Freenet 匿名上网方法原理及分析

孙蕗 [1)]

[1)]（南开大学网络空间安全学院，天津，300350）

摘 要

Freenet 是一种旨在实现匿名和抗审查的分布式网络系统，允许用户在不暴露身份的情况下自由分享和获取信息。本文详细介绍了 Freenet 的具体用途、组织结构、工作原理及其所采用的核心算法。Freenet 的主要用途包括匿名发布和检索文件、创建和访问匿名网站，以及抵御信息审查。Freenet 采用了去中心化的节点网络，使用数据存储和路由算法确保信息的分散和安全。其关键技术包括哈希表、随机路由、数据分片和冗余存储，以保证数据的可靠性和匿名性。本文通过分析 Freenet 的设计和实现，展示了其在提供匿名和抗审查互联网访问方面的独特优势和挑战。

关键词 Freenet；匿名网络；分布式网络；数据路由

中图法分类号 TP DOI 号: * 投稿时不提供 DOI 号

Principles and Analysis of Freenet Anonymous Browsing Method

Sun Lu[1)]

[1)](College of Cyber Science, Nankai University, Tianjin ,300350)

Abstract Freenet is a distributed network system designed to achieve anonymity and resist censorship, allowing users to freely share and access information without revealing their identities. This paper provides a detailed introduction to Freenet's specific uses, organizational structure, working principles, and the core algorithms it employs. The main applications of Freenet include anonymous publishing and retrieval of files, creation and access of anonymous websites, and resistance to information censorship. Freenet utilizes a decentralized network of nodes and employs data storage and routing algorithms to ensure the distribution and security of information. Its key technologies include hash tables, random routing, data fragmentation, and redundant storage to ensure data reliability and anonymity. By analyzing the design and implementation of Freenet, this paper demonstrates its unique advantages and challenges in providing anonymous and censorship-resistant Internet access.

Keywords Freenet; Anonymous Network; Distributed Network; Data Routing

1 引言

1.1 研究背景与意义

随着信息技术的快速发展，网络应用爆炸式的增长，互联网成为了全世界交换信息和共享资源的平台，这也对在线隐私保护带来了新的挑战，隐私泄露问题给互联网用户带来了巨大的困扰，也开始威胁到了社会利益和人身安全。现有加密机制可以保证通信数据的安全，但是不能隐藏通信关系，即使攻击者无法获取确切的通信内容，他们仍然可能利用通信协议中的缺陷侵犯用户的隐私，匿名通信技术为解决此类问题而产生。匿名通信[1]是一种通过采用数据转发、内容加密、流量混淆等措施来隐

藏通信内容及关系的隐私保护技术。为了提高通信的匿名性，这些数据转发链路通常由多跳加密代理服务节点构成，而所有这些节点即构成了匿名通信系统。匿名通信系统是一种建立在应用层之上结合利用数据转发、内容加密、流量混淆等多种隐私保护技术来隐藏通信实体关系和内容的覆盖网络[2]，可以向普通用户提供 Internet 匿名访问功能以掩盖其网络通信源和目标，向服务提供商提供隐藏服务机制以实现匿名化的网络服务部署。作为匿名通信系统的核心功能，隐藏服务机制通常利用多跳反向代理或通过资源共享存储来掩盖服务提供商的真实地址，可以保证匿名服务不可追踪和定位[3]。

根据匿名通信系统的延迟性能可以分为低延迟和高延迟系统[4]。高延迟匿名通信系统源于 1981 年由 Chaum[5]提出的 Mix 技术。Mix 函数的输入为经过加密的填充消息，Mix 节点对收集到的一批消息进行处理并将处理后的部分或全部消息进行解密和转发，以达到隐藏通信关系和通信内容的目的。Mix 技术需要保证 Mix 节点一定是可靠可信的，形成了 MixNet[6]。MixNet 中消息经过了一系列 Mix 节点组成的链路，只要链路中的 Mix 节点有一个是可信的就可以提供输入和输出之间对应关系的保密性。MixNet 对输入的消息进行加密和混淆，使得攻击者无法追踪消息的网络传播过程，Mix 节点隐藏了各个消息之间的输入和输出关系，可以防止攻击者对输入和输出消息进行关联。MixNet 主要用于对时效性要求不高的场景，如高延迟匿名电子邮件转发服务等

