文章编号：2095-2163(2020)02-0365-05 中图分类号：TP393.08 文献标志码：A

基于 IPFS 的域名解析技术研究

刘姝言，翟健宏
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：由于 DNS 设计之初没有考虑其安全性因素，使其脆弱性日益暴露出来。本文利用 IPFS 技术构建新型的域名解析服务。域名解析服务简化为根和顶级域两层结构，减少查询次数。将资源记录文件上传到 IPFS 网络，IPFS 是一种基于内容寻址的分布式网络，该网络可永久保存上传的文件，并且使用 P2P 传输协议从邻居节点传输资源记录文件缩短时延。针对域名解析的安全问题，本文还增加了签名验证机制，使用非对称密码从文件父域到子域形成信任链，能够有效地解决域名解析的安全问题。

关键词：域名系统；星系文件系统；信任链

Research on domain name resolution technology based on IPFS

LIU Shuyan, ZHAI Jianhong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 Since the DNS design did not consider its security factors at the beginning, its vulnerability is increasingly exposed. This paper uses IPFS technology to build a new domain name resolution service. The domain name resolution service is simplified into a two-tier structure of root and top-level domains, reducing the number of queries. The resource record file is uploaded to the IPFS network, which is a distributed network based on content addressing, and can permanently save the uploaded file and reduce the delay by transmitting the resource record file from the neighbor node using the P2P transmission protocol. Aiming at the security issue for domain name resolution, a signature verification mechanism has been added to form a trust chain from a parent domain to a child domain using an asymmetric password, and the security problem of domain name resolution could be effectively solved.

【Key words】 DNS; IPFS; chain of trust

0 引 言

作为现代互联网的基础服务，DNS 域名系统已经发展成为当前最成功、最庞大的分布式数据系统[1]。但是传统的 DNS 服务器存在数据中心化，安全性差等特点。为防止 DNS 的单点失效等各类问题，需要对其进行大量的镜像工作。IPFS 的技术原理是将文件内容用哈希算法生成一个哈希指纹[2]，并且同一个哈希算法对相同内容只能产生唯一的指纹，这样 IPFS 系统中的文件就具有去重性，降低了系统的冗余度。同时，IPFS 网络可以永久存储文件，某一节点中文件被删除也会在其它节点中查询到。基于此，本文将 IPFS 技术与 DNS 服务相结合，研究出一种新型、安全和高效的 DNS 解析服务。

1 基于 IPFS 设计域名解析器

1.1 IPFS 概述

区别于传统 HTTP 协议，IPFS 是按内容查找文件。研究将一个文件放在 IPFS 的节点，会得到一个新名字 QmYVCByHMDUSqedGkXqCC3V5Vn2nmoPoSpFQtnBuUAGc8V，这是一个由文件内容计算出的加密哈希值[3-6]。使用同一算法加密相同数据时，产生的哈希值是唯一的，并且散列的长度很短又固定，对于 IPFS 这种分布式网络来说，方便文件在不同节点之间存储和查询，Hash 函数示意图如图 1 所示。当要查询一个文件时，运行着 IPFS 节点的计算机会询问其所有对等点是否存在这个特定哈希值的文件，拥有该文件的节点将会返回整个文件。使用哈希值在网络中传输会减少非常多的网络带宽和时延[7]。每个 IPFS 节点中都记录的公网 ipv4 地址<addrs>，通过 id 获取操作可得到全部 IPFS 节点公网 ipv4 地址的列表 Address List，见表 1，将节点 IP 互连便可形成对等节点。

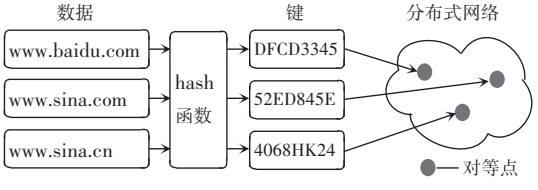

图 1 Hash 函数示意图

Fig. 1 Hash function diagram

作者简介：刘姝言（1996-），女，硕士研究生，主要研究方向：DNS、区块链；翟健宏（1968-），男，副教授，硕士生导师，主要研究方向：网络内容安全、网络安全、云安全等。

