基于路由证据的域间路由不一致路径恶意自治系统检测机制

蒋 健 李 伟 罗军舟 陆 悠 夏 怒

(东南大学计算机科学与工程学院 南京 211189)

摘 要 域间路由系统中,自治系统依据路由策略选择报文转发路径,并将路径通告给邻居自治系统.为了追求更多的网络利益,自治系统向邻居自治系统通告的路径可能并不是实际转发报文的路径,从而产生域间路由路径不一致的问题.域间路由不一致路径不仅欺骗正常自治系统的路由选路过程,损害其网络利益,而且对域间路由的稳定性也会造成严重影响.现有的检测机制主要关注不一致路径的检测,没有致力于发现导致路径不一致的恶意自治系统,无法为后续解决路径不一致的问题提供支撑.该文提出一种域间路由不一致路径恶意自治系统检测机制,采用路由证据将自治系统的路由行为与自治系统本身相绑定.路径源自治系统对路径中其他自治系统进行逆序比较路由证据,确定可疑自治系统,然后查询离自身较近的可疑自治系统的直接上游自治系统的路由通告历史记录,对可能接收可疑自治系统报文的自治系统请求路由证据,根据比较结果最终确定恶意自治系统.实验结果表明,该文的检测机制在查全率和查准率两个指标上均优于现有的检测机制,有效提高了检测不一致路径中恶意自治系统的准确率.

关键词 域间路由;路径不一致问题;恶意自治系统检测机制;路由证据;互联网

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2016.01147

A Routing Evidence Based Malicious AS Detection Mechanism for Interdomain Routing Inconsistent Path

JIANG Jian LI Wei LUO Jun-Zhou LU You XIA Nu

(*School of Computer Science and Engineering*, *Southeast University*, *Nanjing* 211189)

Abstract Autonomous System (AS) applies its own routing policy to select the data forwarding path and announce the path to neighbor domains in the interdomain routing system. To obtain more revenues, AS may announce the path which is inconsistent with the actual data forwarding path to neighbor domains, which causes the path inconsistency problem. The inconsistent path not only cheats the path selection of rational Autonomous Systems, but also harms their network revenues. Even the stability of Internet is destroyed seriously. Previous work only focused on the discovery of inconsistent path, and did not try to detect the malicious AS in the path, which could not support the following work on the problem. In this paper, we presented a malicious AS detection mechanism for the interdomain routing inconsistent path. It used routing evidence to bind Autonomous Systems with their routing behaviors. Source AS compared the routing evidence with other Autonomous Systems in the path to get the suspicious nodes. Source AS collected Route Log from the direct upstream node of suspicious node which closest to itself, and then it

收稿日期:2015-01-02;在线出版日期:2015-07-23.本课题得到国家自然科学基金(61320106007)、国家"八六三"高技术研究发展计划项目基金(2013AA013503)、高等学校博士点专项科研基金(20110092130002)、江苏省未来网络创新研究院未来网络前瞻性研究项目(BY2013095-2-07)、教育部计算机网络与信息集成重点实验室(东南大学)(93K-9)、江苏省网络与信息安全重点实验室资助项目(BM2003201)和住建部科学技术计划项目(2015-K6-012)资助.**蒋 健**,男,1986 年生,博士研究生,主要研究方向为网络管理.E-mail:jiangjian@seu.edu.cn.**李 伟**,男,1978 年生,博士,副教授,主要研究方向为下一代网络体系结构、服务计算.**罗军舟**,男,1960 年生,博士,教授,博士生导师,主要研究领域为下一代网络体系结构、协议工程、网络安全、云计算和服务计算.**陆 悠**,男,1977 年生,博士研究生,讲师,主要研究方向为下一代网络体系结构.**夏 怒**,男,1981 年生,博士研究生,主要研究方向为下一代网络体系结构.

compared the routing evidence with the nodes which may receive packets from the suspicious node to discover the malicious AS. The experiment results showed that our mechanism had a better performance than previous work from aspects of recall ratio and precision ratio, which could improve malicious AS detection.

Keywords interdomain routing; path inconsistency problem; malicious AS detection mechanism; routing evidence; Internet

1 引 言

Internet 中的自治系统(Autonomous System, AS)通过 BGP 协议(Border Gateway Protocol)[1]实现域间路由. 依据 RFC 4271[1],域间路由分为路由控制层和路由数据层两个层面:在路由控制层,自治系统从邻居自治系统获得 BGP 路由通告,依据自身的路由策略选择到达目的自治系统的最优路径,并将最优路径通告至其他邻居自治系统;在路由数据层,自治系统沿着最优路径转发数据报文. 正常情况下,自治系统在数据层转发报文的最优路径与控制层对外通告的最优路径是相同的. 然而由于商业利益驱动[2]、路由器配置错误[3]等原因,自治系统向邻居自治系统通告的路径可能并不是其实际转发报文的路径,同时产生域间路由路径不一致问题. 域间路由路径不一致问题在 Internet 中广泛存在. AS 级别的 Traceroute 实验显示 Internet 的域间路由中至少有 8%的路由路径存在数据层与控制层路径不一致的情况[4]. 由于恶意自治系统可以躲避 Traceroute 实验的检测,因此域间路由可能存在更多的不一致路径. 域间路由不一致路径带来一系列问题,一方面使得自治系统的路由选路过程被欺骗,路由策略无法在路由路径中体现,自治系统利益受损;另一方面,由于恶意自治系统所发布的路由通告并非自身最优路径,接收通告的相邻自治系统也无法保证最优的路径,一定范围内会导致路由路径出现环路,影响域间路由的稳定性[5].

近年来,域间路由路径不一致问题逐渐获得研究人员的关注,相关研究工作主要集中在不一致路径检测方面,已取得了许多成果[6-8]. 这些检测机制多采用发送探测报文的形式,利用探测报文反馈的结果检测路径. 然而这些检测机制并不能保证探测报文与正常报文的无差异性,所以使得检测机制不能对恶意自治系统透明,影响检测结果;同时由于这些检测机制主要关注不一致路径的检测,没有致力于发现导致路径不一致的恶意自治系统,所以无法为后续解决路径不一致问题提供支撑[9].

鉴于以上原因,本文提出一种域间路由不一致路径恶意自治系统检测机制(Malicious AS Detection Mechanism, MADM),在发现不一致路径的基础上,检测出导致路径不一致的恶意自治系统. 本文首先提出路由证据的概念,自治系统在接收和转发数据报文的同时生成路由证据. 路由证据反映自治系统声明的报文转发路径,将自治系统的路由行为与自治系统本身相绑定. 其次本文提出基于路由证据的恶意自治系统检测方法,源自治系统依据证据比较结果,确定路径中的可疑自治系统,查询离自身较近的可疑自治系统的直接上游自治系统的路由通告历史记录,对可能接收可疑自治系统报文的自治系统请求路由证据,依据证据比较结果最终确定恶意自治系统.

和已有研究成果相比,MADM 具有以下 3 个方面的优势:(1) MADM 是一种轻量级检测机制. MADM 对路由证据进行加密以保证路由证据的安全性,其不需要进行报文级别的加密,显著减少了网络开销;(2) MADM 提高检测恶意自治系统的准确率. 在路由证据的生成过程中,路径中的恶意自治系统无法辨别证据生成报文和正常报文,不能针对证据生成报文改变转发路径,无法躲避检测;(3) MADM 避免恶意自治系统对检测结果的影响. MADM 检测过程不与恶意自治系统直接交互信息,仅与正常自治系统比较路由证据即可发现恶意自治系统,对恶意自治系统透明.

本文第 2 节介绍不一致路径检测机制的相关工作;第 3 节形式化描述域间路由路径不一致问题;第 4 节介绍路由证据和不一致路径恶意自治系统检测方法;第 5 节对 MADM 进行实验验证;第 6 节总结全文并提出未来工作.

2 相关工作

近年来研究人员在路由数据层部署测量机制检

测路由路径，然而即使不考虑检测恶意自治系统，在数据层发现完整的路由路径仍然是一项繁重的工作，同时数据层的测量机制一般需要报文级别加密，会造成很大的网络开销. 研究人员[6]试图从 Traceroute 数据中得到完整的路由路径，然而由于路由聚集和路由过滤、边界路由器接口编号混乱、ICMP 报文接口错误、恶意路由等原因，得到的结果和实际路径存在很大的偏差. Secure traceroute[7]在自治系统之间设立一个共享的密钥谓词，自治系统选择一些报文作为探测报文并检测报文是否能够正常到达目的节点的元素. 它要求路由器对探测报文的每个应答信息计算消息鉴别码(Message Authentication Codes，MAC)，在探测报文数量庞大的情况下，路由器会消耗巨大的计算资源. Icing[8]提出了一种路由策略优化加密结构，要求报文发送者和报文接收者共享报文转发路径所涉及的路由策略. 由于自治系统的高度自治性，策略共享在实际网络中难以实现.

Zhao 等人[10]提出了一种域间路由决策检测协议，允许自治域的对等邻居检测自治域是否遵守了路由策略承诺. 除了域间路由协议公开的信息之外不需要暴露其他域内隐私，多个邻居自治系统通过检测公开信息即可得出检测结果. 与其他研究方法相比，Zhao 等人的检测机制需要自治域提前向邻居自治域提供路由选择承诺(Promise)，这种假设在当前域间路由系统中并不成立. 其次它只能判断自治域通告的路径是否违背承诺，即只能够验证路由控制层通告路径的真实性，并没有关注路由控制层和数据层的路径不一致问题. 类似的工作有 NetReview[11]，它检测一个自治域是否偏离对其他域的路由策略承诺. 为了实现此方法，自治域必须公开所有接收到的 BGP 更新消息. 这种需求不能保护自治系统的隐私，另外 NetReview 和 Zhao 等人的工作类似，只关注路由控制层的路由行为验证问题.

