计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(2):366−375 [doi: 10.15888/j.cnki.csa.008389]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于事实所有权的 RPKI 缓存更新冲突检测机制①

肖文龙[1,2], 马　迪[1,2,3], 毛　伟[2,3], 邵　晴[3]

[1](中国科学院 计算机网络信息中心, 北京 100190)
[2](中国科学院大学, 北京 100049)
[3](互联网域名系统国家地方联合工程研究中心, 北京, 100190)
通信作者: 肖文龙, E-mail: xwl0012@163.com

摘　要: 随着 RPKI 覆盖的域间网络的范围不断扩大, RPKI 在实际部署中的数据同步一致性的问题, 运维失误和权威机构权力滥用的风险已成为影响 RPKI 全面部署的主要障碍. 本文提出了一种基于事实所有权的 RPKI 缓存更新冲突检测机制. 该机制利用反向 RTR 协议与 RPKI 数据层级分发架构进行事实路由起源信息的采集与同步, 并通过比较事实路由起源信息与 RPKI 缓存更新数据检测出冲突的 RPKI 缓存更新数据, 保护了 RPKI 缓存的真实有效. 最后, 本文就该机制的数据同步时间效率和检测性能同其他方案进行了对比, 实验结果表明本方案有一定的检出优势.

关键词: 资源公钥基础设施; 事实所有权; 路由起源信息; 冲突检测; 缓存更新

引用格式: 肖文龙,马迪,毛伟,邵晴.基于事实所有权的 RPKI 缓存更新冲突检测机制.计算机系统应用,2022,31(2):366–375. http://www.c-s-a.org.cn/1003-3254/8389.html

Fact Ownership-based Conflict Detection Scheme for RPKI Cache Update

XIAO Wen-Long[1,2], MA Di[1,2,3], MAO Wei[2,3], SHAO Qing[3]

[1](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)
[3](Internet Domain Name System Beijing Engineering Research Center, Beijing 100190, China)

Abstract: As the resource public key infrastructure (RPKI) coverage of the inter-domain network expands, the consistency of RPKI data synchronization in the actual deployment, the risk of operational errors and abuse of authority power have become major obstacles to the full deployment of RPKI. This study presents a scheme for detecting conflicts of updating RPKI cache based on fact ownership of route origin. This scheme uses reverse RTR protocol and multi-layer transmission architecture of RPKI data to collect and synchronize fact route origin information. Then, it compares fact route origin information and RPKI cache update data to detect conflicting data of RPKI cache update, which ensures authenticity and effectiveness of RPKI cache. Finally, the data synchronization efficiency and detection performance of this scheme are compared with those of other schemes. The experimental results show that this scheme has some detection advantages.

Key words: resource public key infrastructure (RPKI); fact ownership; route origin information; conflict detection; cache update

1　背景

随着全球互联网规模不断扩大, 各个 AS (autonomous system, 自治系统) 间的流量转发也愈发频繁. 而不同 AS 之间能够互联互通的关键就是 BGP (border gateway protocol, 边界网关协议) 协议[1]. 但其基于互联网安全可信原则, 没有提供保障 BGP 消息真实性

① 收稿时间: 2021-04-19; 修改时间: 2021-05-19, 2021-07-01; 采用时间: 2021-07-13; csa 在线出版时间: 2022-01-17

和有效性等的安全机制, 致使它很容易因网络运维者的配置失误或恶意攻击的影响, 引发严重的 BGP 路由泄露与劫持事件, 严重威胁全球互联网的安全与稳定.

有鉴于此, 业界与学术界着手制定替代 BGP 的安全域间路由方案. 其中应用最为广泛的是 IETF (Internet Engineering Task Force, 互联网工程任务组) 提出的通过 RPKI (resource public key infrastructure, 资源公钥基础设施)[2] 来进行的 BGP 路由起源验证的安全方案, 即 ROV (route origin validation, 路由起源验证)[2].

1.1 RPKI 体系工作机制

RPKI 体系的核心设计思想是通过构建一套层次结构的 PKI (public key infrastructure, 公钥基础设施) 实现对 INR (Internet number resource, 互联网码号资源) 所有权的分配和验证[3]. 如图 1 中所示, RPKI 是以 RIR (regional internet registry, 区域互联网注册机构) 为单一信任锚点的层级信任模型. 该模型通过逐层签发证书, 构建出从根证书到最低层次的 CA (certificate authority, 证书颁发机构) 证书的信任链, 使得 RP (relaying partiy, 依赖方) 从统一的根证书就可以验证任意 CA 证书[3]. 此外, 从图 1 可知 RPKI 体系对于 IP 地址前缀与路由起源分配关系的签发和验证并不依赖 BGP 协议, 这最大程度的降低了在 BGP 上广泛实施 ROV 时, 对全局 BGP 路由收敛速度的不利影响.