收稿日期：2019-06-10

表 1 IPFS 节点属性

Tab. 1 IPFS node properties

IPFS 节点属性	内容
<ip>	172.104.139.115
<id>	QmZeEq2QTWU9m4t9eiz2zbSPdYrSE3v7qX1ZmCHXqBtFkh
<addrs>	/ip4/172.104.139.115/tcp/4001/ipfs/QmZeEq2QTWU9m4t9eiz2zbSPdYrSE3v7qX1ZmCHXqBtFkh
<protocolversion>	ipfs/0.1.0

在 IPFS 网络中上传的每个文件及其所有数据块,都会根据内容返回唯一一个固定的散列字符串哈希指纹,也称为 CID。当文件大小超过数据块容量大小时,会被网络分块上传[8]。上传文件的节点会维护一个该文件的块哈希列表,方便其它节点检索时通过多节点并行下载数据块,如图 2 所示。

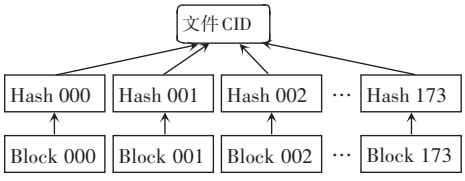

图 2 文件分块 Hash 示意图

Fig. 2 File block Hash schematic

同时每个对等节点都维护一张 DHT,DHT 中描述了每个对等节点拥有的数据块。DHT 表数据结构为二叉树,在下载文件时只需在 IPFS 公网中搜索文件哈希值,便会通过 DHT 找到包含该文件的节点,获得完整文件。多节点都拥有文件时根据节点的信用分选择节点,从高信用分节点优先下载数据。

IPFS 网络节点如图 3 所示,在云服务器上搭建了若干 IPFS 节点并且互连成对等点,对等点之间可以相互将节点信息和本节点存储的固定文件记录在自己的 DHT 表中,便于文件的检索和分发。整个网络都加入在 IPFS 公网中, IPFS 网络中的节点均能互相检索到文件信息。在其中若干节点上传域名文件,在上传节点成为永久文件并返回唯一的哈希指纹,其它节点想访问该文件时只需使用此文件的哈希指纹即可查看完整文件,并将文件缓存在本地中。

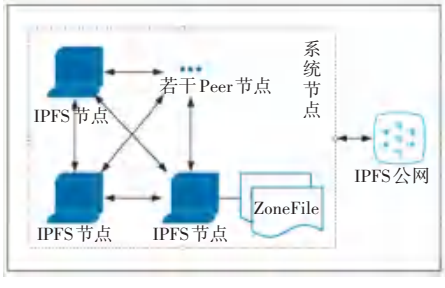

图 3 IPFS 网络结构图

Fig. 3 IPFS network structure diagram

1.2 资源记录数据源

鉴于 IPFS 网络节点之间传输数据低时延、多备份和防篡改的特点,研究将传统 DNS 的根、顶级域和权威的三层树状查询结构改为根和顶级域两层结构。将资源记录直接由顶级域分类存储,简化查询次数,缩短查询所用的时间。并且将资源记录文件存在 IPFS 网络中可实现文件的永久保存。

在配置解析服务之前要先在 IPFS 网络中上传域名解析文件,在搭建的不同节点并行上传顶级域名的解析数据。DNS Server 第一层文件为 root_file,内容见表 2,其上存储着所有顶级域名上传到 IPFS 网络的文件哈希值。第二层文件为 top_level_file,内容见表 3,其中存储着该顶级域名的所有资源记录。系统中所存域名是在 Alexa 网站上爬取的全球访问量前 100 万的域名。

表 2 root_file 格式

Tab. 2 root_file format

内容名	含义	示例值
DomainName	域名	com
DomainHash	域名哈希指纹	QmUpMrXsDCKFv4GsUAJFrjqZjEpTr3wTrxkCxwMyHT45kr