Wong 等人[12]提出一种轻量级的路由路径验证机制 Lightweight. 验证方在一些数据报文中附加共享密钥元组的前半部分并发送数据报文至被验证方. 被验证方拥有共享密钥元组的后半部分，收到数据报文后发送应答报文. 验证方依据应答报文判断被验证方的上游自治系统是否沿着当前路径转发报文. 与本文类似，它需要多个自治域的协作，同时必须确保附加共享密钥元组的报文与常规报文一致，否则恶意自治系统将探测报文正常转发，而将正常报文则转发至其他路径. 虽然 Lightweight 可以检测当前路径是否存在路径不一致问题，但是无法准确定位不一致路径中的恶意自治系统，不能为后续解决路径不一致问题提供支撑.

3　域间路由路径不一致问题的形式化描述

为了便于理解，本节对域间路由路径不一致问题进行形式化描述，在此基础上给出具体实例.

采用有向图 $G=(V,E)$ 表示网络域间路由拓扑，其中 $V=\{v_0,v_1,\cdots,v_n\}$ 表示域间自治系统集合，v_n 表示一个域间节点(为形式化表述方便，自治系统在本文中也被称为节点).

$E=\{e_{v_i,v_j}\mid v_i\in V,v_j\in V,v_i\neq v_j,i\geq 0,j\geq 0\}$ 是相邻自治系统直连边的集合，其中 e_{v_i,v_j} 表示节点 v_i 到节点 v_j 的有向边，边的方向表示报文传输方向.

定义 1.　路由路径. 路由路径 P 以路径中的节点序列表示，$P=(v_0,v_1,\cdots,v_i,d)$，v_0 为路径 P 的源节点，d 为路径的目的节点，v_i 是路径 P 中的第 $i+1$ 个节点.

定义 2.　上下游节点. 如果 $v_i\in P,v_j\in P$，且 $P=(v_0,v_1,\cdots,v_i,\cdots,v_j,\cdots,d)$，$i<j$，则 v_i 为 v_j 的上游节点，v_j 为 v_i 的下游节点. 如果 v_i 和 v_j 是邻居节点，则 v_i 为 v_j 的直接上游节点，v_j 为 v_i 的直接下游节点.

根据 BGP 的路由选路过程可知 BGP 路由通告总是由直接下游节点发送给直接上游节点.

定义 3.　路径权值. 路径权值反映自治系统的路由策略. 节点 v 利用路径权值函数 $W_v(P)$ 计算路由路径的权值，每条路由路径的权值不同，权值最高的路径为最优路径.

一般情况下 $W_v(P)>0$，如果节点接收的路由路径 P 存在环路，则 $W_v(P)=-\infty$. 基于以上定义，我们给出路径不一致问题和恶意自治系统的定义.

定义 4.　路径不一致问题. 设节点 v_0 依据路由策略选择到达目的节点 d 的最优路径为 P，表示为 $P=(v_0,v_1,\cdots,v_i,d)$. 设数据报文从 v_0 到 d 的实际传输路径为 P'，表示为 $P'=(v_0',v_1',\cdots,v_j',d)$. 如果存在节点 v_s 使得 $v_s\in P\cup P'$ 且 $v_s\notin P\cap P'$，则我们称路径 P 存在不一致问题.

路由策略是指自治系统制定的接收路由、选择最佳路由以及对外通告路由的策略，具有多样性和自主性的特点. 路由策略的多样性首先表现在决定路由策略的因素多样化. 典型因素包括自治系统之间的商业关系、路径长度和流量负载等；其次是路由

策略的偏好多样化. Gao-Rexford[13] 首次提出了自治系统商业关系的概念,将自治系统分为客户自治系统、服务自治系统以及对等自治系统. 客户自治系统需要向服务自治系统支付费用以获得网络接入权限. 为了追求更多的经济利益,服务自治系统会选择通过客户自治系统的路径发送报文,同时将最优路径通告给上游的客户自治系统. 而某些自治系统则考虑报文传输的实时性要求,即使付出一定的经济损失,也对最短路径有所偏好. 路由策略的自主性表现在策略的自主决策,其他自治系统无法决定也无法获知本自治系统的最终策略.

根据 RFC4271 对域间路由的描述,我们可以发现域间路由的默认准则即自治系统向邻居自治系统通告的路径必须是自身选择的最优路径. 虽然路由策略具备多样性和自主性,但是如果自治系统在制定以及实施路由策略的过程中违反了此准则,对其他自治系统利益和网络路由系统造成危害,则被称为恶意自治系统,定义如下.

定义 5. 恶意自治系统. 设节点 v_0 到目的节点 d 的最优路径 P 存在路径不一致问题,表示为 $P=(v_0,v_1,\cdots,v_i,d)$. $P[v_r]=(v_r,v_{r+1},\cdots,v_i,d)$ 表示 P 中源节点为 v_r,目的节点为 d 的子路径,其中 $r\in(0,i]$. 设 P_{v_r} 为 v_r 依据自身路由策略选择的到达 d 的最优路径,如果 $P_{v_r}\neq P[v_r]$,则 v_r 为恶意自治系统.

图 1 是不一致路径的实例. 节点 a 的 3 条路径 abd、ad 和 $abcd$ 的路径权值关系为 $W_a(abd)>W_a(ad)>W_a(abcd)$. 相应的,$b$ 的路径权值关系为 $W_b(bcd)>W_b(bd)$. 图的左侧不存在路径不一致问题. 节点 b 沿着路径 bcd 发送报文至 d,同时将 bcd 通告给上游节点 a. 在 a 的路由策略中路径 ad 的权值高于 $abcd$,因此 a 选择 ad 作为最优路径,将报文直接发送至 d 节点. 图的右侧存在路径不一致问题. 假设 a 是 b 的客户节点,由于 bcd 并不被 a 选为最优路径,为了吸引 a 的流量,节点 b 向 a 通告非最优路径 bd. 虽然 bd 不是 b 的真实报文转发路径,但是 a 并不知晓 b 如何转发报文,因此 a 节点选择路径 abd 发送报文,abd 成为不一致路径.

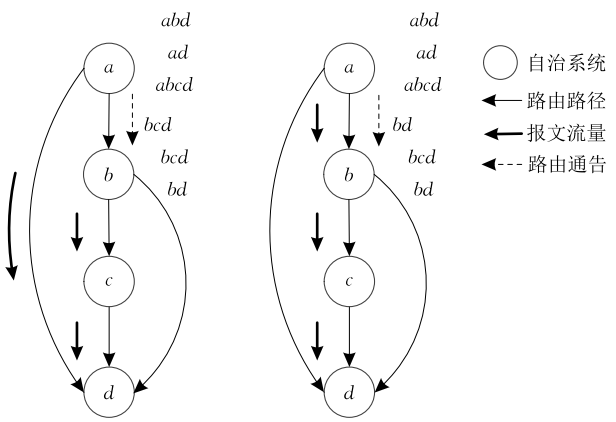


图 1 路径不一致问题实例

4 域间路由不一致路径恶意自治系统检测机制

在本节我们首先介绍路由证据的概念,其次设计基于路由证据的恶意自治系统检测方法,最后讨论 MADM 的实现和部署方案.

4.1 路由证据

路由证据反映自治系统的路由行为. 根据路由证据的生成节点和生成方式,路由证据分为 3 类:源节点报文发送证据(Source Sending Evidence,SSE)、目的节点报文发送证据(Destination Sending Evidence,DSE)和目的节点报文接收证据(Destination Receiving Evidence,DRE). 自治系统采用 RSA 密钥算法生成本自治域的公钥和私钥,在全网内发布公钥并获取其他自治系统的公钥,利用密钥加密路由证据保证路由证据的鲁棒性.

在生成路由证据之前,源自治系统设计证据生成报文的发送策略. 首先制定证据生成报文的发送时间区间 T. T 的起始时间晚于源自治系统的当前时间. 源自治系统将 T 平均分为若干个小区间,在小区间中随机选择 k 个不连续小区间发送证据生成报文,每个小区间内报文发送的持续时间也不同. 例如源自治系统将 1000 ms 划分为 10 个区间,每个区间为 100 ms,选择 5 个不连续区间,即 2、4、6、8 和 10,在这 5 个区间内发送证据生成报文;其次还要设计证据生成报文格式. 依据 RFC791①,IP 报文分为头部和载荷两个部分. 载荷包含报文的主要内容,而头部则包含一系列报文选项,例如报文源地址,报文目的地址,时间戳等. 为了防止证据生成报文被恶意自治系统辨别,源自治系统不对报文进行任何加密操作,仅在报文的时间戳附加报文的发送时间. 由于报文发送的时间区间是不连续的,且不同区间内报文发送的持续时间也存在差异,因此相同区间内报文的附加时间戳是相近的.

证据生成报文发送策略制定完毕后,源自治系

① IP, ftp://ftp.ietf.org/rfc/rfc0792.txt

统向目的自治系统发送路由证据请求消息(Request of Routing Evidence,RRE),定义如下.

定义 6.　路由证据请求消息 RRE.

$RRE=\{\{RET_{flag},Time_s,Time_e,k,IP_u\}_{RK_s}\}_{PK_d}$,

其中,RET_{flag}是证据请求类型标志位.如果源自治系统请求的证据类型是 DRE,$RET_{flag}=0$;当源自治系统请求的证据类型是 DSE 和 DRE 时,$RET_{flag}=1$.$Time_s$是整个报文发送区间 T 的起始时间,结束时间为 $Time_e$.k 表示发送报文的小区间数量,IP_u是目的自治系统的直接上游自治系统 IP 地址.源自治系统用私钥 RK_s加密消息保证其不被中途恶意自治系统篡改,用目的自治系统公钥 PK_d加密消息保证只有目的自治系统可以获悉消息内容.