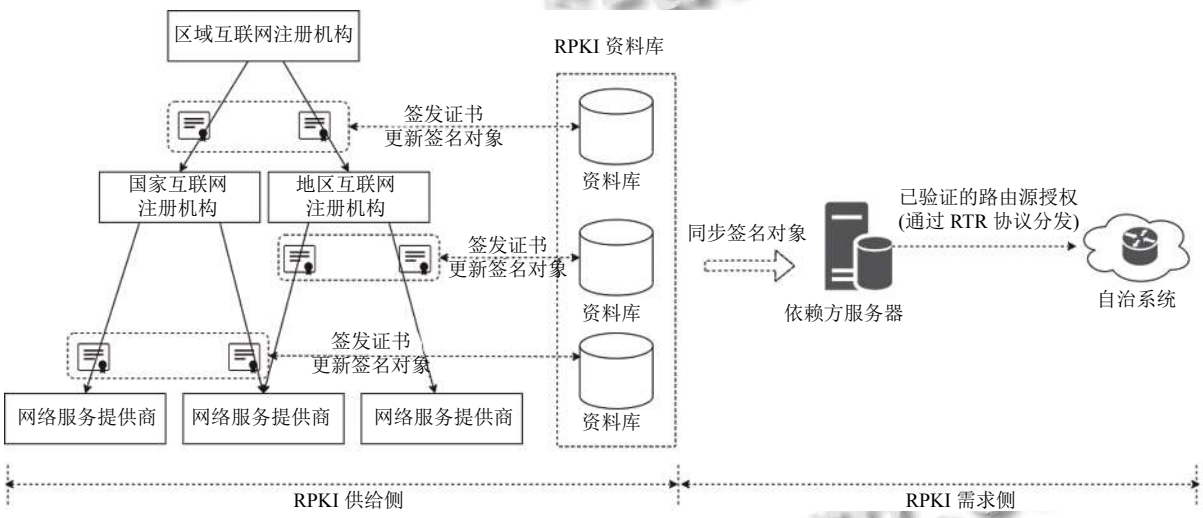

图 1 RPKI 基本架构的工作机制

1.2 RPKI 全面部署面临的障碍

RPKI 带外验证的优势使其迅速成为实施 ROV 的主流技术, 但至今为止 RPKI 覆盖的 IP 地址空间占比仍然较小, 实现全面部署依然漫长. 根据来自 ICANN 的最新报告[4] 以及 APNIC (Asia Pacific Network Information Centre, 亚太网络信息中心) 的研究[5] 表明 RPKI 全面部署主要面临如下几个障碍:

(1) 数据同步一致性问题;

(2) 供给侧运维失误风险;

(3) 5 大 RIR 的权利滥用风险.

1.2.1 数据同步非一致性问题

由于各个 RP 服务器与 RPKI 资料库进行数据同步及其验证是独立的, 各个 RP 之间的数据同步, 验证并不会相互影响. 这使得各个 RP 的本地数据不一致, 这将致使其输出至各个 AS 的有效路由授权与全局实际有效路由信息发生冲突, 最终导致不同 AS 之间存在路由信息冲突.

1.2.2 供给侧运维失误风险

目前 RPKI 的 CA 证书资料库依然是通过网络管理人员进行运维, 难免会发生一些人为失误, 主要的失误为: CA 资料库数据更新不及时. ISP (Internet service provider, 网络服务提供商) 以授权模型部署 RPKI 时, 其被上级资源持有机构授权自行管理被分配的 IP 地址前缀资源的 RPKI 签名对象. ICANN 指出一些小型 ISP 对于其 CA 资料库的数据运维不及时或是放弃运维, 导致资料库中的 RPKI 签名对象文件过期[3]. 这将导致其他部署 RPKI 的 AS 因该 ISP 签发的 ROA (route origin authorization, 路由起源授权) 证书过期而将其验

证为无效.

1.2.3 5 大 RIR 的权利滥用风险

(1) 单一撤销机制

RPKI 证书颁发机构通过签发 CRL (certificate revocation list, 证书撤销列表), 对未过期的有效证书的撤销, 并且这种撤销操作是单方面的, 即无需要证书的私钥持有者许可. 文献 [6] 指出这种权力的失衡使 IP 前缀资源的使用者暴露在随时可能被剥夺合法网络使用权的风险下.

(2) 隐性撤销

RPKI 权威机构亦可以通过不发布绑定了 IP 前缀使用者的签名对象文件, 或是拒绝 RP 访问该签名对象文件, 来达到 IP 前缀与使用者的授权关系实际不生效的效果[6].

(3) 资源覆盖与重写

RPKI 权威机构将已分配给下级资源持有者的前缀区块重新签发给其他自治系统, 亦或通过签发新 ROA 覆盖未部署 RPKI 的事实有效前缀, 这些情况的发生都将影响 BGP 流量的正常转发.

1.3 RPKI 缓存更新风险防范方案研究现状

由于上述风险与问题将直接影响 RP 对 RPKI 缓存更新结果的正确性. 因此学者们提出了在 CA 侧对 CA 恶意或误操作的预防和在 RP 侧的错误检测的方案.

刘晓伟等人提出建立 CA 侧 CA 资源分配误操作的检测方案[7], 以防止 CA 跨级的重复分配等误操作行为. 该方案无需搭建另外的信任机制, 仅通过规范 CA 的资源分配行为, 提高了 CA 的资源分配的准确性, 降低了供给侧运维失误风险. 但是无法预防 CA 恶意的权力滥用行为. Xing 等人利用区块链的去中心化的信任模型优势提出了 BGPCoin 方案[8,9]. BGPCoin 通过区块链公共账本记录网络资源所有权的实时变化, 同时区块链的公开、不可篡改的特性使得 CA 对网络资源所有权的分配更加透明和可追溯. 但是 BGPCoin 记账的时间、空间开销较大, 其在实际网络中的性能是否满足实际需求仍然有待验证. Shrishak 等人提出通过门限签名算法使得每个 RPKI 证书的签发与撤销都需要 5 大 RIR 的协商同意, 从而限制了单个 RIR 的绝对权力[10]. 其性能表现在 5 大 RIR 的门限签名模型中比基于区块链得方案更加优秀, 但是为了保证 CA 行为受到严格的限制, 则需要赋予更多方参与门限签名, 这使得该方案的性能表现大大下降. Kent 等人提出当 RP 因无法获取原有证书文件而需要进行缓存删除时, RP 可以暂停一段时间对此缓存删除, 并允许 RP 使用原文件进行更新, 随后核实该操作确为所需后再删除[11]. 该方案的优势在于几乎不需要投入额外的基础设施的建设, 但是该方案难以准确设定延迟操作的时间, 时间过短无法保证能够恢复受影响的数据, 时间太长降低了缓存更新的时效性, 此外对于外部核实步骤亦没有明确的规则, 更是加大了缓存更新的复杂性. Heilman 等人提出资源使用者使用 RPKI 的资源证书签发该资源的.dead 对象来表示同意权威方对该资源的撤销, 以此来防止未经许可的撤销或覆盖发生[12]. 该方案在现有的 RPKI 机制中引入了对资源持有者的权益保障, 对现有的 RPKI 机制具有较好的兼容性. 但该方案中需要对.dead 签名文件的同步与验证, 同样会因为各 RP 同步.dead 文件时间不一致, 导致数据不一致问题. 此外, 通过签发同意撤销的.dead 文件的方式, 消耗了 RP 额外的计算资源.