表 3 DomainFile 的内容格式

Tab. 3 DomainFile content format

内容名	含义	示例值
NAME	域名	google.com.
TTL	缓存有效时间	172 800
CLASS	固定为 IN,Internet	IN
TYPE	类型	A
IP ADDRESS	IP 地址	216.58.200.238

root_file 文件需要上传全部的 top_level_file 文件才能生成。在节点中分别上传 top_level_file,并记录 IPFS 网络返回的 CID 值。将 CID 与顶级域名相对应,写入另一文件中,此文件即为 root_file 文件。最后将 root_file 文件上传到 IPFS 网络中,返回根文件的 CID。根文件 CID 作为查询入口,开始逐级解析。

1.3　域名解析过程

用户端在查询域名 www.cnnic.cn 时，先将顶级域名.cn 剥离出来，在根文件中查找.cn 的顶级域名文件的 CID。再根据顶级域名文件的 CID 找到完整的顶级域名文件，从中解析出 www.cnnic.cn 的完整资源记录，并返回用户端 A 记录。

结合 IPFS 的设计理念，文件解析过程为在连接 IPFS 网络的节点客户端 CLI 上查询域名 www.cnnic.cn，首先会从本地的 blockstore 中查找是否存有 root_file 文件，若没有则通过 IPFS- API 向 swarm 网络发送查询请求。swarm 网络是由与自己节点直接相连的对等节点组成。通过 DHT 表查找拥有 root_file 的节点，根据 root_file 的 CID 找到完整 root_file 并返回到本地节点，本地节点先将文件缓存，系统再根据用户要查询的 DNS 报文 www. cnnic.cn 的顶级域名，在 root_file 中查找.cn 对应的 top_level_file 文件的 hash 值。CLI 重新在对等点间检索拥有 top_level_file 的节点，并获取完整文件，之后进行 DNS 解析并返回用户需要的解析信息。

用户发出一次请求的查询过程如图 4 所示。由图 4 可知，对此查询过程可做阐释分述如下。

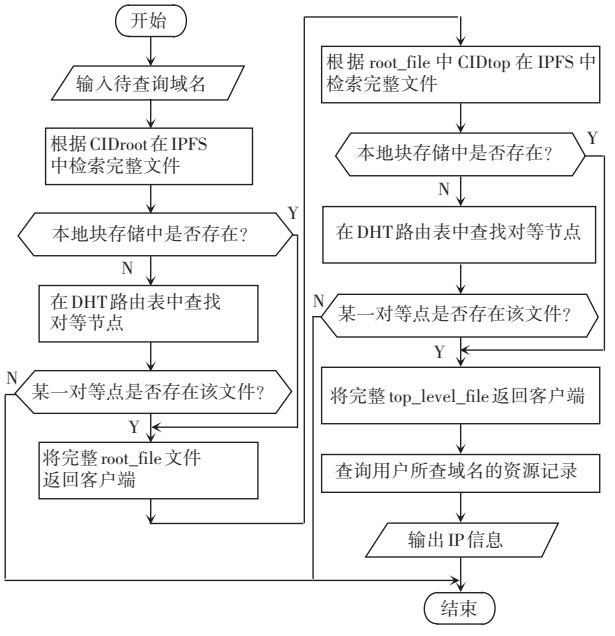

图 4　域名查询流程图

Fig. 4　Domain name query flow chart

(1)用户在客户端输入待查询的域名为 domain_name。

(2)客户端在其 Peer 节点的 DHT 表中查询 root_file 的 CID_{root}。

(3)从含有 root_file 的 Peer 节点下载完整的 root_file 文件。

(4) ipfs cat /ipfs/<CID_{root}>查看文件并获得顶级域的 CID_{top}。

(5)客户端在其 Peer 节点的 DHT 表中查询 top_level_file 的 CID_{top}。

(6)从含有 top_level_file 的 Peer 节点下载完整的 top_level_file 文件。

(7) ipfs cat /ipfs/<CID_{top}>的文件中获得用户查询域名的资源记录。

(8)查询到的 IP 信息返回给客户端。

(9)客户端收到 DNS 响应报文。

root_file 的 CID 作为域名解析的入口，但是需要用户从其它途径获取哈希指纹，并且要确保该哈希指纹的真实性和合法性才能进行正确的解析，但现实条件下哈希指纹的真实性难以保障。因此后续研究中增加了签名验证功能。

