高技术通讯 2020 年 第30卷 第11期:1104 ~1112

doi:10.3772/j.issn.1002-0470.2020.11.002

基于区块链的命名数据网络内容毒化攻击抵御机制①

郭 江②*** 王 森* 许志伟* 张瀚文* 张玉军③***

(*中国科学院计算技术研究所 北京 100190)

(**中国科学院大学 北京 100049)

摘 要 本文提出了一种基于区块链内容毒化攻击抵御机制(BlockIKB)。该机制在边缘路由器上引入区块链数据库,存储内容名字和发布者公钥摘要的绑定信息,在内容获取过程中,用户可以就近从边缘路由器获取发布者的公钥摘要,路由器根据公钥摘要验证内容,从而抵御毒化内容。与已有机制相比,本文机制能够抵御内容毒化攻击。构建了分布式数据库,避免了用户集中获取公钥摘要导致的拥塞问题;提供了就近公钥摘要获取服务,提升了内容获取效率。安全分析证明该机制能够抵御内容毒化攻击,实验结果表明,该机制能够减轻服务器负载,提升内容获取效率。

关键词 命名数据网络(NDN);内容毒化攻击(CPA);摘要验证;区块链;分布式数据库

0 引 言

为了确保用户接收完整、真实的内容,命名数据网络(named data networking,NDN)[1,2]规定内容发布者对每个数据包进行签名,用户对收到的数据包进行验证签名。考虑到计算开销巨大等因素,NDN没有要求沿途路由器对数据包进行验证签名(网内签名验证)。攻击者通过劫持路由器,向其缓存注册毒化内容。对于到来的请求兴趣包,恶意路由器返回对应名字的毒化内容数据包,导致反向路径的路由器转发并缓存毒化内容数据包,最终使得用户接收到毒化内容数据包,这种攻击称为内容毒化攻击(content poisoning attack,CPA)[3]。文献[4,5]指出,根据内容被篡改信息的不同,毒化内容攻击可以分为无效签名和冒名签名。无效签名是指数据包的内容被毒化(破坏或者篡改),而冒名签名是指数据包的签名被毒化,即签名密钥是他人冒名的。

现有解决CPA的方案根据验证实体的不同大致可以归结为2类:基于用户反馈的机制和基于网内验证的机制。基于用户反馈的机制提出让用户对数据包进行签名验证,利用用户反馈使路由器在不执行复杂签名验证的情况下识别被污染的数据包,减少内容污染攻击的影响。Gasti等人[3]提出利用兴趣包中的EXCLUDE域,排除再次接收相同无效的内容包。Ghali等人[6]提出一种统计内容排名算法,路由器根据用户反馈对其缓存副本进行排除次数统计排序,从而区分有效和污染的内容。文献[7]提出2种回避解决方法:即时故障转移法和探针优先法。在第1种方法中,简单地把返回污染数据的路由器作为后续请求包传输时优先级最低的下一跳节点;在第2种方法中,路由器需要验证用户的反馈,并把污染数据包对应的请求包作为探针来检测恶意攻击者。上述机制都需要修改用户端,使其主动发送反馈信息,导致通信开销大。

基于网内验证的机制提出让沿途路由器对数据

① 国家重点研发(2018YFB1800403,2016YFE0121500),国家自然科学基金(61902382,61972381,61672500,61572474)和中国科学院战略性先导科技专项(XDC02030500)资助项目。

② 男,1986年生,博士生;研究方向:未来互联网,网络安全,区块链;E-mail:guojiang@ict.ac.cn

③ 通信作者,E-mail:zhmj@ict.ac.cn

(收稿日期:2019-12-13)

包进行签名验证,一旦发现污染数据就将其剔除,这样整个网络都不会缓存被污染的内容。然而考虑到网络内验证带来的负荷,很多优化的签名验证方法被提出[3,8-11]。其中,Gasti 等人[3]提出选择性验证机制,即路由器随机选择一部分经过它的数据包进行验证,但这种机制无法保证未经验证的数据包的正确性,无法确定到底能够获得多大的安全保证。Bianchi 等人[8]提出通过降低缓存来减少内容验证的计算量,即在路由器设置校验缓冲区,当数据包到达路由器时,若缓冲区存在剩余空间,路由器将按某固定概率实施存入校验;若缓冲区已满,直接将数据交付到下游路由器。Kim 等人[9]提出只验证流行的内容,避免不必要验证。上述机制虽然能够在一定程度上缓解无效签名内容毒化攻击,但都无法抵御冒名签名的内容毒化攻击。Ghali 等人[10]提出兴趣包公钥绑定机制(interest key binding,IKB),通过绑定内容名字和内容发布者的公钥摘要,确保路由器转发的数据包来源于真实的发布者。该机制能够抵御冒名签名的内容毒化攻击,但依赖一个类似 PKI 的集中式数据库系统,容易导致公钥集中获取拥塞问题,甚至导致单点故障问题,而且用户需要从远端访问集中式的数据库获取相关信息,延时较大。