1.4 基于事实所有权的 RPKI 缓存更新冲突检测原理

1.4.1 事实所有权原理

本文的路由起源信息的事实所有权源于 Hlavacek 等人提出的路由起源信息事实所有权的验证系统——DISCO[13]. DISCO 系统通过验证路由起源信息事实所有权来确定自治系统与网络资源的合法绑定关系. 由 AS_i 在 $DISCO_{Registrar}$ 中申请资源绑定证书及其私钥 (pk, sk), 并在向外宣告的 BGP_{ad} 消息的属性字段中附加由该资源证书私钥的签名 $Sign_{sk}(AS_i, IPPrefix)$. 当 DISCO 部署在实际网络中 $n \geqslant N_{Threshold}$ 个有利观测点 $DISCO_{vp}$ 在一定时间间隔 T 内接收到携带于该证书签发的资源绑定关系一致的 BGP_{ad} 消息时, DISCO 判定其路由源对其 IP 地址前缀是享有事实所有权的. 其主要过程描述如下:

(1) $AS_i(pk, sk) \leftarrow DISCO_{Registrar}$

(2) $\{AS_1, \cdots, AS_{neighbor}\} \leftarrow BGP_{ad}(Sign_{sk}(AS_i, IPPrefix))$

(3) $DISCO_{vp}$: $Signed_{AS_{sk}}(BGP_{ad}) \leftarrow \{AS_1, \cdots, AS_n\}$

(4) $DISCO:(AS_i, IPPrefix) \overset{t \leqslant T}{\leftarrow} \{DISCO_{VP1}, \cdots, DISCO_{VPn}\}$

(5) $n \geqslant N_{Threshold} \Rightarrow (AS_i, IPPrefix)_{valid}$

由此可知, 事实所有权是指域间路由系统中其他自治域对于某个自治域宣称对特定 IP 前缀资源所有权声明的接受或认可. 具体体现为自治域向其他自治域通告路由起源消息更新时, 该条 BGP 更新可以通过其他自治域的本地路由通告过滤策略被自治域 BGP

路由器保存至路由转发信息表中. 且该条路由起源在生命周期内 (生命周期指路由起源信息从被其他路由器接收的时间接收到至原自治域撤销的时间段) 内不会被域间路由系统视为异常路由信息, 而被其他路由器删除或忽略.

1.4.2 BGP 路由起源信息的稳定性

BGP 路由劫持者发起的非法的路由起源信息同样可能被全球 BGP 路由器接收, 如果仅凭大部分路由器是否接收该路由起源信息作为路由源合法的依据, 将可能把非法路由起源信息错误识别为合法的. 另一方面, 非法的路由起源信息并不会在全局 BGP 路由系统中长期存在.

本文分析了正常情况下路由起源信息 (即路由源与 IP 地址资源的分配关系) 在网络中存续时间的分布规律. 数据源来自于比较权威的 RIPE (regional internet registry for Europe, 欧洲互联网注册机构) 的 rrc13 采集器的 RIB (route information base, 路由信息表) 数据, 采集范围为协调世界时间 2020 年 10 月 17 日 0 时 0 分至 2020 年 10 月 31 日 0 时 0 分. 结果如图 2 所示, 纵坐标 BGP 路由前缀资源分配关系变化率表示路由源与 IP 地址前缀的分配关系在存活时间内已改变占总数比, 稳定存在时间小于等于 8 h的占 0.29%, 路由源与 IP 地址资源的分配关系稳定存在时间大于 272 h 的占 97.53%.

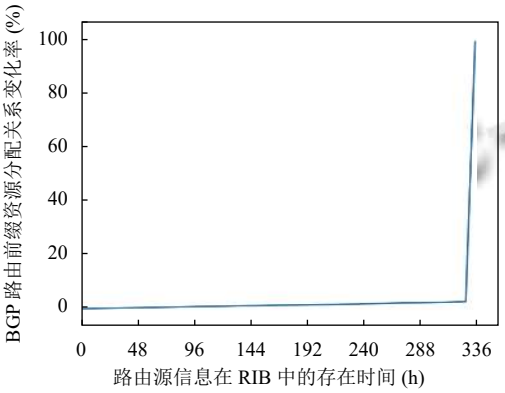

图 2 路由起源信息存在时间的累积分布图

如图 2 所示, 在较长的时间内, 绝大部分的路由源与 IP 地址资源的分配关系是稳定不变的, 只有极少部分的路由源与 IP 地址资源的分配关系是存在变化的. 那么依靠合法路由起源信息的稳定性特征, 我们可以通过设置观察期来观察一个新增的路由起源信息是否是稳定存在的, 进而防止短时存在的非法路由起源信息被误识别为事实存在路由起源信息.

因此, 本文结合路由起源信息的事实所有权原理及其稳定性特征, 提出了一种基于事实所有权的 RPKI 缓存更新冲突检测机制. 下面对本方案的设计与实现进行详细阐述.

2 基于事实所有权的 RPKI 缓存更新冲突检测机制设计

2.1 总体方案架构

BGP 网络里 AS 之间部署的路由器是独立运行的, 全局的事实路由起源信息同步结构为 P2P (peer to peer, 点对点) 型. 这种 P2P 结构的域间信息同步数据交互次数复杂度为 n^2. 如图 3 所示, 文献 [14] 通过在 RP 服务器与 AS 之间构建多层级 VC (validation cache, 有效缓存) 服务器向不同 AS 进行 RPKI 数据分发, 以此将各个独立 AS 规划在统一的管理域内, 保证了 RPKI 数据的一致性, 而且将全球 RP 同步次数复杂度降低为 n. 此外, 依托于此架构与 DISCO 事实所有权验证方案, 可以用底层 VC_i 服务器代替 DISCO 观测点 $DISCO_{vp}$, 顶层 VC_{top} 服务器代替 $DISCO_{Registrar}$, AS_j 接入时上传其 RC_j, ROA_j 至底层 VC_j, 故本方案的事实所有权验证过程如下:

(1) $VC_j \leftarrow AS_j(RC_j, ROA_j)$

(2) $\{AS_1, \cdots, AS_n\} \leftarrow AS_j.BGP_{ad}$

(3) VC_i: $AS_j.BGP_{ad} \leftarrow \{AS_1, \cdots, AS_n\}$

(4) VC_{top}: $(AS_j, IPPrefix) \overset{t\leqslant T}{\leftarrow} \{VC_1, \cdots, VC_n\}$

(5) $n \geqslant N_{Threshold} \Rightarrow (AS_j, IPPrefix)_{valid}$

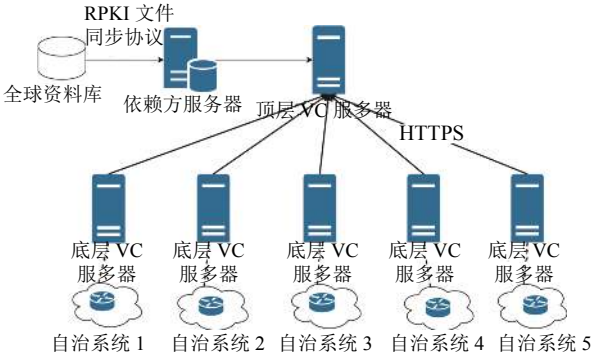

图 3 事实路由起源信息同步架构

基于对 AS 事实权益的保障, 在 AS 对网络资源事实所有权的存续期间对引发其绑定关系正常使用的

RPKI 缓存更新应该视为冲突缓存, 应停止将其更新进入路由器中.

所以, 基于事实所有权的 RPKI 缓存更新冲突检测机制分为如下 3 个阶段, 一是利用底层 VC 服务器同路由器进行本地事实路由起源信息的同步; 二是, 底层 VC 服务器将改变的本地同步数据发送至顶层 VC 服务器进行全局路由起源数据汇总; 三是, 利用全局路由起源信息表对缓存更新进行冲突检测, 排除对表内的路由源的冲突更新. 本文在下文中分别对这 3 个阶段的方案进行了设计.

2.2 本地事实路由起源信息同步方案设计

本文使用反向 RTR 协议进行本地事实路由起源信息同步, 即通过新增 RTR[15] 协议负载类型, 使原有 RTR 的只有从 RP 侧向 BGP 路由器传输的单一方向的数据同步, 增加从 BGP 路由器侧至 RP 侧传输 ROI (router origin information, 路由起源信息) 的反向 RIB 数据同步. 当 RIB_{Local} 发生 ROIs {ROI_1(updateStyle, AS_n, IPPrefix), ROI_2, …, ROI_n}更新或 AS_i 的路由源授权证书 AS_i.ROA 更新时, 通过反向 RTR 协议发送序列号 $AS_i.Notify_n.SN$ 为 n 的 $AS_i.Notify_n$ 开启与 VC_i 本地 ROI 数据库 VC_i.RoiLocal 的 ROIs 数据同步或 (RC_i, ROA_i) 更新, 同步过程参照 RTR 协议设计, 具体同步流程如流程 1 所示.

流程 1. 本地事实路由起源信息同步流程

```
1) while RIB_Local‖(RC_i, ROA_i) update|AS_i ← VC_i.Reset do
2)   case: (RC_i, ROA_i) update
3)     VC_i.RC_i = AS_i.RC; VC_i.ROA_i =AS_i.ROA_i
4)   case: VC_i ← AS_i.Notify_n
5)     if VC_i.RLSN == AS_i.Notify_n.SN-1
6)       VC_i.RoiLocal = VC_i.RoiLocal+ROIs; VC_i.RLSN++
7)     else
8)       AS_i ← VC_i.Reset
9)     end if
10)  case: AS_i ← VC_i.Reset
11)    VC_i ← AS_i.RoiLocal
12)    VC_i.RoiTable = AS_i.RoiLocal;
       VC_i.RoiTable.SN= AS_i.RoiLocal.SN
13) end while
```

2.3 全局事实路由起源信息同步方案设计

如图 3 所示, 本文通过部署在 AS 的 VC 服务器利用反向 RTR 协议实时采集 BGP 路由起源信息, 建立本地路由起源信息数据库 VC_i.RoiLocal. 另外, 通过底层 VC_i 服务器记录下同步更新 ROI 及其接收时间, 再由顶层 VC_{top} 服务器进行汇总统计, 最终 VC_{top} 将满足预设时间间隔 T 的 $ROIs_j$ 更新至 VC_{top}.RoiGlobal, 将不满足时间间隔 T 的 $ROIs_j$ 更新至 VC_{top}.RoiRecv, 并将本次全局更新结果 VC_{top}.RoiGlobalIncr 反馈至底层 VC_i 完成一次 VC_{top}.RoiGlobal 的更新. 具体流程如流程 2.

流程 2. 全局事实路由起源信息同步流程

```
1) while VC_i ← VC_top.RequestSN‖VC_top ← VC_i.Reset‖ VC_i←VC_top.RoiGlobalIncr
2)     case: VC_i ← VC_top.RequestSN
3)       if VC_top.RequestSN == VC_i.RGSN+1
4)         for j=1; j≤VC_i.ROIs.Length;j++ do
5)         if VC_i.ROIs_j.LifeTime ≥ T
6)           VC_top.RoiGlobal = VC_top.RoiGlobal +VC_i.ROIs_j
7)         else
8)           VC_top.RoiRecv = VC_top.RoiRecv+VC_i.ROIs_j
9)         end if
10)      end for
11)      VC_top.RoiGlobal.SN++; VC_i←VC_top.RoiGlobalIncr
12)    else
13)      VC_top ← VC_i.Reset
14)  case: VC_i←VC_top.RoiGlobalIncr
15)      VC_i.RoiGlobal = VC_i.RoiGlobal+VC_top.RoiGlobalIncr;
VC_i.RoiGlobal.SN++
16)  case: VC_top ← VC_i.Reset
17)    VC_i ← VC_top.RoiGlobal
18)    VC_i.RoiGlobal = VC_top.RoiGlobal; VC_i. RoiGlobal.SN=
VC_top.RoiGlobal.SN
19) end while
```

2.4 基于事实所有权的 RPKI 缓存更新冲突检测方案

通过对第 1 节中 RPKI 全面部署面临的障碍的分析, 本文将 RPKI 缓存更新与实际域间网络起源信息的冲突总结为以下 4 种.