目的自治系统接收 RRE 后,确认自身当前时间早于 $Time_s$,向源自治系统返回路由证据确认消息(Acknowledgement of Request,AOR),定义如下.

定义 7.　路由证据确认消息 AOR.

$AOR=\{\{RET_{flag},Time_s,Time_e,k,IP_u\}_{RK_d}\}_{PK_s}$,

其中,RET_{flag},$Time_s$,$Time_e$,k 和 IP_u与 RRE 消息中对应项的值相同,目的自治系统用私钥 RK_d和源自治系统公钥 PK_s加密消息.

源自治系统收到 AOR 后,确认目的自治系统收到 RRE 消息.当时间到达 $Time_s$时,源自治系统执行报文发送策略,在 k 个不连续的时间区间内发送报文.如果当前时间晚于 $Time_s$,源自治系统需重新制定报文发送策略.

源自治系统统计每个发送区间内发送的报文长度 s,获得报文长度集合 $S=\{s_1,s_2,\cdots,s_k\}$,对集合中的元素取对数以减少异常值的影响,得到 $S_{\log}=\{\log(s_1),\log(s_2),\cdots,\log(s_k)\}$,继而生成源自治系统报文发送证据 SSE.

定义 8.　源自治系统报文发送证据 SSE.

$SSE=\{SSE_{label},Time_s,Time_e,k,IP_d,S_{\log}\}_{RK_s}$,

其中,SSE_{label}是 SSE 标签,$Time_s$是 T 的起始时间,结束时间为 $Time_e$.k 表示发送报文的小区间数目,IP_d是目的自治系统的 IP 地址,$S_{\log}$是报文长度集合,源自治系统用私钥 RK_s加密证据内容.

考虑不同自治系统的时钟可能不同步,为了避免时钟不同步对路由证据的生成造成影响,目的自治系统向源自治系统反馈 AOR 信息后即开始监听边界路由器连接直接上游自治系统 IP_u的网络端口,收集源地址为源自治系统 IP、时间戳介于 $Time_s$和 $Time_e$之间的报文,并根据 RRE 中 RET_{flag}的值生成相应的证据.

由于 IP 协议的 best-efforts 机制,目的自治系统接收的 IP 报文可能是混乱排列的.为了生成正确的路由证据,它必须采用 K-means[14] 聚类算法对接收的数据报文进行聚类处理.算法以报文时间戳为聚类特征值,随机选择 k 个对象代表 k 个区间内报文时间戳的平均值.执行算法得到报文长度集合为 $S'=\{s'_1,s'_2,\cdots,s'_k\}$,对报文长度集合中元素取对数可得 $S'_{\log}=\{\log(s'_1),\log(s'_2),\cdots,\log(s'_k)\}$.目的自治系统依据报文分析结果生成报文接收证据 DRE.

定义 9.　目的自治系统报文接收证据 DRE.

$DRE=\{\{DRE_{label},Time_s,Time_e,k,IP_s,IP_u,S'_{\log}\}_{RK_d}\}_{PK_s}$,

其中,DRE_{label}是 DRE 标签,$Time_s$、$Time_e$、k 与 SSE 中对应项的值相同.IP_s是源自治系统的 IP 地址,IP_u是目的自治系统的直接上游自治系统的 IP 地址,$S'_{\log}$是报文长度集合,目的自治系统用私钥 RK_d和源自治系统公钥 PK_s加密证据内容.

根据 DRE 的生成过程,可以发现当路径是正常路径时 DRE 与 SSE 是匹配关系,即对于任意的 SSE,目的自治系统接收来自源自治系统的证据生成报文,生成唯一的 DRE,使得证据内容 $Time_s$、$Time_e$和 k 相同,同时 $S_{\log}=S'_{\log}$.实际网络中链路拥塞等原因会造成网络丢包,这种情况下 $S_{\log}$与 $S'_{\log}$会存在一定的误差.为了解决这个问题,在此给出证据误差 μ 的计算方法:

$$\mu=1-\sum_{i=1}^{k}\left(\rho_i\times\frac{s'_i}{s_i}\right) \tag{1}$$

其中,ρ_i是第 i 个时间区间内发送报文的长度 s_i占整个时间区间发送报文长度的比例:

$$\rho_i=\frac{s_i}{\sum_{i=1}^{k}s_i} \tag{2}$$

设网络丢包率为 ξ,ξ在一定时间内会有波动.如果证据误差 μ 与时间区间 T 内 Max(ξ)的差值小于等于误差容忍度 θ,我们仍然认为 $S_{\log}=S'_{\log}$,即认为 SSE 和 DRE 是匹配的.随着网络丢包率的增加,我们可以加长报文发送区间,增加发送报文长度,同时增大误差容忍度,保证路由证据比较结果的准确性.

在 DRE 的生成过程中,目的自治系统需要监听到来自直接上游自治系统所有的证据生成报文,才可以生成与 SSE 匹配的 DRE.DSE 则反映目的自治系统是否将接收的证据生成报文沿着当前路径转发至直接下游自治系统.然而即使恶意自治系统将

证据生成报文转发至其他直接下游自治系统,它仍然可以生成与 SSE 匹配的 DSE,因此 DSE 反映的路由行为并不可靠.DSE 定义如下.

定义 10. 目的自治系统报文发送证据 DSE.

$$DSE=\{\{DSE_{label},Time_s,Time_e,k,IP_s,IP_{ds},S'_{\log}\}_{RK_d}\}_{PK_s},$$

其中,DSE_{label} 是 DSE 标签,$Time_s$、$Time_e$、k、IP_s、$S'_{\log}$与 DRE 中对应项的值相同.IP_{ds}是直接下游自治系统的 IP 地址.目的自治系统用私钥 RK_d和源自治系统公钥 PK_s加密证据内容.

4.2 基于路由证据的不一致路径恶意自治系统检测方法

MADM 需要路由路径中多个自治系统协同合作执行路径检测.路由路径的源节点是检测机制的发起节点.目的节点基本是网络的边缘节点(Stub AS),并不是传输节点(Transit AS),不存在通告不一致路径的动机,因此我们认为源节点和目的节点为正常节点.与相关研究工作[8,11]类似,本文只考虑路由路径中存在一个恶意节点的情况.

根据第 3 节恶意自治系统的定义,可知如果路径存在不一致问题,则路径中的恶意节点一定沿着其他路径将报文转发给非路径中的节点.受此启发,本检测方法以节点路由证据比较的方式,从目的节点开始逐个验证直接上游节点,判断直接上游节点是否沿着当前路径将报文转发至目的节点.

检测方法分为两个部分:第 1 部分是可疑节点检测算法,第 2 部分是恶意节点检测算法.在可疑节点检测算法中,源节点生成 SSE,与目的节点的报文接收证据 DRE 进行比较.当目的节点的 DRE 与源节点的 SSE 匹配时,将目的节点的直接上游节点替换为目的节点,继续比较证据.如果所有节点的 DRE 均与源节点的 SSE 匹配,表明当前路径不存在不一致问题.当某一节点的 DRE 与源节点 SSE 不匹配时,源节点向此节点的上游节点逐个请求路由证据 DRE 和 DSE,依据证据的比较结果确定可疑节点范围并将范围缩小至两个节点.在恶意节点检测算法中,源节点查询离自身较近的可疑节点的直接上游节点的路由通告历史记录,对可能接收可疑节点报文的节点请求路由证据 DRE,依据 SSE 和 DRE 的比较结果确定恶意节点.

4.2.1 可疑节点检测算法

在介绍检测算法之前,先给出证据比较布尔函数 $F(x,y)$的定义.

定义 11. 证据比较布尔函数 $F(x,y)$.证据比较布尔函数表示路由证据的比较结果:

$$F(x,y)=x\times y \tag{3}$$

其中,x 代表源节点生成的路由证据 SSE,y 代表目的节点生成的路由证据.根据布尔函数的定义[15],我们设定 $x\in\{1\}$,表示源自治系统总是依据报文发送的策略生成正确的 SSE.设定 $y\in\{0,1\}$,y 为 1 时表示目的自治系统生成与 SSE 匹配的路由证据,反之 y 为 0.相应的,$F(x,y)\in\{0,1\}$,$F(x,y)=1$ 表示证据匹配,$F(x,y)=0$ 表示证据不匹配.

设 $P=(v_0,v_1,\cdots,v_i)$,$i>0$ 是源节点为 v_0,目的节点为 v_i的路由路径.v_0生成 SSE,同时要求 v_i生成 DRE.我们用 σ_i和 τ_i分别表示此时的 SSE 和 DRE.v_0与 v_i比较证据,获得 $F(\sigma_i,\tau_i)$.如果 $F(\sigma_i,\tau_i)=1$,表明 v_i的直接上游节点 v_{i-1}沿着正常路径 $v_{i-1}v_i$转发报文.算法将 v_{i-1}标记为正常节点,将 v_i替换为 v_{i-1},继续检查 v_{i-1}的直接上游节点 v_{i-2}.当 $v_i=v_1$ 时,$\sum_{k=2}^{i}F(\sigma_k,\tau_k)=i-1$,表明路径的所有节点均可从直接上游节点正常接收证据生成报文,当前路径不存在不一致问题;如果存在节点 v_j,使得 $F(\sigma_j,\tau_j)=0$,表明 v_j没有从 v_{j-1}正常接收证据生成报文,导致其 DRE 与 v_0的 SSE 不匹配.当 $j=2$ 时,$F(\sigma_2,\tau_2)=0$,由于 v_0节点是正常节点,肯定发送证据生成报文至 v_1,并且算法的执行顺序与节点排列顺序相反,即 v_2已经被证明是正常节点,因此 v_1为恶意节点,整个检测算法结束.当 $j\in(2,i]$时,位于 v_0和 v_j之间的节点均有可能是恶意节点,用 V'表示 v_0与 v_j之间的节点,得到 $V'=\{v_1,v_2,\cdots,v_{j-1}\}$,为后续可疑节点检测提供信息.

v_0将 RRE 中的 RET_{flag}设置为 1,要求 V'内的节点同时生成 DRE 和 DSE,进行两类证据比较.为了表述方便,在证据比较布尔函数的基础上我们定义证据完全比较布尔函数 $R(x,z,r)$.