为了克服 IKB 机制集中获取公钥导致服务器拥塞问题,提升正确内容获取效率,本文提出了一种基于区块链的命名数据网络内容毒化攻击抵御机制(blockchain-based content poisoning attacks defense mechanism in NDN,BlockIKB)。区块链[12,13]作为一种去中心化、不可篡改、可追溯、多方共同维护的分布式多副本数据库,可在互不了解的多方之间建立可靠的信任,在没有第三方中介机构的协调下,实现可信的数据共享。鉴于区块链的上述特性,本文基于区块链建立一个分布式的数据库,存储发布者内容名字和公钥摘要绑定信息,使得用户就近访问对应内容名字的公钥摘要,路由器根据公钥摘要对到来的内容进行 Hash 验证,以抵御内容毒化攻击。

1　NDN 背景

NDN 网络包括用户(Consumer)、发布者(Publisher or Producer)和路由器(Router)3 类实体,其数据传输主要包括 2 种类型:兴趣包(Interest)和数据包(Data)。兴趣包是由用户发送的数据请求包,其中包括请求的内容名称(Name)、发布者公钥摘要(publisher public key digest,PPKD)等;数据包是由发布者或路由器根据用户的请求返回的内容,其中包括内容名称(Name)、内容本身(Content)、发布者的签名(Signature)、发布者签名公钥定位器(Key Locator)等。NDN 网络中单个节点通信流程如图 1 所示。每个实体包含 3 种数据结构,分别是内容缓存(content store,CS)、待定兴趣表(pending interest table,PIT)和转发信息表(forwarding information base,FIB)。CS 用于存储接收到的数据包,对于后续相同的内容请求从本地响应数据包,有利于减少对于发布者的访问次数,提升内容分发的传输效率;PIT 记录待转发兴趣包的内容名称以及接入接口,并且汇聚相同的兴趣包在一个表项中;FIB 依靠路由协议生成,记录兴趣包转发下一跳接口。

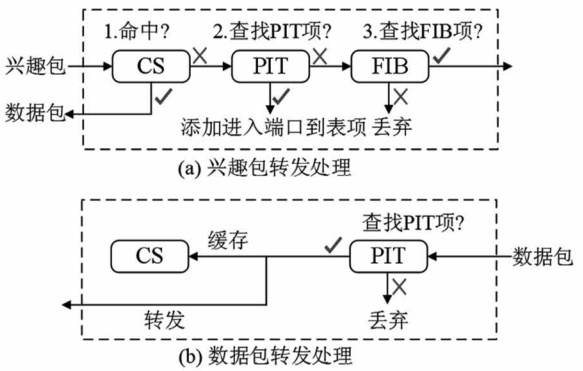


图 1　命名数据网络单个节点通信流程

NDN 网络用户获取内容的过程分以下 3 个步骤:(1)当需要内容时,用户发送一个兴趣包。路由器收到兴趣包后,首先查找 CS 是否有请求的数据,如果有,则从兴趣包接入接口返回数据包并丢弃兴趣包;否则,继续查找 PIT,查找之前是否转发来自其他节点的、并且与该条目的请求内容相同的兴趣包。如果找到,则将本次兴趣包的接入接口添加到对应的 PIT 信息条目中;否则,在 PIT 中创建兴趣包接入接口的信息条目,继续查找 FIB,进行路由寻址。(2)兴趣包到达发布者并找到内容对象时,兴趣

包被丢弃，响应的信息以数据包的形式原路返回。当数据包到达路由器时，首先查找 CS，如果有相同的缓存数据，则丢弃数据包；若没有，则与 PIT 中条目匹配。如果 PIT 中有匹配条目，则向相应的接口转发数据包，缓存数据包在 CS 中，并删除 PIT 中的匹配条目；否则丢弃数据包。(3)用户在接收到数据包后进行签名验证，确保内容完整性和真实性。

2　BlockIKB 机制

2.1　设计思想

本文实现了一种新的抵御内容毒化攻击方法，解决用户集中获取公钥导致服务器拥塞问题以及正确内容获取效率低问题。BlockIKB 机制整体协议框架如图 2 所示，其设计思想是网络边缘各自治域中边界路由器共同维护分布式数据库，存储发布者注册绑定信息，即内容名字和发布者公钥摘要；内容获取过程，用户先就近访问边缘路由器，获取内容名字对应的发布者公钥摘要，并构造请求包携带源公钥摘要；途径的路由器提取请求包中源公钥摘要并存储，依据此凭证，对到来的数据包进行 Hash 验证，若验证成功，则对数据包进行转发并存储；否则，剔除该数据包。下面，首先介绍 BlockIKB 相关存储结构，然后具体说明发布者注册公钥摘要过程和用户获取内容过程。