2　签名验证

2.1　签名机制

为保证文件的真实性和可用性，研究引入一个可信第三方的概念对在 IPFS 网络中上传的域名解析文件进行非对称密钥签名认证。由可信的第三方为每一个顶级域名和根文件分发一对公私钥，选定秘钥算法为 RSASHA256，秘钥长度 *keylen* 为2 048。

上述的两层域名解析结构是通过查询 CID 进行解析，尽管 CID 是每个文件唯一的哈希指纹已确保文件内容不会被篡改，但是无法保证 CID 的真实性。无法信任获得的 CID 所代表的文件是真实的未经篡改的域名文件，因此增加了两层签名机制，利用非对称密钥系统的身份验证功能进行签名。首先上一级文件的私钥对下一级文件进行签名，这样在拿公钥验签时可以确保下一级文件公钥的真实性和文件完整性，多级文件签名连成信任链。顶级域文件再用自己的私钥对自己的资源记录文件签名，每个顶级域名都有一对密钥对，公钥是对外公开且经过根域验证的，以此能验证此子域文件的真实性。

本文由第三方为顶级域名签发一对密钥对，由私钥对顶级域文件中的资源记录签名，由公钥验证文件的完整性。私钥由第三方安全的存储，公钥则是对外公布用于验签。顶级域公钥的真实可靠性则由其上一级根文件来确保。第三方同样为根文件签发一对非对称的密钥对，根文件私钥对文件中的内容，即每一个顶级域及其对应资源记录文件的公钥进行签名，以此保证顶级域文件的真实性。同时顶级域私钥还需将顶级域名与所对应文件 CID 签名，这样可保证域名与文件一一对应。由根文件公钥为

验证入口的双层签名能有效地保证资源记录文件的来源真实性和文件完整性。根文件的私钥由可信第三方秘密保存,公钥则是公布在外用于验签。

从 IPFS 中获取全部的顶级域资源记录文件,对每一项执行 2 次签名操作,得到一个五元组(DomainName, DomainHash, DomainPubkey, PubkeySign, HashSign),结构见表 4。其中,DomainName 为域名,DomainHash 为域名在 IPFS 系统中对应的资源记录文件的哈希指纹,DomainPubkey 为 root_file 的公钥,PubkeySign 是为了保证顶级域名公钥真实性产生的签名,HashSign 是域名保证哈希指纹真实性产生的签名。

表 4 root_file 文件结构

Tab. 4 root_file file structure

变量名	类型	含义
DomainName	string	域名
DomainHash	string	域名哈希指纹
DomainPubkey	string	域名公钥
PubkeySign	string	root 保证域名公钥真实性的签名
HashSign	string	域名保证域名哈希指纹真实性的签名

顶级域名 DomainName 对应的公钥为 DomainPubkey,私钥为 DomainPrikey。root_file 的私钥 RootPriley,公钥 RootPubkey 作为验签的入口公布在网络上。PubkeySign 的形成是由 root_file 的私钥 RootPriley 对 DomainName 和 DomainPubkey 进行签名,得到签名结果 PubkeySign;HashSign 的形成是由 DomainPrikey 对域名 DomainName 和域名哈希指纹 DomainHash 进行签名,得到签名结果 HashSign。私钥都是由可信第三方秘密保存的,公钥均对外公布,用于验证。

两层签名形成信任链,如图 5 所示。用户从可信的第三方获取根文件的 CID 和根文件公钥,由此开始可验证文件真实性。根据 CID 可以找到用 RootPriley 签名的 root_file,用得到的根文件公钥 root_pub 验证根文件。验证成功后显示完整的 root_file 文件。

2.2 验证机制

各文件间依赖关系如图 6 所示。在 root_file 文件中,由于 root_file 的私钥 RootPrikey 对 root_file 文件的哈希指纹签名,因此 root_pub 可以对 PubkeySign 进行验签,进一步得到完整产生的 root_file 文件。再由 root_file 的私钥 RootPriley 对 DomainName 和 DomainPubkey 签名,因此 root_pub 可以对 PubkeySign 进行验签。域名私钥 DomainPrikey 对 DomainName 和 DomainHash 进行了签名,则域名公钥 DomainPubkey 可对 HashSign 验签。2 次签名验证成功后可根据 DomainHash 在 IPFS 网络中查找 top_level_file,用 DomainPubkey 对完整 top_level_file 验签,成功后显示 top_level_file 文件并从中获取域名解析数据。