定义 12. 证据完全比较布尔函数 $R(x,z,r)$.证据完全比较布尔函数表示两类路由证据的比较结果:

$$R(x,z,r)=F(x,z)+F(x,r) \tag{4}$$

其中,x 代表源节点生成的路由证据 SSE,z 代表目的节点生成的路由证据 DRE,r 代表目的节点生成的路由证据 DSE.x 取值范围与 $F(x,y)$中 x 的取值范围相同,z 和 r 的取值范围与 $F(x,y)$中 y 的取值范围相同.据布尔函数的定义[15],$R(x,z,r)\in\{0,1\}$.$F(x,z)$和 $F(x,r)$均为 0 时 $R(x,z,r)=0$,表示目的节点不能产生与源节点 SSE 匹配的 DRE

和 DSE. $F(x,z)$和$F(x,r)$中只要有一个值为 1 时 $R(x,z,r)$即可为 1，我们在下文中根据具体场景分析其意义.

算法执行顺序仍然与节点排列顺序相反，v_0 首先与 v_{j-1} 比较路由证据. 我们用 σ_{j-1}，τ_{j-1} 和 δ_{j-1} 分别表示此时的 SSE，DRE 和 DSE. 依据 $R(\sigma_{j-1},\tau_{j-1},\delta_{j-1})$的取值，检测算法分为两种情况执行：

(1) $R(\sigma_{j-1},\tau_{j-1},\delta_{j-1})=1$，$v_{j-1}$ 是恶意节点，算法结束.

首先假设 $R(\sigma_{j-1},\tau_{j-1},\delta_{j-1})=1$ 时，$F(\sigma_{j-1},\tau_{j-1})$和 $F(\sigma_{j-1},\delta_{j-1})$均为 1，表明 v_{j-1} 从子路径 $v_{j-2}v_{j-1}$ 正常接收且声称正常发送证据生成报文至下游节点 v_j. 根据算法流程可知，$F(\sigma_j,\tau_j)=0$，即 v_j 从子路径 $v_{j-1}v_j$ 接收证据生成报文存在异常. 同时检测结果已经证明 v_j 是正常节点，它的路由证据是可信的. 上文提到 DSE 的可靠性无法保证，因此判定 v_{j-1} 没有生成真实的 DSE，是恶意节点. 由于 P 仅存在一个恶意节点，整个检测算法结束.

其次假设 $F(\sigma_{j-1},\tau_{j-1})=1$ 而 $F(\sigma_{j-1},\delta_{j-1})=0$，表明 v_{j-1} 从 v_{j-2} 正常接收证据生成报文但是并没有沿着子路径 $v_{j-1}v_j$ 发送证据生成报文，“承认”其没有沿着正常路径转发报文. 依据定义 5，我们同样认为 v_{j-1} 是恶意节点，整个检测算法结束.

需要注意的是 $F(\sigma_{j-1},\tau_{j-1})=0$ 且 $F(\sigma_{j-1},\delta_{j-1})=1$ 的情况没有讨论，因为根据路由证据定义，v_{j-1} 不可能在没有正常接收证据生成报文的情况下生成与 SSE 匹配的 DSE.

(2) $R(\sigma_{j-1},\tau_{j-1},\delta_{j-1})=0$，$v_0$ 继续向 v_{j-1} 的直接上游节点 v_{j-2} 请求路由证据 DRE 和 DSE，算法继续执行.

$R(\sigma_{j-1},\tau_{j-1},\delta_{j-1})=0$ 意味着 v_{j-1} 声称没有从子路径 $v_{j-2}v_{j-1}$ 正常接收和发送证据生成报文，此时无法判断证据真实性，不能对 v_{j-1} 进行判定，只能继续执行检测算法.

检测算法最终在 v_{j-h} 节点执行完毕，符合以下两个条件之一：$j-h\geq 1$ 且 $R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=1$，$R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=0$ 且 $j-h=1$. 依据 $R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})$的取值，算法执行结果分为 3 种情况：

(1) $j-h\geq 1$ 且 $R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=1+1=1$，可疑节点是 v_{j-h} 和 v_{j-h+1}，用 V'' 表示可疑节点集合，可得 $V''=\{v_{j-h},v_{j-h+1}\}$.

此时 $F(\sigma_{j-h},\tau_{j-h})$和 $F(\sigma_{j-h},\delta_{j-h})$均为 1，表明 v_{j-h} 从子路径 $v_{j-h-1}v_{j-h}$ 正常接收证据生成报文，并正常发送报文到 v_{j-h+1}. 由于算法是反序进行，依据上文分析可得 $R(\sigma_{j-h+1},\tau_{j-h+1},\delta_{j-h+1})=0$，即 v_{j-h+1} 声称没有从直接上游节点 v_{j-h} 正常接收证据生成报文，与 v_{j-h} 的路由证据矛盾，说明两个节点中必有一个节点伪造路由证据，因此两者构成可疑节点集合.

(2) $j-h\geq 1$ 且 $R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=1+0=1$，恶意节点是 v_{j-h}.

此时 $F(\sigma_{j-h},\tau_{j-h})=1$ 而 $F(\sigma_{j-h},\delta_{j-h})=0$，结合 v_{j-h+1} 的证据结果 $R(\sigma_{j-h+1},\tau_{j-h+1},\delta_{j-h+1})=0$，可得 v_{j-1} 是恶意节点.

(3) $R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=0$ 且 $j-h=1$，v_1 是恶意节点.

$R(\sigma_{j-h},\tau_{j-h},\delta_{j-h})=0$ 且 $j-h=1$，表明节点集合 $V'=\{v_1,v_2,\cdots,v_{j-1}\}$ 中所有节点均没有从直接上游节点正常接收证据生成报文. 由于 v_0 肯定发送了证据生成报文且 P 中只存在一个恶意节点，因此判断 v_0 的直接下游节点 v_1 的路由证据不真实，v_1 是恶意节点.

算法的详细流程见算法 1.

算法 1.　可疑节点检测算法.

输入：$P=(v_0,v_1,\cdots,v_i)$，$i>0$. $F(x,y)$，$R(x,z,r)$. SSE，DRE，DSE

输出：恶意节点 v_r 或者可疑节点集合 V''

```
1.  FOR v_i ∈ P DO
2.    IF F(σ_i,τ_i)=1 THEN
3.      i=i-1;
4.      IF v_i=v_1 THEN
5.        RETURN;
6.      ELSE
7.        CONTINUE;
8.      END IF
9.    ELSE
10.     j=i;
11.     IF v_j=v_2 THEN
12.       v_r=v_1;
13.       RETURN v_r;
14.     ELSE
15.       V'={v_1,v_2,…,v_{j-1}}, j∈(2,i];
16.       BREAK;
17.     END IF
18.   END IF
19. END FOR
20. IF R(σ_{j-1},τ_{j-1},δ_{j-1})=1 THEN
21.   v_r=v_{j-1};
22.   RETURN v_r;
23. ELSE
24.   FOR v_{j-2} ∈ V' DO
```

25. IF $R(\sigma_{j-2},\tau_{j-2},\delta_{j-2})=1+1=1$ THEN
26. $V''=\{v_{j-2},v_{j-1}\}$;
27. RETURN V'';
28. ELSE IF $R(\sigma_{j-2},\tau_{j-2},\delta_{j-2})=1+0=1$ THEN
29. $v_r=v_{j-2}$;
30. RETURN v_r;
31. ELSE IF $R(\sigma_{j-2},\tau_{j-2},\delta_{j-2})=0$ and $j-2=1$ THEN
32. $v_r=v_1$;
33. RETURN v_r;
34. ELSE
35. $j=j-1$;
36. CONTINUE;
37. END IF
38. END FOR
39. END IF

以一个具体实例描述算法过程(如图 2 所示),路由路径为 $P=(a,b,c,d,e)$. 源节点为 a,目的节点为 e,阴影圆形表示目的节点的直接上游节点. 依据检测算法,首先 a 向 e 请求路由证据 DRE,e 监听并分析直接上游节点 d 发送的报文,生成 DRE 并发送至 a. a 生成 SSE 并比较 SSE 与 DRE,如果证据内容匹配,则确认 d 沿着子路径 de 将报文发送至 e,d 是正常节点. 然后 a 将 d 作为目的节点,向 d 请求路由证据 DRE,对 c 进行检测. 如果检测 c 没有问题,算法将 c 作为目的节点继续检测 b. 当 b 为目的节点时表明路径的所有节点均为正常节点,路径不存在不一致问题,算法结束. 如果 e 不能产生与 a 的 SSE 相匹配的 DRE,表明 e 没有从子路径 de 获得证据生成报文,但是并不能确定 d 节点是恶意节点,因为 b,c 和 d 均可能通过 f 转发报文至 e,得到 $V'=\{b,c,d\}$(如图 3 所示). 依据检测算法,a 向 d 请求路由证据 DRE 和 DSE,如果 d 的路由证据结果 $R(\sigma_d,\tau_d,\delta_d)=1$,$a$ 确定 d 是恶意节点,算法结束. 如果 $R(\sigma_d,\tau_d,\delta_d)=0$,$a$ 继续向 c 请求路由证据 DRE 和 DSE. 当 $R(\sigma_c,\tau_c,\delta_c)=0$ 时继续向 b 请求路由证据. 当 $R(\sigma_b,\tau_b,\delta_b)=1$ 时,如果 $F(\sigma_b,\tau_b)$ 和 $F(\sigma_b,\delta_b)$ 均为 1,则 b 和 c 之间有一个节点是恶意节点,$V''=\{b,c\}$,否则 b 是恶意节点. 当 $R(\sigma_b,\tau_b,\delta_b)=0$ 时,b 仍然为恶意节点.