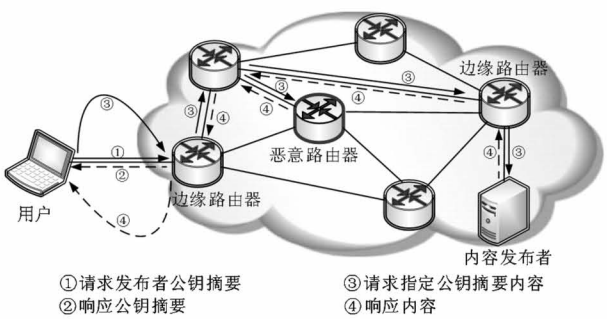


图 2　整体协议框架

2.2　注册信息存储结构

BlockIKB 定义了如图 3 所示的数据结构存储内容名字和发布者公钥摘要。注册信息存储结构是由多个数据块组成的链式结构，每个数据块分为块头部(Header)和块体(Body)两部分。块头部包括前置数据块散列值(Pre Hash)、时间戳(Timestamp)、随机数(Nonce)、难度值(Difficulty)、本数据块散列值(Local Hash)。前置数据块散列值是本块前置数据块的散列值，指向上个数据块，正是这种逐级包含形成链式结构[14]；时间戳表示数据块生成时间；随机数是工作量证明(proof of work，PoW)[15]的解，采用工作量证明机制提高数据块产生的代价，防止分布式系统恶意节点随意生成数据块；难度值表示目标散列值前导零的位数；本数据块散列值表示当前块数据的目标散列值。块体包括内容名称(Name)、内容发布者公钥摘要(PPKD)。

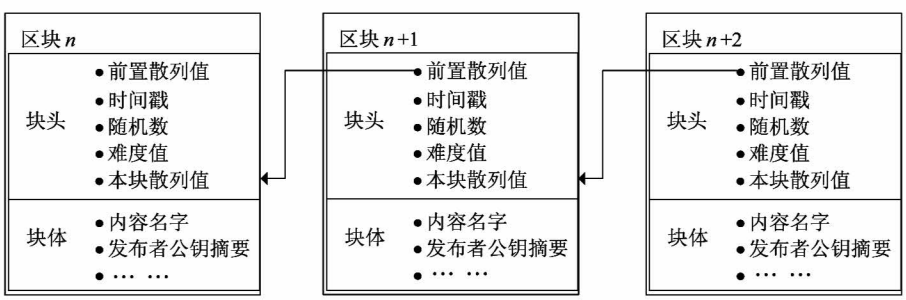


图 3　BlockIKB 数据结构

2.3　发布者注册公钥摘要过程

发布者注册公钥摘要是指发布者将待发布内容的名字和其公钥摘要绑定信息注册到边缘分布式数据库。设计的基本思路为发布者将 < Name，PPKD > 注册到各自治域中边界路由器，边缘路由器通过分布式数据库同步保障数据的一致性。发布者注册公钥摘要过程包括 2 个子过程，即边缘路由器获取发布者公钥摘要过程和边缘路由器同步过程，如图 4和图 5 所示。

边缘路由器获取发布者公钥摘要子过程，如

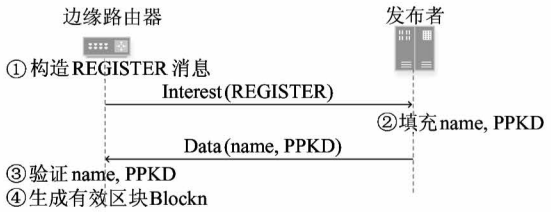


图4　发布者注册信息获取

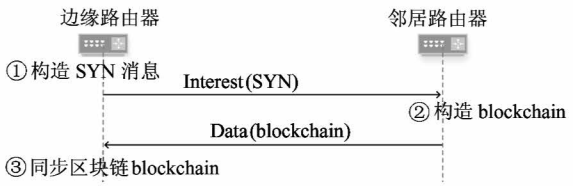


图5　同步交互

图4所示,具体过程如下。

步骤1　边缘路由器周期性向发布者发送REGISTER消息兴趣包。

步骤2　发布者收到REGISTER消息,如果有注册需求,则将<Name,PPKD>作为内容字段并发送,否则不操作。

步骤3　边缘路由器收到对应数据包,提取信息<Name,PPKD>并查询本地数据库是否存在相同的Name,若查询成功,则不操作,否则执行步骤4。

步骤4　为了防止路由器随意产生数据块,采用PoW算法(见表1)。具体地,边缘路由器利用SHA256散列函数,通过改变Nonce值,重复计算数据块散列值,SHA256(Pre Hash, Timestamp, Nonce, Difficulty, Content name, PPKD),直到满足难度值;然后,将其封装成数据块链接到本地数据库。