相对于高延迟匿名系统，低延迟系统因其低延迟性受到了更多的关注。在低延迟系统中，又可根据网络结构进一步分成 P2P 匿名网络和非 P2P 匿名网络，而其中 P2P 匿名网络进一步包含结构化与非结构化 2 种网络模型。结构化 P2P 匿名网络如 Tor[7]等，采用洋葱路由或类洋葱路由协议进行通信。非结构化 P2P 匿名网络如 FreeNet[8]等主要采用基于 DHT(Distributed Hash Table) 路由协议和随机游走协议。

2 FreeNet

Freenet[8] 是学者 Ian Clarke 于 1997 年在爱丁堡大学开始的研究项目。它是一个分布式匿名信息存储与检索系统，用户可以在匿名的情况下通过该系统实现文件上传、下载以及匿名文档检索功能[9]。其主要设计目标是 (1) 保护文件作者与读者的匿名性; (2) 文件存储者的否认的权利; (3) 高效的信息路由和分布式的文档存储; (4) 消除所有的单点控制和单点故障[10]。

与许多其他 P2P 应用程序不同，Freenet 本身并不提供全面的功能。Freenet 是一个独立的网络，用户可以通过 Freenet 匿名的分享文件、浏览和发布匿名 Web 站点、在论坛中发帖等，不用担心被审查。Freenet 不是代理服务器，因而并不能像 Tor 一样允许匿名地访问网络；它没有中央服务器，实现了完全的去中心化。它支持匿名文档存储和检索，使用了本地存储的可否认性保护用户隐私，每个分散的节点只存储一小部分加密数据。Freenet 通过使用高效的分布式存储和路由提高系统的可靠性和安全性，其节点之间的通信是加密的，如果想要了解请求者在请求某些信息以及请求的内容是什么，是极其困难的[11]。Freenet 可以在用户上传、请求和搜索资源时保障其匿名性能够抵抗第三方对可访问信息的破坏。

2.1 组织结构

Freenet 是一个以 Internet 为基础的覆盖网络，下图是这一概念的图形化描述，上层 Freenet 中节点标签表示节点的位置[12]。

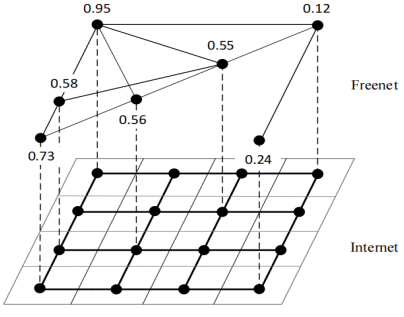

图 1 Freenet 覆盖网络模型

Freenet 参考了小世界模型[13-15]，拥有半结构化的网络拓扑。Freenet 使用逻辑上的位置来组织网络中的节点，位置是一个 0 到 1 之间的实数，分布在一个周长为 1 的圆周逻辑环上，其中 0 和 1 视为同一个位置。

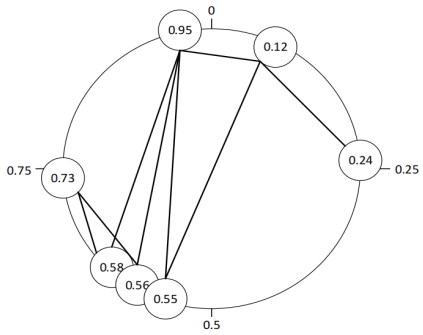

图 2　Freenet 网络拓扑图

两个节点的距离可视为圆周上两点的弧长的最小值，最远距离为 0.5。节点会在首次启动时随机生成一个位置值，位置变动将会影响网络中文件的存储和检索，如果不手动更改后续将不再改变。FreeNet 定义与节点直接相连的节点被称为节点的邻居节点，其中距离不超过 0.01 的邻居节点称为近邻居，网络中每个节点都拥有大量近距离的邻居节点和少量远距离的邻居节点，以满足小世界范式[14]。