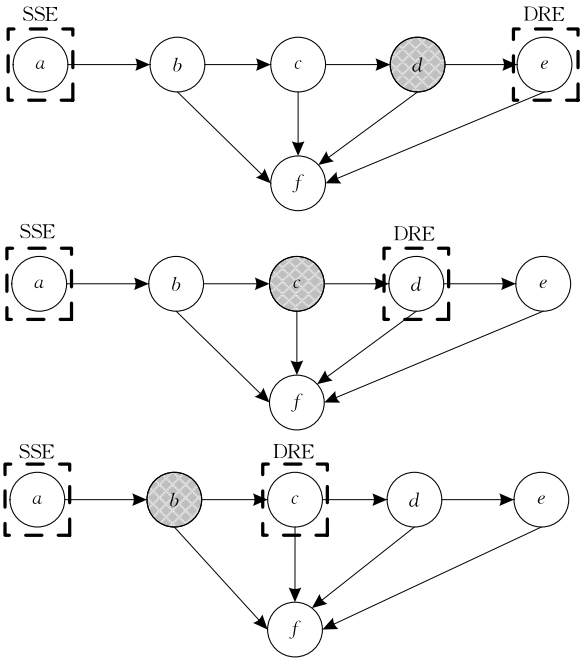


图 2 路径不存在不一致问题

4.2.2 恶意节点检测算法

由于 V' 内的节点均可能是恶意节点,它们的路由证据真实性无法保证,源节点无法与它们比较证据,因此恶意节点检测算法必须设计新的检测方式.

在介绍恶意节点检测算法之前,我们首先介绍路由通告历史记录(Route Log). Route Log 由自治系统维护,按照二元组{路由通告的发送节点,路由通告的目的节点}索引,记录相邻节点的路由通告历史信息.

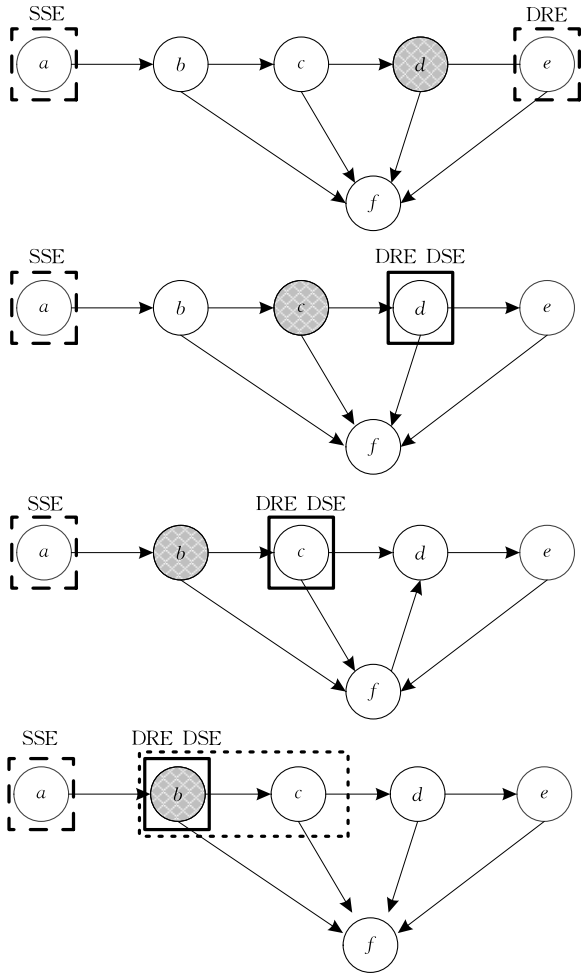


图 3 检测可疑节点 b,c 和 d

设 v_r 是节点 v_k 的邻居节点，则 v_k 的 Route Log 中 $U_{v_r,d}$ 表示 v_r 向 v_k 发送的目的节点为 d 的历史路由通告集合. $U_{v_r,d}$ 分为 3 个部分，分别是包含 v_k 当前最优路径子路径的路由通告 u_t（当路由通告中的路径被 v_k 选为最优路径时，我们称该路由通告包含 v_k 最优路径子路径），包含 v_k 上次最优路径子路径的路由通告 u_0 以及通告时间介于两者之间的路由通告 $u_1,u_2,\cdots,u_{t-1}$，可得 $U_{v_r,d}=\{u_0,u_1,\cdots,u_{t-1},u_t\}$.

随着路由路径的更新，Route Log 的存储信息周期性更新. 每当 v_k 收到来自 v_r 的目的地为 d 的路由通告时，均缓存在 Route Log 中. 假设 v_k 收到 u_n 后更新最优路径，则删除 u_t 之前的记录，添加 u_n 至 $U_{v_r,d}$，将 u_t 和 u_n 之间缓存的路由通告也添加至 $U_{v_r,d}$. 周期性更新保证了 Route Log 的存储空间在可接受范围内，以便于在自治系统中部署 Route Log.

图 4 是节点 a 的 Route Log 中 $U_{b,e}$ 的生成过程示意图. 当前 a 到 e 的最优路径是 $abcde$，a 保存 b 节点的路由通告 $bcde$. 上次 a 到 e 的最优路径是 $abge$，a 保存 b 节点当时的路由通告 bge. 两次通告间隔内 a 收到的通告 bfe 和 $bche$ 也被记录，最终得到 $U_{b,e}$（见表 1）.

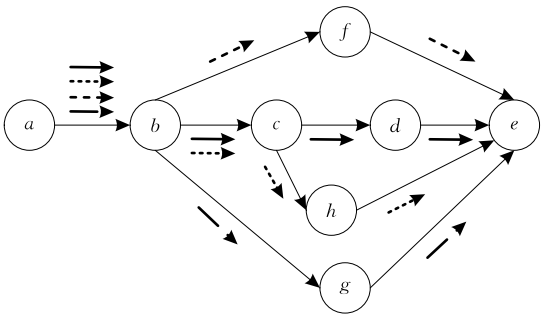


图 4　Route Log 中 $U_{b,e}$ 生成过程

表 1　$U_{b,e}$ 内容

路由通告	是否被选为最优路径
bge	是
bfe	否
$bche$	否
$bcde$	是

需要注意的是，如果 u_0 不存在，只存在 u_t，即 $(v_k)u_t$ 表示 v_k 首次将 v_r 的路由通告选为最优路径. 这种情况下 v_k 需要存储 v_r 发送的目的节点为 d 的所有路由通告，然而由于现实网络中相邻自治系统的连接关系是基于网络可达性创建的，因此这种情况很难存在①.

网络自治系统追求网络利益的最大化，研究成果[16]表明：产生域间路由不一致路径的主要原因是恶意节点为了增加网络利益而吸引正常节点的网络流量. 同时自治系统总是尽可能通过通告最优路径获得最多的网络利益，只有通告非最优路径能够获得更多的网络利益时才会通告非最优路径，据此我们提出定理 1.

定理 1.　设 $U_{v_r,d}=\{u_0,u_1,\cdots,u_{t-1},u_t\}$，$t>1$，表示节点 v_k 的 Route Log 中二元组 $\{v_r,d\}$ 标记的历史路由通告集合. 如果当前最优路径 $(v_k)u_t$ 是不一致路径且 v_r 是导致路径不一致的恶意节点，则 v_k 的实际报文转发路径是 $(v_k)u$，其中 $u\in U_{v_r,d}-u_t$.

证明.　$(v_k)u_t$ 是 v_k 当前最优路径，根据定义 3 可得 $W_{v_k}((v_k)u_t)$ 的值最大. 同时依据 BGP 的选路过程可知，v_r 向 v_k 通告 u_t，则 u_t 是 v_r 当前选择的最优路径. 然而 v_r 的实际报文转发路径 u 才是 v_r 真实的最优路径，可得 $W_{v_r}(u)>W_{v_r}(u_t)$.

节点 v_r 没有通告最优路径 u，而通告了非最优路径 u_t 至 v_k，表明 v_r 可能获悉在 v_k 的路由策略中 u_t 的权值高于 u 的权值. 然而由于自治系统路由策略的自主性，这种情况并不存在，因此 v_r 只能采用发送多次路由通告的方式试探 v_k. 依据 BGP 选路过程，v_r 首先向 v_k 通告真实路径 u，v_k 并没有选择 $(v_k)u$ 作为最优路径，v_r 继续向 v_k 通告非最优路径，直到通告 u_t 时才获得 v_k 的报文流量，因此可得真实路径 u 的通告时间早于 u_t 且 $u\neq u_t$.

$(v_k)u_0$ 是上次 v_k 到达 d 的最优路径，而当前最优路径是 $(v_k)u_t$. 一般来说，自治系统的最优路径改变有 3 种原因，分别是自治系统的路由策略变化导致最优路径权值改变；自治系统接收到最优路径撤销通告；自治系统接收到更符合其路由策略的路径通告. 接下来依次分析这 3 种原因：

(1) 假设 v_k 的最优路径变化是因为其路由策略发生变化. 由于 v_r 不能获悉 v_k 的最新路由策略，依据前文的分析，v_r 只能在 v_k 更新路由策略之后通告最优路径 u 进行试探，可知最优路径 u 的通告时间晚于 u_0，但也可能存在 $u=u_0$.