边缘路由器同步过程是考虑到每个发布者向其所在自治域的边缘路由器进行注册,所以各边缘路由器维护的数据库可能不一致,为了维护这些路由器的分布式数据库的一致性而设计,如图5所示,具体过程如下:

步骤1　边缘路由器周期性给邻居路由器发送SYN消息兴趣包。

步骤2　邻居路由器收到SYN消息,将本地链数据封装成数据包并发送。

步骤3　边缘路由器收到对应数据包,提取链

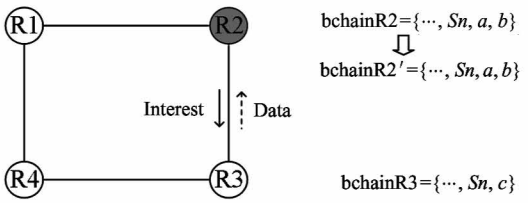
表1　BlockIKB的基本算法

	算法实现流程
1	收集注册消息;
2	Nonce = 0;
3	While (Hash不满足difficult条件)
4	{
5	Nonce + +;
6	Hash = SHA256(Pre Hash,…,Nonce,…,PPKD);
7	}
8	封装成区块并广播;
9	If (length(localbc) > length(synbc))
10	{
11	不执行操作;
12	}
13	Else
14	验证区块并更新数据库;

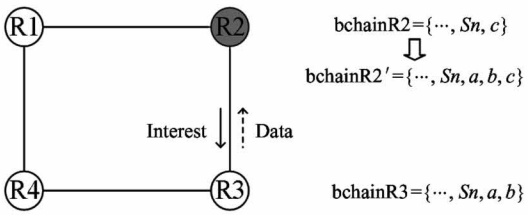
将其存储为synbc,并与本地数据库链localbc比较长度以解决同步冲突。由于数据库链是单链结构,随着新的数据块的产生,数据库链不断更新延长。因此链上的数据块数量越多,说明数据库链累计的难度越大,故按照数据库链最长原则处理冲突,即当数据块链不一致时,选择链长较大的链作为主链。具体地分2种情况处理:(1)若本地数据库长度大于接收数据库长度,即length(localbc)>length(synbc),则不操作。例如图6(a)中,路由器R2请求邻居路由器R3同步,R3返回链为bcchainR3={…, Sn, c},路由器R2本地链为bcchainR2={…, Sn, a, b},更新之后的链为bcchainR2′={…, Sn, a, b}。(2)若本地链长度小于等于接收链长度,即length(localbc)<=length(synbc),则截取2个链不同的部分,将localbc截取的部分,以synbc最后一个数据块为基准,依次重新封装数据块,链接到synbc最后,并以该synbc为本地链。例如图6(b)中,路由器R2请求邻居路由器R3同步,R3返回链bcchainR3={…, Sn, a, b},路由器R2本地链为bcchainR2={…, Sn, c},更新之后的链为bcchainR2′={…, Sn, a, b, c}。

2.4　用户获取内容过程

用户获取内容过程是用户先就近从临近节点获



（a）冲突处理情形 1



（b）冲突处理情形 2

图 6　数据同步冲突解决示意图

取对应内容名字的 PPKD，再请求指定 < Name，PPKD > 的对应内容。具体来说，用户首先从边缘路由器 < Name，PPKD > 数据库获取 PPKD 值，然后将其存储在兴趣包的 PPKD 字段中。当收到用户兴趣包时，路由器把 PPKD 字段中的值记录到相应的待定请求表 PIT 中，并转发兴趣包。在返回请求的内容时，内容发布者将其公钥（public key，PK）存储在数据包的 Key Locator 字段中。这样路由器就可以对存储在 Key Locator 字段中的公钥进行散列运算，检查是否与相应 PIT 条目中的 PPKD 值匹配。如果匹配，则缓存并转发这个数据包；否则，丢弃该数据包。具体过程如图 7 所示。

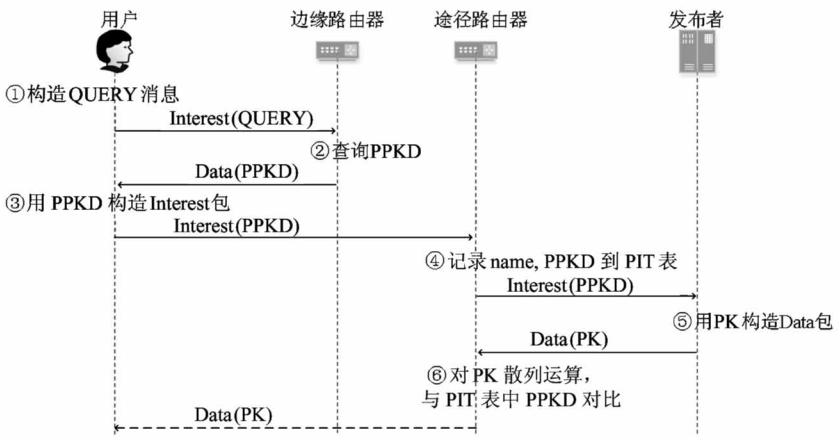


图 7　内容获取过程

步骤 1　用户向其边缘路由器发送 QUERY 消息兴趣包。

步骤 2　边缘路由器收到 QUERY 消息，根据内容名字，在本地数据库查询对应的 PPKD 值，若查找成功，返回携带 PPKD 值数据包；否则返回查询失败数据包。