(1) 因 RPKI 证书维护不及时导致过期失效冲突.

(2) CA 失误撤销 RPKI 证书导致证书无效冲突.

(3) 资料库访问失败, 原有证书过期失效冲突.

(4) CA 跨级签发已分配的 IP 前缀的子前缀冲突.

因此, 当顶层 VC 服务器接收 RPKI 缓存更新时, 应提取撤销和新增操作的 RPKI 缓存更新分别与最新的全局路由起源信息表核对, 校验是否存在撤销当前仍然存在于网络中的路由起源信息, 并输出存在冲突的 RPKI 缓存更新.

2.4.1 RPKI 缓存撤销更新冲突检测

在对 ROA_i 与当前全局路由起源信息核对过程中, 首先需要查找出与该 ROA_i 相关的 ROI. 虽然在当前 ROA 中 IP 前缀资源绑定大多是某条特定的前缀, 但也

存在部分的是 IP 前缀区块 (例如, {AS_n:3, IPPrefix: 103.134.63/22, maxLength: 24}), 故本前缀匹配函数 ROILPM(ROA_i) 输出的是对 ROA_i 前缀长度匹配的最长前缀, 及其 maxLength 划定的区块中包含的全部 ROI 前缀. 另外, 当匹配的 ROI_j 与撤销的 ROA_i 相同或为其区块内子前缀的资源分配关系, 则此撤销更新 ROA_i 为无效. 具体算法如算法 1.

算法 1. RPKI 缓存撤销更新冲突检测

1) 输入: 撤销缓存更新 ROA_i.
2) 输出: 本次检测结果 R_i.
3) 初始化: R_i=valid
4) ROI = ROILPM(ROA_i)
5) for j=1; j≤ROI.Length;j++ do
6) if (ROI_j.IPPrefix ϵ ROA_i.IPPrefixBlock)||(ROI_j.IPPrefix = = ROA_i.IPPrefix)
7) if ($ROI_j.AS_n$==$ROA_i.AS_n$)
8) R_i=invalid; break
9) end if
10) end if
11) end for

2.4.2 RPKI 缓存新增更新冲突检测

ROILPM 函数输出的 ROI_j 同上, 当 ROI_j 为其区块内子前缀或相同前缀, 但资源分配关系不一致时, 则此新增更新 ROA_i 为无效. 另外 ROI 为 IP 前缀与 AS_n 的绑定关系, 为防止出现 ROI_j 所属的 ROA_j 区块与检测的 ROA_i 存在交叉覆盖, 故当该 ROA_j 与被检测的 ROA_i 存在子前缀交集时, 其也为无效. 具体算法如算法 2.

算法 2. RPKI 缓存新增更新冲突检测

1) 输入: 新增的缓存更新 ROA_i.
2) 输出: 本次检测结果 R_i.
3) 初始化: R_i=valid
4) ROI =ROI LPM(ROA_i)
5) for j=1; j≤ROI.Length;j++ do
6) if (ROI_j.IPPrefix∈ROA_i.IPPrefixBlock)&&($ROA_i.AS_n$≠$ROI_j.AS_n$)
7) R_i=invalid
8) end if
9) if (ROA_i.IPPrefix == ROI_j.IPPrefix) &&($ROA_i.AS_n$ ≠ $ROI_j.AS_n$)
10) R_i=invalid
11) end if
12) if ROA_j∩ROA_i
13) R_i=invalid
14) end if
15) end for

2.4.3 实际部署中的方案优化

此外, 在实际 BGP 网络的运行中还需要针对 BGP 网络与 RPKI 的运维特性对本方案的相关参数及判定进行相应优化.

(1) 时间间隔参数 T

在实际 BGP 网络中 BGP 通告全局传播存在一定的时延, 即其他 AS 收到的 BGP 通告存活时间相较于真实存活少了时延 T_s, 因此对于各个 AS 收到通告的存活时间 T 应该加上传播时延 T_s, 而一般 BGP 消息的全局传播时延 T_s 在 38 s 至 2 min 之间[16].

(2) 全局快速撤销与未在网使用撤销确认

如图 4 所示, 正常情况下 AS 的 BGP 路由器停用前缀, 会先主动断开 BGP 连接, 如有必要在 RPKI 中撤销该前缀, 还会进行请求 CA 撤销该 ROA. 当全局 VC 绝大多数都收到一个 BGP 撤销消息时, 此时不必等待时间间隔 T, 立即确认全局 ROI 撤销事实成立. 另一方面, ISP 会签发少量特定用途的 ROA, 比如用在流量工程中, 这些 ROA 签发后并不会立即启用, 故针对此类 ROA 的撤销应该通过 VC_{top} 与该 AS 相连的 VC 进行撤销确认.

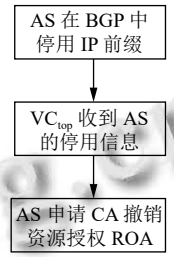

图 4 RPKI 下 AS 对前缀资源停用过程

(3) 新增快速确认

如图 5 所示, 在部署了 RPKI 的 AS 每当启用新的前缀资源时需要先签发 ROA, 即使冲突检测为有效, 但此时 AS 与该前缀绑定关系存在时间小于预设 T, 此期间对其发起的错误撤销则无法检测出, 故当此 ROA 冲突检测为有效且新增的 ROA 与 VC_{top} 接收的 ROA 相同时, 应立即确认其新增事实成立.

3 实验与分析

为了验证方案的可行性和性能表现, 本文配置了 1 台服务器分别对全局路由起源信息的同步和 RPKI 缓存更新冲突检测的数据同步性能进行了测试, 实验所用服务器配置如表 1.