(2) 假设 v_k 的最优路径变化是因为 v_r 主动撤销路由通告 u_0. 由于 v_r 是恶意节点，因此 $(v_k)u_0$ 的一致性无法保证. 首先考虑 u_0 不是 v_r 当时最优路径的情况，前文提到 v_r 向 v_k 发送 u_0 的动机是吸引 v_k 的流量，此时 v_r 已经获得 v_k 的流量，没有撤销路由通告 u_0 的动机，因此这种情况不成立；其次考虑 u_0 是 v_r

① The Cooperative Association for Internet Data Analysis (CAIDA), http://www.caida.org/

当时最优路径的情况，v_r当前的最优路径是u，表明u成为其最优路径的时间晚于u_0，因此通告u的时间同样晚于u_0，但是同样存在$u=u_0$的可能.

(3)假设v_k的最优路径的变化是因为收到更符合其路由策略的路径通告，而该路径通告可能来自非v_r节点.当该路径通告来自非v_r节点时，v_k取消最优路径$(v_k)u_0$，v_r无法分辨v_k的路由策略是否改变，情况与(1)类似，v_r只能在v_k改变最优路径后通告最优路径u进行试探，可知最优路径u的通告时间晚于u_0，但是可能存在$u=u_0$.当该路径通告来自v_r时，表明v_r最优路径从u_0变为u，且u不同于u_0.由于u无法被v_k选为最优路径，v_r只能通过通告u_t获得v_k的流量.同样可知u的通告时间晚于u_0.

经过以上分析，我们发现包含真实报文传输路径子路径的路由通告u的通告时间介于u_0和u_t之间，u与u_t不同但是可能与u_0相同.根据$U_{v_r,d}$的定义，可得$u\in U_{v_r,d}-u_t$. 证毕.

由4.2.1节可知可疑节点集合为$V''=\{v_{j-h},v_{j-h+1}\}$，$j-h\geq 1$.为了在两个可疑节点中发现恶意节点，源节点与离自身较近的可疑节点的直接上游节点v_{j-h-1}协作.v_0向v_{j-h-1}申请查询Route Log，得到v_{j-h}向v_{j-h-1}发送的目的节点为v_i的路由通告历史记录U_{v_{j-h},v_i}. v_0处理U_{v_{j-h},v_i}中的信息，排除包含当前最优路径子路径的路由通告，找出剩下的历史路由通告中v_{j-h}的下一跳节点(Next hop)，并向所有的下一跳节点依次请求DRE(源节点可能就是v_{j-h-1}节点，此时$v_{j-h}=v_1$).根据定理1，如果v_{j-h}的某个下一跳节点反馈的DRE与v_0的SSE匹配，则证明v_{j-h}是恶意节点，反之v_{j-h+1}为恶意节点(具体见算法2).

算法2. 恶意节点检测算法.

输入：$P=(v_0,v_1,\cdots,v_i)$，$i\geq 0$. $V''=\{v_{j-h},v_{j-h+1}\}$，$j-h\geq 1$. $U_{v_{j-h},v_i}=\{u_0,u_1,\cdots,u_{t-1},u_t\}$，$t>1$. SSE，DRE

输出：恶意节点v_r

```
FOR EACH u∈U_{v_{j-h},v_i}−u_t DO
  IF SSE=DRE THEN
    v_r=v_{j-h};
    RETURN v_r;
  ELSE
    CONTINUE;
  END IF
END FOR
v_r=v_{j-h+1};
RETURN v_r;
```

以图4为例，假设b和c是可疑节点，为了在b和c之间找出唯一恶意节点，a首先得到b的路由通告历史记录bge，$bche$，$bcde$，bfe.由于$bcde$是当前最优路径，因此a仅选择bge，$bche$，bfe中的下一跳节点g、f和c进行证据比较.如果g、f和c中有一个节点产生与a的SSE匹配的DRE，则证明b是恶意节点，否则c是恶意节点.

4.2.3 算法分析

分析算法执行过程可以发现，源节点与路径中的节点最多进行一次证据交互，与历史路由通告中v_{j-h}的每个下一跳节点也仅交互一次证据信息，因此检测算法的计算复杂度取决于路径节点数和历史路由通告中v_{j-h}的下一跳节点数.

设路径总共包含n个节点，最坏的情况下，源节点的直接下游节点是可疑节点且最终被检测为恶意节点.源节点执行可疑节点检测算法，与目的节点比较SSE和DRE，生成V'. V'包含位于源节点和目的节点之间的$n-2$个节点.源节点与V'内的每个节点分别比较SSE和DRE、SSE和DSE，获得V''.源节点总共需要进行$2n-3$次证据比较.源节点执行恶意节点检测算法，设U_{v_{j-h},v_i}的历史路由通告中v_{j-h}的下一跳节点数是m，最坏的情况下，源节点必须与每个下一跳节点比较SSE和DRE，总共进行m次证据比较.综合两个算法，可得源节点总共进行$2n+m-3$次证据比较以发现恶意节点，算法复杂度为$O(2n+m)$.

通过分析MADM检测算法的执行过程，可知源节点并不直接与恶意节点交互信息，而是与恶意节点的邻居节点比较路由证据.这种方式对恶意节点透明，避免恶意节点干扰检测结果.同时节点之间的交互信息和路由证据均用密钥加密，防止中间人攻击影响检测结果，保证检测机制的安全性.

4.3 MADM的实现与部署

随着软件定义网络(Software-Defined Networking，SDN)[17]的蓬勃发展，基于控制器(Controller)的集中式路由控制方案也愈发成熟.为了实现MADM，可在自治域内的控制器中部署路由检测模块(Routing Detection Model，RDM)，集中管理自治域的密钥和路由证据，并维护路由通告历史记录，执行检测机制.

图5是RDM的结构图，总共包含5个功能子模块，分别是密钥生成子模块、密钥管理子模块、路由证据生成子模块、路由证据验证子模块和路由通告历史记录子模块.密钥生成子模块采用RSA算法

计算本自治域的密钥对. 我们在全网范围内采用 Chord 环[18]连接多个自治系统的 RDM，自治系统的 RDM 通过 Chord 环发布自身的公钥，并根据其他自治系统的 AS number 获取它们的公钥. 密钥管理子模块存储其他自治域的公钥信息，以便于对路由证据进行加解密操作. 路由证据生成子模块的主要功能是通过分析证据生成报文，生成相应的路由证据. 路由证据验证子模块将其他自治系统反馈的路由证据与自身的路由证据进行比较，依据比较结果检测路径和自治系统. 路由通告历史记录子模块存储相邻节点的路由通告历史信息. 研究人员通过统计 AT&T 的骨干网络的路由更新日志，得到现有自治系统到某一子网的路由平均一天会有 5 次更新，而到一些包含著名地址(如知名网站等)的路由平均一天只有 0.2 次更新[19]，因此每个自治域 RDM 的 Route Log 更新频率较低，在实际网络中易实现.

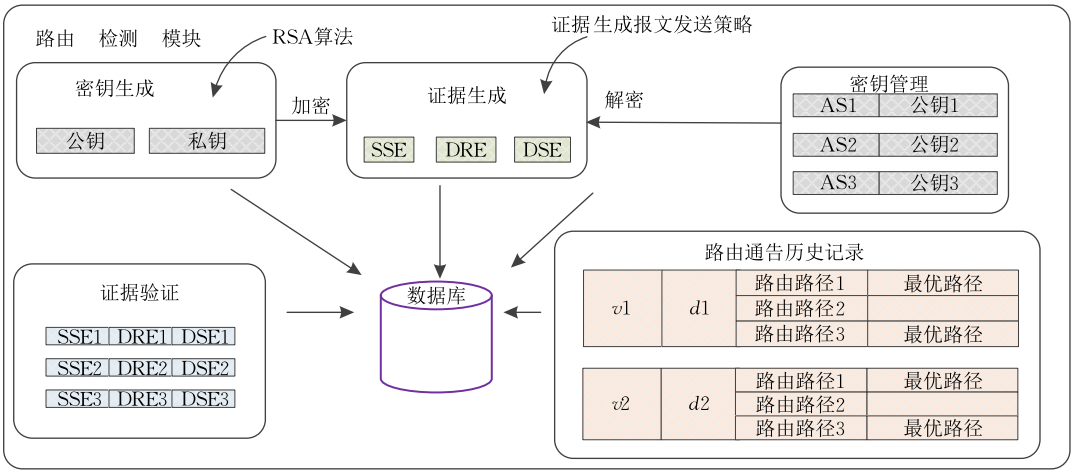


图 5　RDM 结构图

分析 MADM 的检测原理，我们可以发现对于一条路由路径的源节点，它并不需要全网自治系统的配合，而仅需要当前路径中的其他自治系统和恶意节点的邻居节点协作检测恶意自治系统. 因此在最初阶段，我们可以选择在小范围网络内，从网络安全和网络利益两个角度推动 MADM 在自治系统中的部署，进而实现全网范围内的部署.

5　实验与分析

为了更加直观地阐述 MADM 的价值，本节通过实验验证 MADM 对恶意自治系统的检测效果. 为了使得实验结果更具有说服力，我们实现了相关工作中介绍的 Secure traceroute 和 Lightweight 两种路径检测机制. Secure traceroute 在自治系统之间设立了一个共享的密钥谓词，自治系统选择一些报文作为探测报文，通过检测报文是否能够正常到达目的节点来发现不一致路径和恶意节点. Lightweight 在一些数据报文中附加共享密钥元组的前半部分，被验证方拥有共享密钥元组的后半部分. 被验证方收到验证方的数据报文后发送应答报文. 依据应答报文，验证方检测被验证方的上游自治系统. 我们将 Secure traceroute 和 Lightweight 对恶意自治系统的检测结果与 MADM 的检测结果进行比较，依据比较结果分析 MADM 的性能.