步骤 3　用户获得查询 PPKD 值，构造并发送兴趣包请求获取内容。

步骤 4　途径路由器收到兴趣包，先查询 CS，若缓存命中，则返回数据包；否则查询 PIT 表，若有相同项，追加接口项；否则添加新表项，记录内容名、PPKD 值、传入的接口，并根据 FIB 表转发。

步骤 5　发布者收到兴趣包，构造内容，并将自身的公钥 PK 存储在数据包 Key Locator 字段，最后返回数据包。

步骤 6　途径路由器收到数据包，提取 Key Locator 字段中的发布者公钥 PK，计算 PK 的散列值 PKD，即 PKD = Hash（PK），并根据 Name 和 PKD 查找 PIT 表，检查是否与相应 PIT 条目中的 PPKD 值匹配。如果匹配，则缓存该数据包并转发到对应接口；否则，丢弃该数据包。

3　安全分析

本节分析 BlockIKB 机制的安全性。首先给出关于散列函数和签名策略的 2 个定义。

定义 1　**抗第二原像性**。散列函数 H 具有抗第二原像性。对于任意给定数值 x，不存在概率多项

式时间内,找到一个值 $x_0(x_0 \neq x)$,使得 $H(x_0) = H(x)$,散列碰撞发生。即 $\Pr[H(x_0) = H(x)] \leqslant \theta(n)$,其中 $\theta(n)$ 可忽略不计[16]。

定义 2　不可篡改性。签名策略具有不可篡改性。对于任意给定消息 m,不存在概率多项式时间内,恶意节点 A 获知公钥却不知私钥的情况下,篡改签名并使签名有效。令 A 对 m 篡改签名并使签名有效的事件,$A^{\text{forge}}(m) = 1$。$\Pr[A^{\text{forge}}(m) = 1] \leqslant \theta(n)$,其中 $\theta(n)$ 可忽略不计。

下面对 BlockIKB 机制进行形式化描述,并给出定理及证明过程。

给定 PPKD 字段值为 $H(PK)$ 的兴趣包 Int,作为恶意节点 A 的输入,它输出一个数据包 C_0,其中包含 Key Locator 字段中的公钥 PK_0,PPKD 字段公钥摘要 $H(PK_0)$,Signature 字段的签名信息 σ_0。如果输出是以下情况之一,那么 A 成功注入毒化内容到网络,记为 $A^{\text{pois}}(Int) = 1$:

(1) $PK \neq PK_0$ 且 $H(PK) = H(PK_0)$。A 违反 H 抗第二原像性,即发生散列碰撞,其散列碰撞成功的概率由碰撞概率 $\Pr^{\text{collision}}$ 与 $\Pr[H(PK) = H(PK_0)]$ 决定;

(2) $PK = PK_0$ 和 σ_0 签名有效。A 违反签名策略的不可篡改性,其发生篡改签名成功的概率由篡改签名发生的概率 $\Pr^{\text{forge}}$ 与 $\Pr[A^{\text{forge}}(m) = 1]$ 决定。

定理　如果散列函数 H 具有抗第二原像性,签名策略具有不可篡改性,那么恶意节点 A 以可忽略的概率成功注入毒化内容 C_0 到网络,即 $\Pr[A^{\text{pois}}(Int) = 1] \leqslant \theta(n)$,$\theta(n)$ 可忽略不计。

证明(反证法)　假设 A 以一定的概率成功注入毒化内容 C_0,即 $\Pr[A^{\text{pois}}(Int) = 1] > \theta(n)$。

构造另外一个恶意节点 A′,利用 A 破坏 H 抗第二原像性或者签名策略的不可篡改性。即满足给定 x,创建兴趣包 Int,设置 $H(x)$ 作为 PPKD 字段值,运行 $A(Int)$ 以获取 C_0。从 C_0 中可以得到结果,使得 $x \neq PK_0$,$H(x) = H(PK_0)$;或者使得 $x = PK_0$,σ_0 是 C_0 篡改签名且有效。可以确定 A′成功注入毒化内容概率,即散列碰撞或者篡改签名事件发生的概率:

$$\begin{aligned}
&\Pr[A'\text{成功注入}] \\
&= \Pr[\text{散列碰撞} \cup \text{篡改签名}] \\
&= \Pr^{\text{collision}} \times \Pr[H(x) \\
&= H(PK_0)] + \Pr^{\text{forge}} \times \Pr[A'^{\text{forge}}(C_0) = 1] \\
&> \theta(n)
\end{aligned}$$