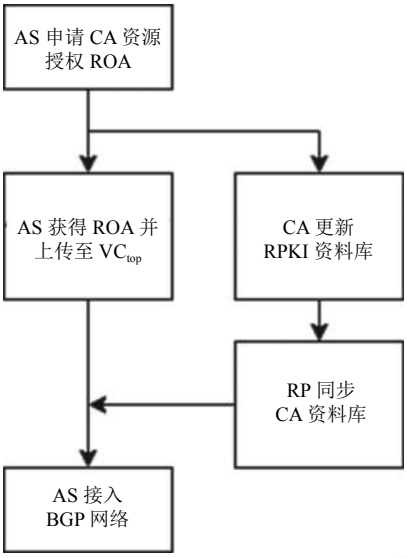

图 5 RPKI 下 AS 对前缀资源启用过程

表 1 服务器配置

操作系统	CPU	内存
CentOS 8.2	Intel Xeon E3-1220	24 GB

3.1 全局路由起源信息同步实验

全局路由起源信息同步实验中各模块数据流如图 6 所示, 底层 VC 服务器通过反向 RTR 来采集路由起源更新信息, 并且在解析, 校验反向 RTR PDU 数据包和请求更新的序列号无误后, 进行更新本地路由起源信息表. 底层 VC 服务器定期向汇聚层发送对全局路由起源信息表的路由起源更新信息. 汇聚层 VC 服务器校验更新序列号无误后, 进行更新信息汇总, 统计每条更新信息提交的 AS 数, 统计完毕后将汇聚结果发送至顶层 VC 服务器进行全局同步.

实验使用 1 号物理机 22 229 端口模拟底层 VC 服务器进行本地同步和底层全局同步, 2181 端口模拟汇聚层 VC 服务器进行汇聚处理, 33 339 端口作为顶层 VC 服务器进行顶层全局路由起源信息同步. 并且通过向 1 号物理机的汇聚层服务器的路由起源消息接收队列中分别并行发送 (2 000 个, 4 000 个, 6 000 个, 8 000 个, 10 000 个) 路由源数据包, 每个数据包含有 100 条路由起源更新信息, 以此模拟不同的 AS 接入规模, 并记录同步花费的时间 (单位: s).

由图 7 可知, 随着接入的底层 VC 服务器增多, 本方案的全局路由起源信息同步时间也将越来越长.

3.2 冲突检测分析

3.2.1 效率分析

本文依照文献 [12] 的全网部署的假设条件, 在实际 RPKI 数据中随机选择 5 组不同规模的 AS 组成的 5 个规模的域间网络. 分别测试了不同规模下单节点进行. dead 对象和全量全局路由信息数据同步的时间消耗. 如图 8 所示, 横坐标表示网络中 AS 的数量, 纵坐标为同步耗时 (单位: s).

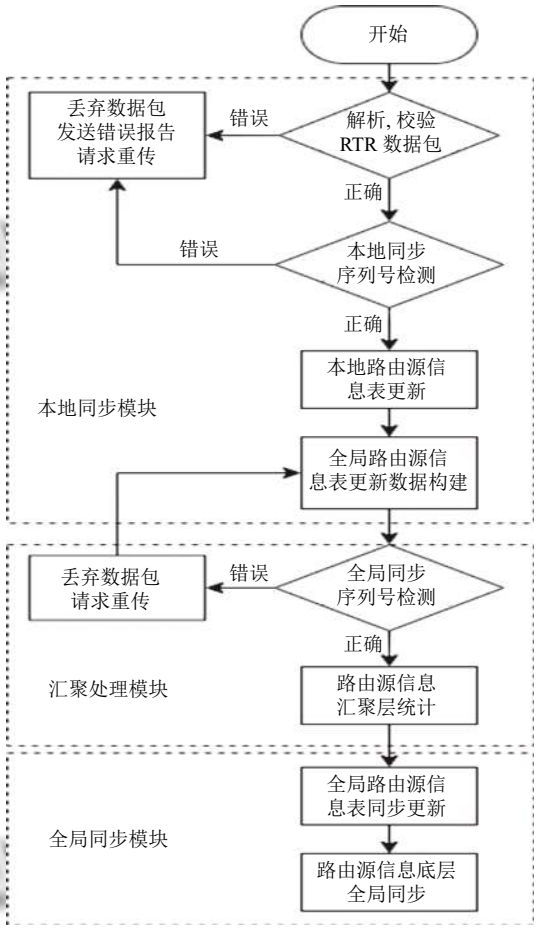

图 6 同步实验流程图

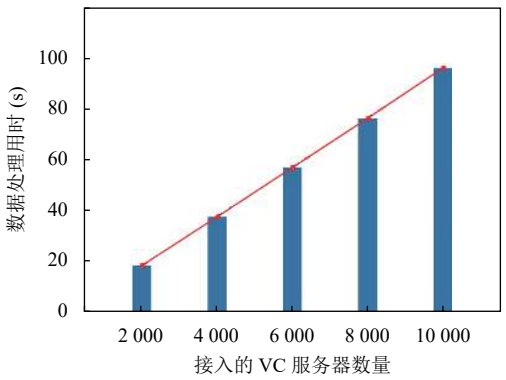

图 7 全局路由起源信息同步数据处理时间图

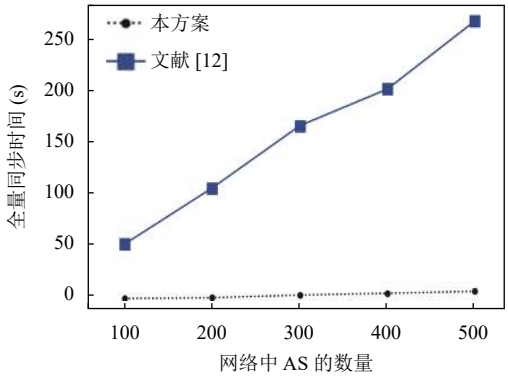

图 8 全局路由起源信息同步时间效率对比图

本方案不仅在数据的全量同步时间消耗上表现更佳, 而且部署在服务器上的全量数据所占用的存储空间也更少. 对比结果如图 9 所示.