5.1　实验环境

我们采用 SSFnet① 对 MADM、Secure traceroute 和 Lightweight 进行仿真实验. 为了模拟现实网络中的拓扑，从 CAIDA② 项目组下载了 2014 年 6 月 11 日的网络拓扑和 AS 连接关系数据集 cycle-aslinks.l7.t1.c003225.20140607，在数据集中选择 300 个节点，其中 30 个节点为 Tier-1 节点，50 个节点为 Tier-2 节点，220 个节点为 Tier-3 节点. 基于节点之间的连接信息，采用 BRITE[20] 拓扑生成器生成节点的仿真拓扑. 我们从 Route View③ 项目组下载到 2014 年 6 月 11 日的域间路由更新通告数据集 updates.20140611 以及自治系统路由信息表数据集 rib.20140611，在数据集中找出仿真拓扑中节点的路由表和路由通告信息，在拓扑中构建路由路径.

在拓扑中选择 3 个节点 A，B，C 作为源节点，A 节点为 Tier-1 节点，连接度为 150，以 A 节点为源节

① The SSFnet network simulator. http://www.ssfnet.org/homepage.html
② The Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org/data, 2014.6.11
③ University of Oregon Route Views Project (Route View). http://www.routeviews.org, 2014.6.11

点的路由路径有 112 条. B 节点为 Tier-2 节点,连接度为 90,以 B 节点为源节点的路由路径有 45 条. C 节点为 Tier-3 节点,连接度为 30,以 C 节点为源节点的路由路径有 12 条(见表 2).

表 2 源节点参数设置

源节点	Tier	连接度	路由路径数
A	1	150	112
B	2	90	45
C	3	30	12

我们将网络报文传输速率设置为 100 Mbps,证据生成报文长度设置为 16 bytes,证据生成报文发送时间区间设置为 50 ms. 网络丢包率设置为 0.5%,误差容忍度设置为 1%. 在 MADM 检测过程中,采用 RSA 算法为每个参与检测过程的节点生成密钥对.

5.2 实验评估指标

本文采用相关研究中广泛使用的查全率和查准率作为实验评估指标. 在网络丢包率相同的条件下,分别测试 MADM、Secure traceroute 和 Lightweight 在不一致路径比例不同时检测恶意节点的查全率. 设拓扑包含 E 个恶意节点,被正确检测出的恶意节点数目为 F,则查全率(Recall Ratio)为

$$RR = F/E.$$

在不一致路径比例相同的条件下,分别测试 MADM、Secure traceroute 和 Lightweight 在网络丢包率不同时检测恶意节点的查准率. 设拓扑中总共被检测为恶意节点的节点数目为 H,则查准率(Precision Ratio)为

$$PR = F/H.$$

5.3 实验方案与实验结果

实验 1. 验证在不一致路径比例不同的情况下 MADM、Secure traceroute 和 Lightweight 对恶意节点的检测效果. 分别在 A、B 和 C 的路由路径中随机选择路径作为不一致路径,在不一致路径中任意选择一个节点作为恶意节点. 为了比较不同不一致路径比例下的实验结果,设定不一致路径占总路径的比例分别为 5%、8%、10%、15%和 18%.

实验对比分析结果如图 6～图 8 所示,图 6 是节点 A 的路由路径的检测结果. 可以发现,当不一致路径比例为 5%时,MADM 对其路由路径中恶意节点的查全率是 89%,仍然有大约 11%的恶意节点不能被检测. 分析拓扑发现虽然每条不一致路径只包含一个恶意节点,但是一部分路径在某些节点相交,导致这部分路径包含多个恶意节点. 由于 MADM 的检测算法只处理路由路径存在一个恶意节点的情况,因此不能检测这些恶意节点. Secure traceroute 对恶意节点的查全率是 60%,分析发现 Secure traceroute 检测的恶意节点大部分是没有分辨 probe 报文的恶意节点以及兄弟节点(Sibling AS),这些节点对 Secure traceroute 发送的 probe 报文给予正面回应,从而被检测出存在问题. 而剩余恶意节点可以分辨 probe 报文,并正常转发 probe 报文,因此并没有被 Secure traceroute 发现. Lightweight 对恶意节点的查全率是 68%,它与 MADM 一样均可以避免恶意节点的识别,但是查全率并不高. 分析发现只有当恶意节点是目的节点的直接上游节点时,Lightweight 才能检测出恶意节点,而非直接上游节点的恶意节点不能被检测出,因此影响了查全率. 随着不一致路径比例的增加,一条不一致路径出现多个恶意节点的情况增多,MADM、Secure traceroute 和 Lightweight 的查全率缓慢下降. 当不一致路径比例达到 18%时,MADM 查全率是 78.3%,Secure traceroute

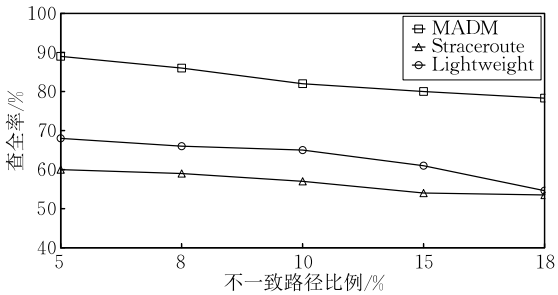


图 6 节点 A 的路由路径检测结果

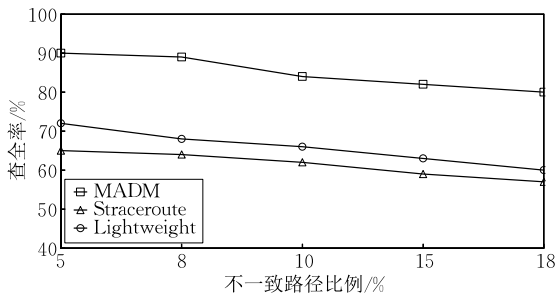


图 7 节点 B 的路由路径检测结果

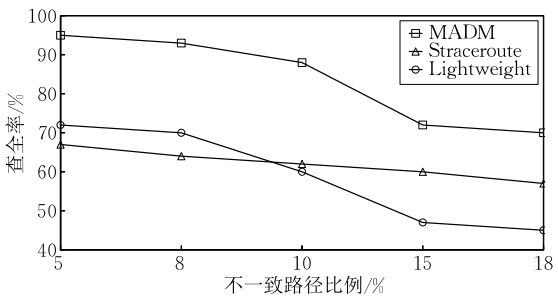


图 8 节点 C 的路由路径检测结果

查全率是 53.5%，而 Lightweight 则是 54.6%. 综合曲线可以发现 MADM 的查全率总体优于 Secure traceroute 和 Lightweight 的查全率.

图 7 显示节点 B 的路由路径的检测结果. 相比于节点 A，节点 B 的路由路径较少，一条路径存在多个恶意节点的情况也较少，因此当不一致路径比例相同时，MADM、Secure traceroute 和 Lightweight 对节点 B 的路径中恶意节点检测结果均优于对节点 A 的路径中恶意节点的检测结果. 图 8 显示节点 C 的路由路径的检测结果，当不一致路径比例是 5% 时，MADM 查全率达到 95%左右，Secure traceroute 的查全率是 67%，Lightweight 的查全率则是 72%. 观察曲线走势可以发现 MADM 的查全率优于 Secure traceroute 和 Lightweight 的查全率.

现实网络中不一致路径比例保持在 8%左右，这种情况下的实验结果表明 MADM 对恶意节点取得较高的查全率，优于 Secure traceroute 和 Lightweight 的检测结果.

实验 2.　验证在网络丢包率不同的情况下 MADM、Secure traceroute 和 Lightweight 对恶意节点的检测效果. 选择 A 作为测试节点，在 A 的路由路径中随机选择路径为不一致路径，设置不一致路径占总路径比例为 8%，不一致路径中任意选择一个节点为恶意节点. 分别设定网络丢包率为 0.5%，2.0% 和 5.0%，在不同的网络丢包率下执行 Secure traceroute 和 Lightweight，获得检测结果. 在网络丢包率为 0.5%时，设置证据生成报文发送区间为 50 ms，证据生成报文发送长度为 16 bytes，误差容忍度为 1%；在网络丢包率为 2.0%时，设置证据生成报文发送区间为 100 ms，证据生成报文发送长度为 32 bytes，误差容忍度为 5%；在网络丢包率为 5.0%时的证据生成报文发送区间设置为 150 ms，证据生成报文发送长度设置为 64 bytes，误差容忍度设置为 10%(见表 3). 在不同丢包率下执行 MADM，获得检测结果.

表 3　证据生成报文参数设置

网络丢包率/%	时间区间/ms	报文长度/bytes	误差容忍度/%
0.5	50	16	1
2.0	100	32	5
5.0	150	64	10

实验结果如图 9～图 11 所示，可以发现在不同的网络丢包率下，Secure traceroute 和 Lightweight 检测恶意节点的查准率不同. 图 9 是丢包率为 0.5% 的情况，Secure traceroute 的查准率尚能保持在 90%左右. 图 10 和图 11 显示丢包率上升时，Secure traceroute 发送的 probe 报文丢失率增加，节点没有接收到 probe 报文，不能反馈检测结果，因此 Secure traceroute 误判率上升，查准率下降. Lightweight 也有类似的情况，当丢包率为 0.5%时查准率在 89% 左右，而丢包率为 5.0%时，查准率已经下降到 49%.