上式成立是因为 A′与 A 有相同概率成功注入毒化内容,A 以一定的概率成功注入,所以 A′同样以一定的概率成功注入。如果 2 个函数的概率之和是不可忽略的,那么它们中至少一个是不可忽略的。

由于 $\Pr^{\text{collision}}$ 和 $\Pr^{\text{forge}}$ 不是指数增长的函数,可以得出:

$\Pr[H(x) = H(PK_0)] > \theta(n)$ 或 $\Pr[A'^{\text{forge}}(C_0) = 1] > \theta(n)$,这与定义 1 和定义 2 产生矛盾。

故 $\Pr[A^{\text{pois}}(Int) = 1] \leqslant \theta(n)$。

证毕。

以上定理说明 BlockIKB 机制是安全的,能够抵御内容毒化攻击。

4　实验评估

4.1　方案部署

本文在 NDNSim[17] 网络仿真平台上,实现了基于边缘分布式数据库的内容毒化攻击抵御机制。仿真平台在本地计算机上部署,其配置如下:CPU Intel Core i7 3.4 GHz,内存 16 GB,硬盘 1 TB,操作系统内核 Ubuntu 14.04,NDNSim 版本 1.0。模拟参数配置如表 2 所示。

表 2　模拟参数及其取值

参数类型	参数描述	值
L	链路带宽	10 Mbps
D	链路延时	1 ms
S	缓存策略	LRU
C	缓存大小	1 000 content items
M	待定兴趣表大小	15 000 entries
I	数据包大小	1 024 bytes
R	请求速率	50 Interests/s
F	持续时间	0 ~ 600 s

仿真实验采用的网状拓扑包括 50 个路由器、1 个 IKB 机制服务器、3 个用户和 2 个内容发布者。

设置内容发布者为 A 和 B,分别以组件名“/google.com”和“/youtube.com”为前缀,后缀为随机数。内容发布者以 3.6 packet/min 进行注册,用户以50 pakcet/s 获取前面的组件名内容。设置 PIT 大小为15 000,缓存大小 1 000,采用 RSA 算法对内容进行签名,使用 SHA1 散列算法计算内容发布者的公钥摘要。

4.2 测量指标

为了评估 BlockIKB 机制的安全性、服务器负载、内容获取效率和通信开销,使用了 4 项指标:检出率、服务器负载、延迟和通信开销。检出率是网内路由器能够检测出的毒化内容比例,即检测出的毒化内容数据包数量与毒化内容数据包总数的比值。服务器负载是指服务器节点单位时间处理查询兴趣包的数量。延迟是指内容获取过程中,从用户发送兴趣包到接收到响应内容的时间,单位为 ms。通信开销主要集中在注册过程的开销,故本文以发布者注册信息过程的通信开销为主,即发布者注册信息时平均转发兴趣包和数据包的数量。

4.3 抵御毒化内容安全性分析

通过毒化内容的检出率来评估 BlockIKB 抵御毒化内容的安全性。每个用户以 50 packet/s 发送合法内容请求,请求兴趣包的名字由名字前缀和随机数组成,这里使用 5 项名字前缀:“/sina.com”,“/readfar.com”,“/cs-bu.edu”,“/google.com”,“/youtube.com”。1 个内容生产者提供毒化内容到网络。图 8 显示了 BlockIKB 和 IKB 机制毒化内容检出率,可以观察到它们都能检测出所有的毒化内容,检出率为 100%。这是因为路由器在转发和缓存数据包之前,先对到来的数据包,根据从兴趣包提取的内容