2.2　原理分析

2.2.1　网络连接模式

节点可以自由的加入和离开网络，Freenet 提供了两种不同的连接模式：Opennet 模式，Darknet 模式[16]。

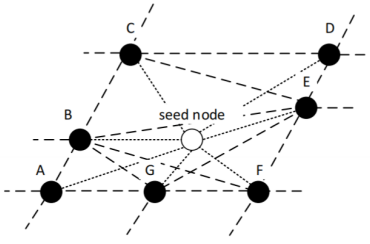

图 3　Opennet 模式

Opennet 模式中，为了使任意节点能够方便的加入网络，Freenet 提供了一组种子节点。新节点可以通过向种子节点发送请求连接到 Freenet 上的其他节点，种子节点与普通节点相比，多了转发连接请求的功能，每个节点都可以通过修改配置的方式成为种子节点。

当节点在位置 L 启动或当节点仍然需要更多邻居时，携带该节点引用文件内容的宣告消息将被发送到种子节点，如上图中的点线所示。沿着消息转发路径的节点如果愿意和请求节点建立连接，它们可以通过与请求节点交换引用文件的方式成为邻居，节点间通过发送请求连接到 Freenet 上的其他节点如上图中虚线所示。假定宣告消息被路由到位置 L，则该请求经过的大多数节点可能接近位置 L。半结构化的 Freenet 拓扑结构极大地改善了 Freenet 中数据消息的路由和查找，对节点的加入、离开和出现故障都具有很强的弹性。

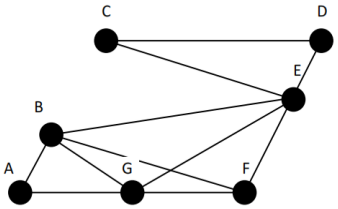

图 4　Darknet 模式

与 Opennet 相比，Darknet 连接模式下节点只与信任的朋友建立联系，Darknet 节点只对它们的朋友可见，它构建了一个 Friend to Friend (F2F) 网络，如上图所示。Darknet 模式通过手动交换引用文件建立连接。构建一个纯粹的 Darknet 需要一个符合真实社交关系的网络拓扑，并且拥有合理数量的用户才能工作。

2.2.2　分布式存储原理

一个 Freenet 节点就是运行在一台主机上的一个守护进程，它有两层抽象，如下图所示。底层抽象称为节点（Node），负责处理 Freenet 中的网络 IO（UDP 报和 FNP 消息）；上层抽象称为客户端（Client），负责提供用户和 Freenet 网络交互的接口。

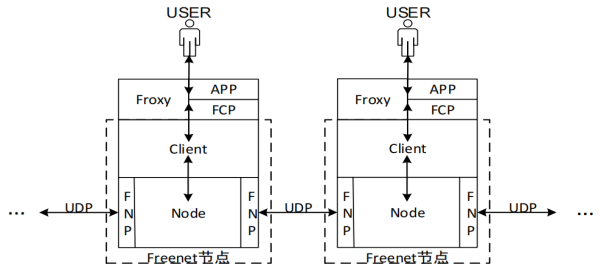

图 5 Freenet 节点

Freenet 中的资源空间类似于一个大型的分布式数据存储设备。每个成员节点需要贡献一定的磁盘空间和网络带宽，用来存放文件和代理请求。成员节点使用自己计算机中的部分硬盘来存储已被加密的文件片段，这些数据默认会保存在客户端安装路径的 datastore 文件夹[11]，每一份文档都有一个与之唯一关联的键。键（Key）由二进制字符串组成，既用于决定文档在网络中的存储位置，也用于在传输过程中验证文档完整性。