在丢包率为 0.5%时，MADM 的查准率保持在 91%左右. 当丢包率上升的时候，由于采取增加证据生成报文发送时间、增加证据生成报文长度、同时调整误差容忍度的措施，图 10 和图 11 显示 MADM 的查准率仍然保持在 86%左右，表明 MADM 在网络丢包率较大时仍然具有较好的查准率.

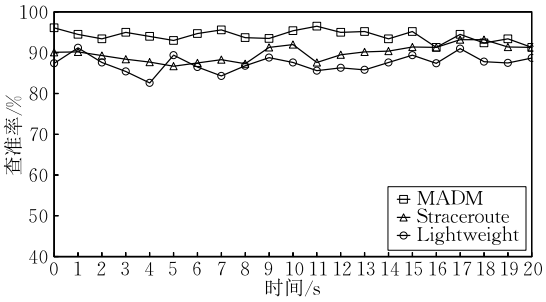


图 9　0.5%的网络丢包率

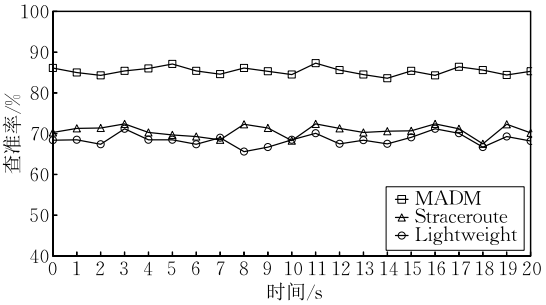


图 10　2.0%的网络丢包率

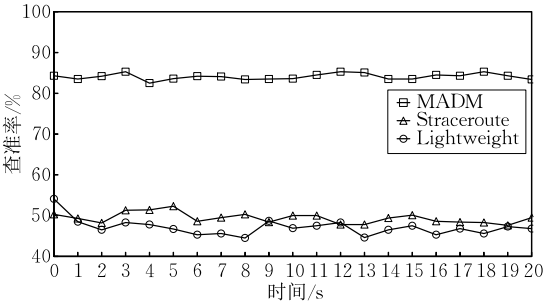


图 11　5.0%的网络丢包率

综合以上实验结果可知，与 Secure traceroute 和 Lightweight 检测机制相比，MADM 能够对不一致路径中的恶意节点进行有效检测，在查全率和查准率两个指标上都有优于另外两个机制的表现，即使在网络丢包率较高的情况下仍然可以有效检测恶意节点.

6 结束语

本文对域间路由路径不一致问题进行了深入研究,提出了一种不一致路径恶意自治系统检测机制MADM,弥补了已有工作的不足.MADM 利用路由证据将自治系统的路由行为与其自身相绑定,从而可以追究自治系统的路由行为.MADM 设计了不一致路径恶意节点的检测方法,源自治系统与其他自治系统比较路由证据,依据证据比较结果发现恶意节点.仿真结果表明,MADM 在查全率和查准率两个指标上的表现都优于 Secure traceroute 和 Lightweight,并适用于网络丢包率较高的环境.

下一步工作中,我们会研究当不一致路径包含多个恶意节点时,如何对多个恶意节点进行检测.同时本文仅介绍了不一致路径中恶意节点的检测方法,在未来工作中将研究如何对恶意节点实施路由控制,从根本上解决域间路由路径不一致问题.

参 考 文 献

[1] Rekhter Y, Li T, Hares S. A border gateway protocol 4. RFC 4271

[2] Deng W, Muhlbauer W, Yang Y, et al. Shedding light on the use of as relationships for path inference. Journal of Communications and Networks, 2012, 14(3): 336-345

[3] Boldyreva A, Lychev R. Provable security of S-BGP and other path vector protocols: Model, analysis and extensions//Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York, USA, 2012: 541-552

[4] Gürsun G, Ruchansky N, Terzi E, et al. Inferring visibility: Who's (not) talking to whom?//Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York, USA, 2012: 151-162

[5] Goldberg S, Schapira M, Hummon P, et al. How secure are secure interdomain routing protocols? Computer Networks, 2014, 70: 260-287

[6] Avramopoulos I, Rexford J. Stealth probing: Securing IP routing through data-plane security//Proceedings of the USENIX Annual Technical Conference. Boston, USA, 2006: 267-272

[7] Padmanabhan V N, Simon D R. Secure traceroute to detect faulty or malicious routing. SIGCOMM Computer Communication Review, 2003, 33: 77-82

[8] Naous J, Walfish M, Nicolosi A, et al. Verifying and enforcing network paths with Icing//Proceedings of the 7th Conference on Emerging Networking Experiments and Technologies (CoNEXT'11). New York, USA, 2011: 30:1-30:12

[9] Jian J, Junzhou L, Wei L, Yu L. Constraint conditions to eliminate AS incentive of lying in interdomain routing//Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management. Gentle, Belgium, 2013: 27-31

[10] Zhao M, Zhou W, Gurney A J, et al. Private and verifiable interdomain routing decisions//Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York, USA, 2012: 383-394

[11] Zhou W, Fei Q, Narayan A, et al. Secure network provenance //Proceedings of the 23rd ACM Symposium on Operating Systems Principles. New York, USA, 2011: 295-310

[12] Wong E L, Balasubramanian P, Alvisi L, et al. Truth in advertising: Lightweight verification of route integrity//Proceedings of the 26th Annual ACM Symposium on Principles of Distributed Computing (PODC'07). New York, USA, 2007: 147-56

[13] Gill P, Schapira M, Goldberg S. A survey of interdomain routing policies. ACM SIGCOMM Computer Communication Review, 2013, 44(1): 28-34

[14] Patel V, Mehta R. Data clustering: Integrating different distance measures with modified k-means algorithm//Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011), AISC 131. Roorkee, India, 2011: 691-700

[15] Wen Qiao-Yan, Niu Xin-Xin, Yang Yi-Xian. Boolean Function in the Modern Cryptography. Beijing: Science Press, 2000 (in Chinese)
(温巧燕,钮心忻,杨义先. 现代密码学中的布尔函数. 北京:科学出版社, 2000)

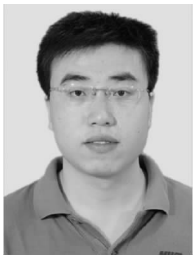
[16] Gill P, Schapira M, Goldberg S. Let the market drive deployment: A strategy for transitioning to BGP security//Proceedings of the ACM SIGCOMM 2011 Conference (SIGCOMM' 11). New York, USA, 2011: 14-25

[17] Jain S, Kumar A, Mandal S, Ong E. B4: Experience with a globally-deployed software defined WAN//Proceedings of the ACM SIGCOMM 2013 Conference (SIGCOMM'13). New York, USA, 2013: 3-14

[18] Li W, Xiong Q, Xu J. A peer-to-peer solution for sip servers cooperation in wireless mesh networks//Proceedings of the 2012 2nd International Conference on Consumer Electronics, Communications and Networks. Yichang, China, 2012: 1561-1564

[19] Vanbever L, Vissicchio S, Pelsser C, et al. Seamless network-wide IGP migrations. ACM SIGCOMM Computer Communication Review, 2011, 41(4): 314-325

[20] Magharei N, Rejaie R, Rimac I, et al. ISP-friendly live P2P streaming. IEEE/ACM Transactions on Networking, 2014, 22(1): 244-256



JIANG Jian, born in 1986, Ph. D. candidate. His research interest is network management.

LI Wei, born in 1978, Ph. D., associate professor. His research interests include next generation network architecture and service computing.

LUO Jun-Zhou, born in 1960, Ph. D., professor, Ph. D. supervisor. His research interests include next generation network architecture, protocol engineering, network security, cloud computing and service computing.

LU You, born in 1977, Ph. D. candidate, lecturer. His research interest is next generation network architecture.

XIA Nu, born in 1981, Ph. D. candidate. His research interest is next generation network architecture.

Background

This work is supported by the National Natural Science Foundation of China under Grant No. 61320106007, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2013AA013503, the China Specialized Research Fund for the Doctoral Program of Higher Education under Grant No. 20110092130002, the Prospective Research Project on Future Networks of Jiangsu Future Networks Innovation Institute under Grant No. BY2013095-2-07, the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant No. BM2003201, the Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under Grant No. 93K-9, the Ministry of Housing and Urban-Rural Development under Grant No. 2015-K6-012.

Interdomain routing is the core support technology of Internet. It consists of a control plane, where Autonomous Systems discover and establish paths with route announcements from neighbor Autonomous Systems, and a data plane, where they actually forward packets along these paths. BGP is a path-vector protocol used in the interdomain routing. In BGP, every AS makes its routing policy independently and applies its policy to choose the best path, and announce path information to its neighbors. Normally, the path used in the data plane and announced path in the control plane are consistent. However, BGP does not include any mechanism to enforce that announced path matches actual routing path. AS-level traceroute experiments indicate an inconsistency problem between announced path and actual routing path for at least 8% of Internet routes. Some research work such as SBGP (Secure Border Gateway Protocol) and SoBGP (Secure origin Border Gateway Protocol) were used to verify the route announcements from neighbors. These countermeasures guaranteed the authenticity of announced path in the control plane. However, they could not ensure that these paths are the same as the forwarding paths in the data plane. Much research work has also been devoted in the data plane to detect the inconsistent paths. They created an encrypted secure tunnel between end routers and sent probe packets to check the path. However, they needed the cryptographic operation on packets which incurred a high overhead. Furthermore, they could only detect the inconsistency path problem rather than the malicious AS which causes the inconsistency path problem. In this paper, we make an improvement on the detection mechanism that could detect the malicious AS in the path. With the help of routing evidence, source AS could discover the actual packets forwarding path of intermediate Autonomous Systems in the path. Autonomous Systems in the path collaborate with each other to locate malicious AS finally. The experiment results show that our mechanism has a better performance on malicious AS detection than existing methods.