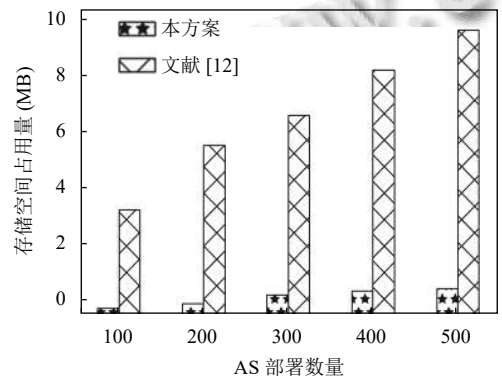

图 9 文件空间存储占用对比图

文献 [12] 的方案需要每个 AS 对来自不同的上级 CA 签发的证书, 动态维护一个包含同意删除的资源集合签名.dead 对象, 则每个 AS 至少维护一个.dead 资料库. 并且 RP 需要对其进行同步和链式验证, 不同于 ROA 的验证的是.dead 的验证是逆向的, 即叶子节点的.dead 对象只需验证叶子证书的有效性, 中间 CA 签发的 .dead 对象则需验证其包含的全部子资源的.dead 对象. 因此, 如图 8 所示, 虽然其需要签发的文件少于 ROA, 但对其签名文件的同步验证依然将消耗比本方案更多的计算时间. 同时, 随着部署率上升对于部署文献 [12] 方案的存储消耗也远超本方案.

3.2.2 检出性能分析

本文基于文献 [12] 中的假设, 即在网络全部部署 RPKI, 各个 AS 与 RP 行为是诚实可信且公钥密码体系无法破解的条件下, 仅就两方案对于由权威方过失行为导致的冲突缓存的检测性能进行分析. 实验数据取自 6 月 4 日的 RPKI 数据和实际长期存活的 BGP 数据, 随机选取的 100 个 AS 对应的前缀分配数据, 并就 3 种冲突情况分别进行了模拟. 在本方案和文献 [12] 的部署率下降时的冲突检出率进行了对比. 实验结果如下:

(1) CA 单边撤销证书与资料库证书异常过期失效

撤销更新的检测实验模拟的数据分为两类, 一类是对已在网络中使用 BGP 前缀的 ROA 进行单边撤销 (异常过期失效), 另一类是对已签发 ROA 但是 ROA 的前缀未在 BGP 中使用的单边撤销 (异常过期失效), 两类数据量占比相等.

如图 10 所示, 本方案的检出率下降速度慢于文献 [12] 方案. 因为本方案对全局 ROI 的采集率不会随着部署率的下降而下降, 这使得其在部署率下降的情况下仍能保护网络中在使用的 BGP 前缀. 另一方面, 对于未在现网中使用的 ROA 的撤销依赖于 AS 对 VC_{top} 的撤销确认机制, 此类情况下同文献 [12] 一样检出率等同于方案的部署率. 故在相同部署率下本方案的综合检出率更高.

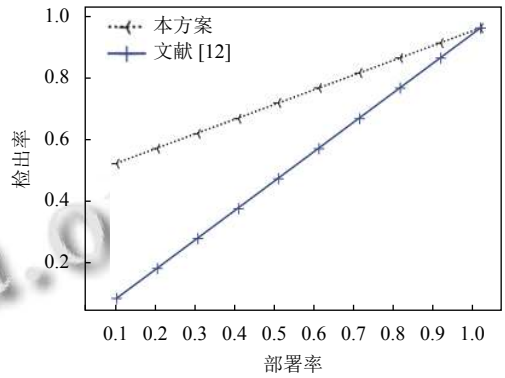

图 10 撤销更新冲突综合检出率对比图

(2) 资源重写冲突检测

资源重写冲突检测实验模拟的数据分为两类, 一大类是对已在网络中使用 BGP 前缀的 ROA 重写, 其中重写的类型亦分为全部重写 (ROA 包含的全部前缀区块) 与部分重写 (ROA 包含的部分前缀区块). 另一大类是对 ROA 已签发但是其前缀未在 BGP 中使用的 ROA 重写, 其中重写的类型同上. 各类数据量比例相等.

如图 11 所示, 对于已在网使用的 ROA, 本方案可以不受部署率限制获取网络中的全局 ROI, 但是

CA 可以部分重写 ROA 区块, 避开与 ROI 存在冲突的部分. 文献 [12] 则需要通过发布的.dead 文件, 才能对重写冲突进行校验. 故在相同部署率下本方案的检出率更高.

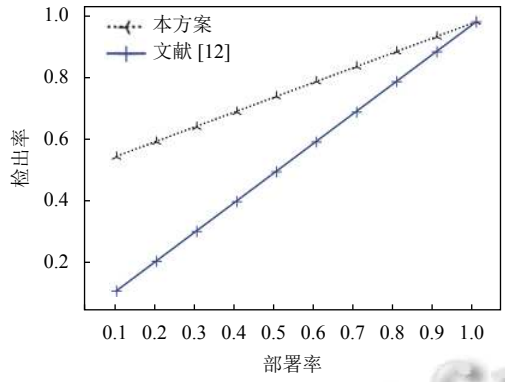

图 11 ROA 已在网使用的重写更新冲突检出对比图

如图 12 所示, 对于未在网使用的 ROA, 本方案无法获取在网使用的 ROI, 只能通过 AS 上传至 VC 的 ROA 进行重写检测, 检出性能受部署率变化与文献 [12] 相同故在相同部署率下本方案的检出率无优势.