Freenet 提供四种不同的键用于映射文件。文件标识符主要分为内容 Hash 键值 (CHK) 和签名子空间键值 (SSK)，Content Hash Key (CHK)，可更新子空间键 (USK) 与关键字签名键 (KSK) 是 2 种特殊的签名子空间键值。内容散列键，是由文件的散列值生成的，相同的文件会产生相同的 CHK；Signed Subspace Key (SSK)，签名子空间键，由随机源产生，分为 requestURI 和 insertURI 两部分，只有持有 insertURI 的用户可以向网络中插入文件，只有知道 requestURI 的用户才可以从网络中检索到对应文件；Keyword Signing Key (KSK)，关键字签名键，是由人类可读的字符串经过散列得到的，方便在 Freenet 中检索资源；Updateable Subspace Keys (USK)，可更新子空间键，是对 SSK 的友好封装，功能是在进行检索请求时总是获取到最新版本的 Freesite。

CHK 是最基础的一种键值，适合静态文件资源使用，如 MP3 或 PDF 文档。这种键值包括文件的 Hash 值、解密密钥和加密设置等信息，其格式为“CHK@Hash 值，解密密钥，加密设置”。CHK 通过对二进制文件内容进行 SHA256 Hash 运算生成一个由文件内容决定的 Hash 值，该 Hash 值也作为对应文件在 Freenet 中的索引，生成该索引的同时会生成一个随机密钥用来对文件内容进行对称加密，文件被加密存储在 Freenet 中。两个不同内容的文件不可能有相同的 CHK，一个 CHK 键值指向一个唯一的文件，即使文件内容只是改变了一点点，文件的散列值也会随之变化。键值中的加密设置主要包括使用的加密算法等信息。

网站一般需要更新、添加、删除或修正其内容，SSK 通常用在经常更新的网站如 Freesite 等上。SSK 的作用就是让其他人无法冒充网站的拥有者来创建并上传一个新版本，可以保证只有文件所有者才能更新。SSK 键值由 5 部分组成：公钥 Hash、文件解密密钥、加密方式、文件描述符（由文件上传者制定的一个单词或句子）以及文件当前版本号，其格式为“SSK@ 公钥 Hash, 文件解密密钥, 加密方式/文件描述符-版本号”。文件所有者在上传 SSK 类型的文件时，首先需要随机生成一对公私钥[17]，标识键值公钥 Hash 用于在 FreeNet 中索引数据，其中私钥将用来对文件密文进行签名从而提供完整性校验，公钥用来验证签名。另外，文件所有者需要提供一个文件对称加密密钥，用于文件的加解密，并将原始文件加密后的密文、利用私钥对密文的签名以及公钥数据，共同存储在节点中，以便用户对其进行校验。Freesite 拥有者还可以为该文件指定一个简短的文本字符串作为文件描述符，同时用版本号指明了当前文件的版本，用以区分文件的不同版本。

USK 与 KSK 是 2 种特殊的 SSK。可更新子空间键 USK 主要用于链接到最新版本的 SSK, 这种键值本质上是对 SSK 进行了一层封装，向用户隐藏了对于最新版本 SSK 的搜索过程，使其更人性化。这种键值格式为“USK@ 公钥 Hash, 文件解密密钥, 加密方式/文件描述符/版本号/”。Freenet 中的节点会储存它已知的 USK 版本号列表，而并不储存所有的数据。该列表在上一次下载该文件的时候被创建，并在后台尝试请求它们。当访问 USK 文件，节点会从列表中查询是否有当前版本或更高版本，如果找到新版本，则直接返回最新版本的网站，然后会在后

台继续搜索未知的已更新的版本并添加到 USK 注册表中，以供下次访问该网址时使用。

关键字签名键 KSK 是简化的 SSK, 仅仅由一个描述性的文本字符串构成，其格式为“KSK@ 文本字符串”。当选择向 Freenet 中上传这种键值类型的文件时，用户只需要提供一个描述性的文本字符串。节点首先根据用户提供的字符串生成一对公私钥，然后同样对公钥进行 Hash 产生该文件的索引，私钥则用来对文件进行签名以提供一定的完整性校验。文件最终利用该文本字符串作为对称密钥加密后与签名一起存储在 Freenet 中。