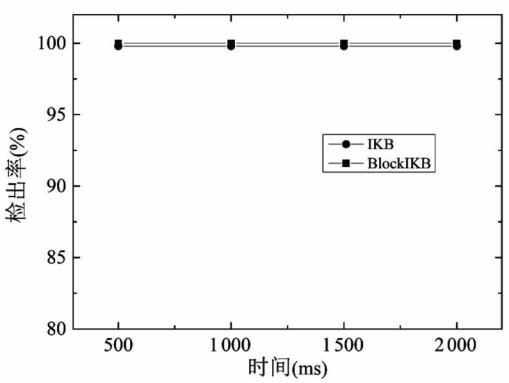


图 8 毒化内容检出率

发布者公钥摘要记录验证数据包所携带的公钥,从而可以识别出毒化内容包。

4.4 服务器负载分析

为了评估 BlockIKB 提供的分布式数据库均摊用户请求的流量,统计服务器负载。3 个用户以 50 packet/s发送公钥或者公钥摘要请求,分别统计 BlockIKB 机制 3 个边缘路由器平均处理请求的负载情况以及 IKB 机制服务器处理请求的负载情况,单位时间 s。图 9 显示了 BlockIKB 和 IKB 机制的服务器负载累积分布函数。如图所示,BlockIKB 在多处情况下负载低于 50,而 IKB 机制几乎所有情况均大于 60。而且 BlockIKB 边缘路由器 85% 的概率单位时间处理请求 46 packet,而 IKB 机制服务器 85% 的概率单位时间处理请求 63 packet。此数据表明针对相同速率的请求,BlockIKB 的负载情况要减轻 37%。究其原因在于分布式服务器均摊了用户请求流量,避免单个服务器负载过大。

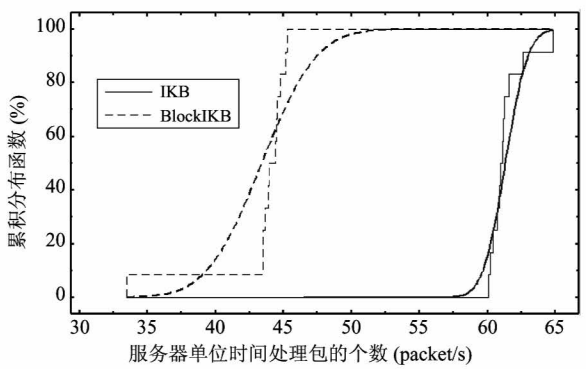


图 9 服务器负载累积分布函数

4.5 传输效率分析

通过用户内容获取的延迟来分析 BlockIKB 机制的传输效率。图 10 显示了 BlockIKB 和 IKB 机制的内容获取延迟,从图中可以看出,IKB 用户获取内容延时约 22 ms,而 BlockIKB 延时约 11 ms。实验表明 BlockIKB 机制比 IKB 机制延时节省了 50%。这是因为 BlockIKB 就近获取公钥摘要数据,从而缩短了获取内容的时间,大幅提升了获取内容的效率。

4.6 通信开销分析

BlockIKB 机制引入的通信开销主要存在于发布者注册公钥摘要过程,而对比方案IKB机制引入

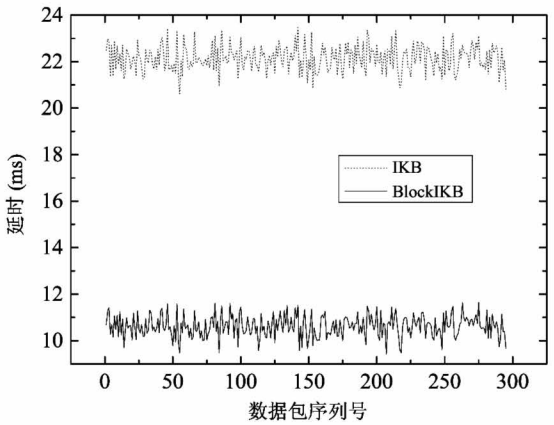


图 10　内容获取延迟

的通信开销主要存在于注册公钥过程。通过比较上述 2 个过程所需转发的同步兴趣包和数据包数量,对 2 种机制的通信开销进行分析。设置难度值为 5,使得边缘节点平均大约每 50 s 生成 1 个数据块,以每 25 s 进行发送同步兴趣包。图 11 显示了以每注册 5 条信息为单位进行统计的 BlockIKB 和 IKB 的通信开销。从图中可以看出,BlockIKB 注册信息所需开销至少为 120 packet,而 IKB 机制所需开销大约维持在稳定水平 40 packet。结果表明 BlockIKB 比 IKB 维护注册信息开销平均约高 3 倍,主要原因是 BlockIKB 要通过周期性发送同步兴趣包维护分布式数据库。

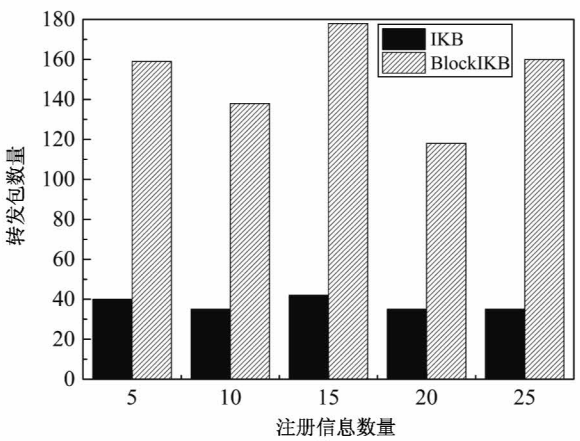


图 11　通信开销

5　结　论

针对内容污染攻击,提出了一种基于区块链内容毒化攻击抵御机制——BlockIKB 机制。构建了区块链数据库来存储内容名字和发布者公钥摘要的绑定信息;改进了内容获取过程,使得用户就近获取发布者的公钥摘要;路由器根据公钥摘要验证内容,实现了污染内容抵御功能。安全分析证明了 BlockIKB 机制能够抵御内容毒害攻击;评估结果表明,与已有机制相比,BlockIKB 机制减轻了服务器负载,提升了内容获取效率。