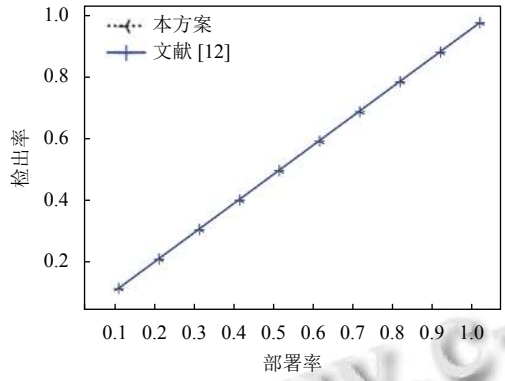

图 12 ROA 未在网使用的重写更新冲突检出对比图

(3) 资源覆盖冲突检测

资源覆盖的冲突检测实验针对未被 ROA 数据覆盖但已在网使用的 IP 前缀, 故 RPKI 的部署率始终为 0. 即仅依靠 ROI 或签发.dead 对象能够检出资源覆盖冲突的比率.

如图 13 所示, 对于已在网使用的 IP 前缀, 虽然本方案检出性能不受部署率变化的影响, 但是本方案事实所有权路由源判定模型属于多方共识决策模型, 故实际部署中参与决策的节点 (部署的节点) 的数量比例不应太低.

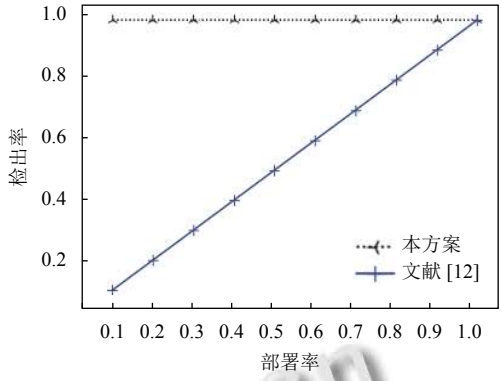

图 13 ROA 覆盖冲突检出对比图

(4) RPKI 数据一致性问题分析

文献 [12] 描述了一种镜像世界的数据不一致情况, 在这种情况下两个资源持有者因同步的时间先后, 产生完全不同的 RPKI 视野, 该种情况下文献 [12] 方案无法通过验证.dead 文件保持全局 RPKI 视野一致. 而本方案的全局路由起源信息表是严格序列全局同步的, 全局各节点处在统一的事实有效路由信息和 RPKI 缓存视图.

4 结束语

随着部署 RPKI 进行 ROV 的网络运营商的不断增多, RPKI 在域间路由系统实际部署中的存在的问题和风险也愈加凸显. 在此背景下, 本文通过分析 RPKI 进入全面部署的问题和风险, 定位阻碍 RPKI 全面部署的根源所在, 并提出基于事实 BGP 路由起源信息的原理和路由起源信息的稳定性, 实现了一种基于事实所有权的 RPKI 缓存更新冲突检测机制. 最后模拟了 5 种不同规模的 BGP 网络, 实验对比了本方案与现有的其它方案的时间效率. 另外对 3 种冲突缓存更新的检出性能, 实验结果与分析表明本文方案 3 种冲突缓存更新检出性能上存在一定优势.

参考文献

1 Rekhter Y, Li T, Hares S, *et al*. A border gateway protocol 4 (BGP-4). RFC 4271, 2006. [doi: 10.17487/RFC4271.]
2 Lepinski M, Kent S. An infrastructure to support secure internet routing. RFC 6480, 2012. [doi: 10.17487/RFC6480.]
3 马迪. RPKI 概览. 电信网技术, 2012, (9): 30–33.
4 Durand A. Resource public key infrastructure (RPKI) technical analysis. OCTO-014. California: ICANN Office of the Chief Technology Officer, 2020. 15–24.

5 Kristoff J, Bush R, Kanich C, *et al*. On measuring RPKI relying parties. Proceedings of the ACM Internet Measurement Conference. New York: ACM, 2020. 484–491. [doi: 10.1145/3419394.3423622]

6 Cooper D, Heilman E, Brogle K, *et al*. On the risk of misbehaving RPKI authorities. Proceedings of the 12th ACM Workshop on Hot Topics in Networks. College Park: ACM, 2013. 16.

7 刘晓伟, 延志伟, 耿光刚, 等. RPKI 中 CA 资源分配风险及防护技术. 计算机系统应用, 2016, 25(8): 16–22. [doi: 10.15888/j.cnki.csa.005313]

8 Xing QQ, Wang BS, Wang XF. POSTER: BGPcoin: A trustworthy blockchain-based resource management solution for BGP security. Proceedings of 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: ACM, 2017: 2591–2593. [doi: 10.1145/3133956.3138828]

9 Xing QQ, Wang BS, Wang XF. BGPcoin: Blockchain-based Internet number resource authority and BGP security solution. Symmetry, 2018, 10(9): 408. [doi: 10.3390/sym10090408]

10 Shrishak K, Shulman H. Limiting the power of RPKI authorities. Proceedings of Applied Networking Research Workshop. Virtual Event: ACM, 2020. 12–18. [doi: 10.1145/3404868.3406674]

11 Kent S, Ma D. Adverse actions by a certification authority (CA) or repository manager in the resource public key infrastructure (RPKI). RFC 8211, 2015. [doi: 10.17487/RFC8211]

12 Heilman E, Cooper D, Reyzin L, *et al*. From the consent of the routed: Improving the transparency of the RPKI. ACM SIGCOMM Computer Communication Review, 2015, 44(4): 51–62. [doi: 10.1145/2740070.2626293]

13 Hlavacek T, Cunha I, Gilad Y, *et al*. DISCO: Sidestepping RPKI's deployment barriers. Proceedings of Network and Distributed Systems Security (NDSS) Symposium. San Diego, 2020. 1–17. [doi: 10.14722/ndss.2020.24355.]

14 耿新杰, 马迪, 毛伟, 等. 基于 HTTPS 的 RPKI 缓存更新机制. 计算机系统应用, 2019, 28(9): 72–80. [doi: 10.15888/j.cnki.csa.007050]

15 Bush R, Austein R. The resource public key infrastructure (RPKI) to router protocol. RFC 6810. 2013. [doi: 10.17487/RFC6810]

16 Asturiano V. The shape of a BGP update. https://labs.ripe.net/author/vastur/the-shape-of-a-bgp-update/. [2021-06-30].