原因就是它缺乏部署激励性.为此,刘冰洋等人近年来提出一种基于对等过滤的反匿名联盟构建方法,与已有方案相比,它在可行性和效率方面具有明显优势[9].而本文就是在此工作的基础上做了以下努力:通过改善它的过滤器开销、通信开销、计算开销和过滤规则优化精度,进一步提高方法的可扩展性,适应可增量部署.

5 结论与未来工作展望

出边界过滤是当前网络流行的一种基于过滤器的反匿名技术,但是因为搭便车问题,缺乏部署激励性,致使它一直无法大范围推广.基于对等过滤的域间源地址验证方法就是针对上述问题而提出的,该方法借鉴社会学的互动关系理论,通过建立反匿名联盟,使得未部署运营商从受益者列表中严格剥离.然而,过于扁平化、单一化的联盟体系结构以及过于随机的非成员判定方式和低效的成员数据处理方式,使得它在面对大规模网络时存在可扩展性问题.为此,本文提出一种层次化的反匿名联盟构建方法,通过减小过滤器、通信、计算开销以及提高过滤规则优化精度来增强可扩展性,实现增量部署.与已有方法相比,本文方法具备以下特征:1) 构建了新型的层次化的反匿名联盟体系结构,避免成员之间建立不必要的全连接双向过滤关系,降低过滤器需求量;2) 引入 Transit 域对等过滤模块作为联盟边界,将每一层级联盟和外界成员隔离,减少因成员更新而带来的通信开销;3) 设计高精度、可增量更新的过滤规则优化算法,在加快求解速度的同时提高解的质量.

未来的研究工作主要包括:

1) 与公用网络(例如互联网)相比,本文的工作更适合于专用网络(例如中国教育网),主要原因如下:一方面,公用网络作为一个松散的商业联盟,很难集中管理所有自治域,更别说让它们服从于一个全局管理服务器 AMM;另一方面,由于历史遗留或部署激励不足,公用网络的上下层自治域网络前缀并非完全具备层次结构,虽然本文方法在非层次网络模型下也能正常工作,但是其优势会有所下降.基于此,我们希望在今后,以当前本文研究为基础放宽前缀层次化的假设条件,尽可能设计一种去中心化的体系结构.其目标是在不久的将来为互联网环境中的部署和商业应用提供参考和借鉴,以便为网络用户提供可靠的反匿名机制;
2) 虽然本文通过解决搭便车问题增强了传统 E-Filtering 的部署激励性,但是在反匿名联盟构建初期,因为联盟成员较少,根据定义 4 可知,新部署者只能获得较少的收益,这会再次影响构建方法的部署激励性.因此,如何利用早期少量成员所带来的市场压力刺激更多的新成员加入[19],使联盟构建方法从始至终都能保持较高的部署激励性是非常必要的.

References:

[1] Park KH, Heejo L. On the effectiveness of route-based packet filtering for distributed DoS Attack prevention in power-law Internets. ACM SIGCOMM Computer Communication Review, 2001,31(4):15−26. [doi: 10.1145/383059.383061]

[2] Hai DZ, Xin Y, Jaideep C. Controlling IP spoofing through interdomain packet filters. IEEE Trans. on Dependable and Secure Computing, 2008,5(1):22−36. [doi: 10.1109/TDSC.2007.70224]

[3] Jun L, Jelena M, Qiu WM. SAVE: Source address validity enforcement protocol. In: Proc. of the IEEE INFOCOM. 2002. 1557−1566. [doi: 10.1109/INFCOM.2002.1019407]

[4] Barr A, Band Levy H. Spoofing prevention method. In: Proc. of the IEEE INFOCOM. 2005. 536−547. [doi: 10.1109/INFCOM.2005.1497921]

[5] Li J, Wu JP, Xu K, Chen WL. An hierarchical inter-domain authenticated source address validation solution. Chinese Journal of Computers, 2012,35(1):85−100 (in Chinese with English abstract).