本文在部署区块链节点时,采用的是静态添加方式,主要考虑在 NDN 网络架构中小范围内使用区块链,今后在大范围内扩展,需要考虑动态接入。另外,在维护区块链时采用 PoW 机制,今后尝试其他高效的共识机制。下一步将从以上 2 方面进行改进,为 NDN 网络架构的应用提供更好的服务。

参考文献

[1] Zhang L, Estrin D, Burke J, et al. Named data networking (NDN) project, Technical Report NDN-0001[R]. Los Angeles: Xerox Palo Alto Research Center-PARC, 2010

[2] Saxena D, Raychoudhury V, Suri N, et al. Named data networking: a survey[J]. *Computer Science Review*, 2016, 19: 15-55

[3] Gasti P, Tsudik G, Uzun E, et al. DoS and DDoS in named data networking[C]// IEEE International Conference on Computer Communication and Networks, Nassau, Bahamas, 2013: 1-7

[4] Gasti P, Tsudik G, Uzun E, et al. DoS DDoS in named-data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2013, 44(3): 66-73

[5] Tan N, Marchal X, Doyen G, et al. Content poisoning in named data networking: comprehensive characterization of real deployment[C] // Integrated Network and Service Management, Lisbon, Portugal, 2017: 72-80

[6] Ghali C, Tsudik G, Uzun E. Needle in a haystack: mitigating content poisoning in named-data networking[C]// Proceedings of NDSS Workshop on Security of Emerging Networking Technologies, San Diego, USA, 2014: 1-10

[7] Dibenedetto S, Papadopoulos C. Mitigating poisoned content with forwarding strategy[C]// Computer Communications Workshops, San Francisco, USA, 2016: 164-169

[8] Bianchi G, Detti A, Caponi A, et al. Check before storing: what is the performance price of content integrity

verification in LRU caching? [J]. *ACM SIGCOMM Computer Communication Review*, 2013, 43(3): 59-67

[9] Kim D, Nam S, Bi J, et al. Efficient content verification in named data networking[C]//ACM Conference on Information-Centric Networking, San Francisco, USA, 2015: 109-116

[10] Ghali C, Tsudik G, Uzun E. Network-layer trust in named-data networking[J]. *ACM SIGCOMM Computer Communication Review*, 2014, 44(5): 12-19

[11] Ghali C, Tsudik G, Wood C A. Mitigating on-path adversaries in content-centric networks[C]//IEEE Local Computer Networks, Singapore, 2017: 27-34

[12] Yaga D, Mell P, Roby N, et al. Blockchain technology overview[J]. *arXiv*:1906.11078, 2018

[13] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system[EB/OL]. http://www.btcpapers.com/bitcoin.pdf: Bitcoin, 2008

[14] Jin T, Zhang X, Liu Y, et al. Blockndn: a bitcoin blockchain decentralized system over named data networking[C]//Ubiquitous and Future Networks, Milan, Italy, 2017: 75-80

[15] Liu D, Camp L J. Proof of work can work[C]//Workshop on the Economics of Information Security, England, UK, 2006: 1-16

[16] Katz J, Lindell Y. Introduction to Modern Cryptography[M]. New York: CRC Press, 2008

[17] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3, Technical Report NDN-0005[R]. Los Angeles: University of California, 2012

Blockchain-based content poisoning attacks defense mechanism in named data networking

Guo Jiang[*,**], Wang Miao[*], Xu Zhiwei[*], Zhang Hanwen[*], Zhang Yujun[*,**]

([*] Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**] University of Chinese Academy of Sciences, Beijing 100049)

Abstract

This paper proposes a blockchain-based content poisoning attacks defense mechanism (BlockIKB). Blockchain database is introduced into the edge routers which collect binding information including the content name and producer's public-key digest. In content retrieval process, consumers pre-acquire producer's public-key digest from their nearby edge routers. Then, according to this public-key digest, in-network routers validate incoming content to defense fake content. Compared with the existing solutions, the proposed solution can defense content poisoning attacks. Constructing a distributed database, it avoids congestion problem caused by consumers centralized acquisition. In addition, it improves the consumer's efficiency in retrieving content by providing a nearby acquisition service of public-key digests. Security analysis shows that the proposed solution can defense content poisoning attacks. The experimental results confirm that this solution can mitigate server load and improve the consumer's efficiency in retrieving content.

Key words: named data networking (NDN), content poisoning attack (CPA), public-key digest, blockchain, distributed database