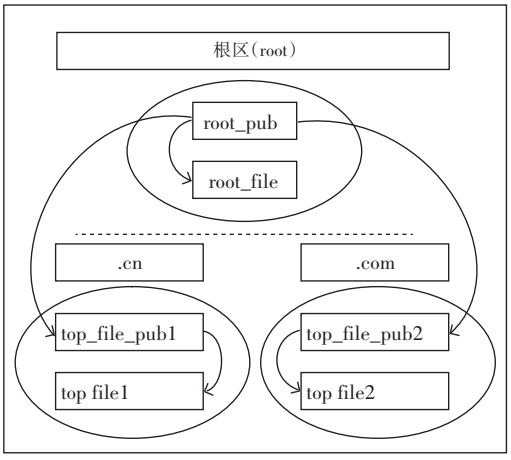

图 5 信任链

Fig. 5 Chain of trust

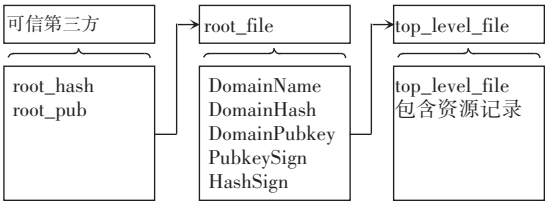

图 6 文件间依赖关系

Fig. 6 Dependency between files

3 实验结果

为了验证该方案的可行性,本文实验从 3 个方面进行:只查询根文件,直接通过文件 hash 值从 IPFS 网络实现 DNS 解析,以及经过私钥签名后验签再进行 DNS 解析。

对于 IPFS 网络的域名解析,本文使用一个连入 IPFS 网络节点的客户端进行 DNS 报文查询。由于最初搭建节点时是从不同节点分散上传解析文件的,因此使用一个从未上传过域名解析文件的 peer 节点进行查询操作。在该节点上,对访问量较高的 5 000 个包含不同顶级域的域名进行一千余次域名解析实验,并记录每次域名解析所需时间。分别采用如下 3 种方式:只查询根文件、直接通过文件 hash 值从 IPFS 网络实现 DNS 解析,以及经过私钥签名后验签再进行 DNS 解析。

将 3 种实验所用时间画成同一张折线图,如图 7 所示,研究从中随机选取了 sina.cn 这个域名进行 3 种实验的时间对比。实验结果表明基于 IPFS 网

络带签名验证机制的一次域名解析时间在 1.2 s 上下波动，波动范围为 0.3 s。解析时间明显增加，增加的时间主要是验证签名产生的时延。带签名验证机制的 IPFS 域名解析需要 4 次验签过程：根文件公钥对根文件哈希值验签、根文件公钥对 PubkeySign 验签、顶级域名公钥对 HashSign 验签，以及顶级域名公钥对顶级域文件验签。因此解析一次域名时间明显变久。