[6] Liu X, Li A, Yang X. Passport: Secure and adoptable source authentication. In: Proc. of the USENIX Symp. 2008. 365−378.

[7] Cisco IOS. Unicast reverse path forwarding. 1999.

[8] Ferguson P, Senie D. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827, 2000.

[9] Liu BY, Athanasios VV. Toward incentivizing anti-spoofing deployment. IEEE Trans. on Information Forensics and Security, 2014, 9(3):436−450. [doi: 10.1109/TIFS.2013.2296437]

[10] Halabi B. Internet Routing Architectures. 2nd ed., Beijing: Posts & Telecom Press, 2003.

[11] Liu BY. Deployability evaluation model and method design for inter-domain source address validation on the Internet [Ph.D. Thesis]. Beijing: Tsinghua University, 2014 (in Chinese with English abstract).

[12] Lu N, Wang YL, Shi WB. Filtering location optimization for defending against large-scale BDoS attacks. Chinese Journal of Electronics, 2017,26(2):435−444. [doi: 10.1049/cje.2017.01.016]

[13] Soldo F, Argyraki K, Markopoulou A. Optimal source-based filtering of malicious traffic. IEEE/ACM Trans. on Networking, 2012, 20(2):381−395. [doi: 10.1109/TNET.2011.2161615]

[14] Wang LJ, Wu JP, Xu K. BGP extension to support inter-domain distributed packets filtering. Ruan Jian Xue Bao/Journal of Software, 2007,18(12):3048−3059 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/3048.htm [doi: 10.1360/jos183048]

[15] Internet topology collection. 2012. URL:http://irl.cs.ucla.edu/topology

[16] Quoitin B. C-BGP. 2012. URL:http://c-bgp.sourceforge.net

[17] Lu N, Wang SG, Li F, Shi WB, Yang FC. An efficient and precise approach for single-packet traceback. Ruan Jian Xue Bao/Journal of Software, 2017,28(10):2737−2756 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5149.htm [doi: 10.13328/j.cnki.jos.005149]

[18] Lu N, Wang YL, Su S, Yang FC. A novel path-based approach for single-packet IP traceback. Security and Communication Networks, 2013,7(2):309−321. [doi: 10.1002/sec.741]

[19] Gill P, Schapira M, Goldberg S. Let the market drive deployment: A strategy for transitioning to BGP security. In: Proc. of the ACM SIGCOMM. 2011. 14−25. [doi: 10.1145/2043164.2018439]

附中文参考文献:

[5] 李杰,吴建平,徐恪,陈文龙.Hidasav:一种层次化的域间真实源地址验证方法.计算机学报,2012,35(1):85−100.

[11] 刘冰洋.互联网域间源地址验证的可部署性评价模型与方法设计[博士学位论文].北京:清华大学,2014.

[14] 王立军,吴建平,徐恪.支持域间分布式分组过滤的 BGP 扩展.软件学报,2007,18(12):3048−3059. http://www.jos.org.cn/1000-9825/18/3048.htm [doi: 10.1360/jos183048]

[17] 鲁宁,王尚广,李峰,史闻博,杨放春.一种高精度、低存储的单包溯源方法.软件学报,2017,28(10):2737−2756. http://www.jos.org.cn/1000-9825/5149.htm [doi: 10.13328/j.cnki.jos.005149]

鲁宁(1984－),男,内蒙古包头人,博士,副教授,主要研究领域为网络安全.

李峰(1978－),男,博士,讲师,CCF 专业会员,主要研究领域为机会网络,信任管理.

王尚广(1982－),男,博士,副教授,博士生导师,CCF 高级会员,主要研究领域为服务计算,移动云计算,车联网及网络安全.

史闻博(1980－),男,博士,副教授,博士生导师,主要研究领域为网络服务与网络智能化,物联网应用技术.

杨放春(1957－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为通信软件,网络安全,网络智能化.