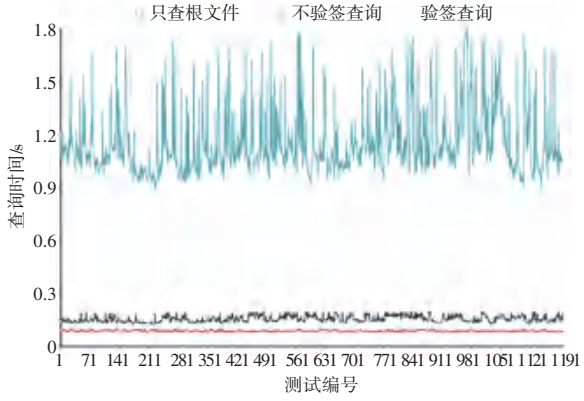

图 7　域名解析实验结果

Fig. 7　Domain name analysis experiment results

对于直接读取根文件和不验证签名的解析结果放大后如图 8 所示。

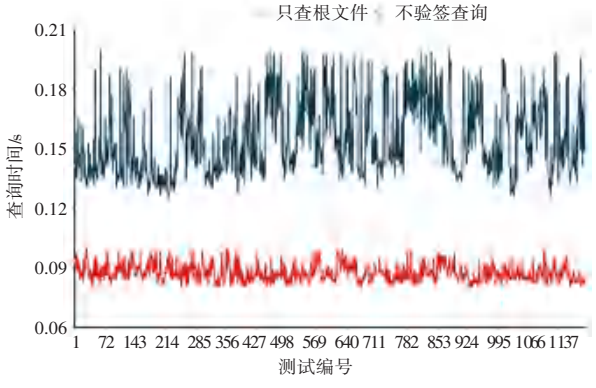

图 8　不进行签名验证的查询时间

Fig. 8　Query time without signature verification

测试结果表明基于 IPFS 网络的一次域名解析时间在 0.16 s 上下波动，波动范围上下约为 0.03 s，对于同一域名的多次查询有时间波动，说明该波动为 IPFS 网络状况对域名解析造成的影响。域名解析所产生的时间主要由在 IPFS 上下载完整文件和对文件进行解析两部分构成。每次查询文件都需要通过 IPFS 网络在节点上查找本地节点是否有该文件，有文件则直接返回完整文件再进行域名解析。在 IPFS 网络中，即使文件在本地有缓存，但每次查询时还是会问一次 IPFS 节点网络，经由 IPFS 网络节点判断文件在本地、还是对等点。因此对于 IPFS 网络来说，无论文件在不在本地节点，都会访问 IPFS 网络，这个过程会和当时的网络情况有关。本地节点不存在，则通过 DHT 路由表在对等节点间查询。所有的域名解析均有以上的过程。

结果表明本系统在域名解析方面确实可行，但花费时间较多。这也是 DNSSEC 迄今为止仍不能完全部署的原因。区块链技术虽然安全，但读取数据所用时间代价仍然较大。

4　结束语

随着互联网规模日益增大，网络复杂度也越来越高。高度中心化的 DNS 系统越来越脆弱。为解决 DNS 的安全问题，专家学者们也都提出很多的改进方案，但大多是在现有 DNS 协议基础上进行修改，不能彻底解决中心化的问题。例如 DNSSEC，同样基于密码学理论和算法来保证解析数据的权威性和完整性，但却带来了非常严重的性能问题。并且 DNSSEC 目前只在根域和二级域进行了部署，想要覆盖全部 DNS 系统十分困难。而本文基于分布式的 IPFS 技术，签名验证机制覆盖全部解析过程，所产生的时延也在可容忍范围内，可以有效解决 DNS 的安全问题。目前验签根文件的公钥是从某一可信第三方获得，今后工作会将 IPFS 网络与 Fabric 区块链相结合，Fabric 作为可信第三方，确保根文件公钥的真实性，能够进一步解决域名解析的安全问题。

参考文献

[1] 邹学强，杨海波. 从网络域名系统管理权看国家信息安全[J]. 信息网络安全，2005(9)：23.

[2] BENET J. IPFS - Content addressed, versioned, P2P file system [J]. arXiv preprint arXiv：1407.3561, 2014.

[3] 殷龙，王宏伟. 基于 IPFS 的分布式数据共享系统的研究[J]. 物联网技术，2016，6(6)：60.

[4] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system[J]. Acm Sigops Operating Systems Review, 2003, 37(5)：29.

[5] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and smart contracts for the Internet of Things[J]. IEEE Access, 2016, 4：2292.

[6] CHEN Yongle, LI Hui, LI Kejiao, et al. An improved P2P file system scheme based on IPFS and Blockchain[C]// 2017 IEEE International Conference on Big Data (Big Data). Boston, MA, USA：IEEE, 2017：2652.

[7] KRYZA B, KITOWSKI J. Hypergraph based abstraction for file-less data management[M]// WYRZYKOWSKW R, et al. PPAM 2015, Part I, LINS 9573. Switzerland：Springer International Publishing, 2016：322.

[8] CONOSCENTI M, VETRO A, MARTIN J C D. Peer to peer for privacy and decentralization in the Internet of Things[C]// IEEE/ACM International Conference on Software Engineering Companion. Buenos Aires, Argentina：IEEE, 2017：288.