Freenet 中的节点能够自主选择其用于共享资源的本地存储空间的大小。每个节点本地的空间都是通过最近最久未使用（least recently used, LRU）的方式来控制的, 文件以最近一次被请求的时间点（未被请求过的文件即为上传时间）按照从新到旧来排列。当一个新文件的存在（可能由上传或请求新文件的行为导致）使得空间内的数据量大于之前所配置的大小时，最近最少使用的文件将被按顺序删除，直至产生足够的剩余空间来存储最新的文件[13]。

在 Freenet 中键、文件和节点的关系可以描述为：键和文件映射，键可以用来定位节点，文件存储在通过键定位的节点中。键使用位置定位节点, 每一个键都可以计算出一个位置。将位置定义为键的一个属性，称为键的位置，文件会被存储在一簇和键的位置相近的节点中。如下图所示，<Keys, Files, Locations> 是一个节点内存储信息的三元组，Keys 是一个节点内存储过的所有键的集合, Files 是 Keys 中键对应的文件的集合, Locations 是 Keys 的位置的集合，那么 <KA, FA, 0.54～ 0.56> 就可以描述为位置为 0.55 的节点 A 存储了位置范围为 0.54～ 0.56 的键集合 KA 对应的文件集合 FA。Freenet 通过位置来组织文件和节点，节点倾向于存储与之位置相近的键对应的文件。

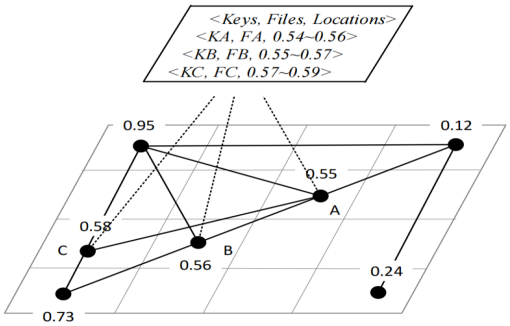

图 6　键、文件和节点的关系

2.2.3　文件路由过程

Freenet 是一个点对点的网络，每个独立自治的节点仅会知道其最近的邻居，对整个网络的组成结构是不知道的。不同于 Gnutella 中的泛洪机制，Freenet 文件路由过程中使用的算法为带回溯的基于位置的深度优先贪心搜索算法。网络中的节点使用有限的位置信息来做出最佳的路由决策，每个节点可以获得其一跳范围内节点的物理 IP 地址和两跳范围内节点的逻辑位置地址。

路由过程中节点会通过综合比较其两跳范围内的邻居节点和请求中键的距离来选择下一跳节点。Freenet 中的文件路由请求分为插入请求和检索请求两类，在 Freenet 中数据有两种本地保存形式，深存储和浅存储，插入请求可以触发深存储，检索请求则仅缓存数据。

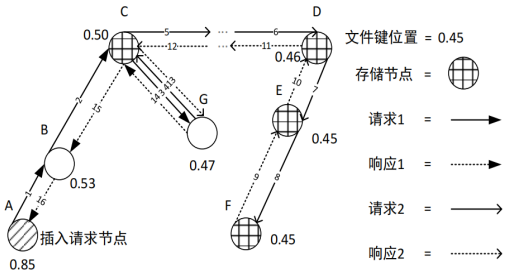

图 7　向 Freenet 插入文件

向网络中存储数据首先节点计算出文件对应键的位置，然后请求按照基于位置的深度优先搜索算法转发, 最终将收敛到一簇与键位置相近的节点, 这些节点将作为数据的初始存储节点，数据将随着后续的文件检索请求扩散。插入文件的过程如上图所

示。

当用户要上传一个新文件时，需要先生成相应的二进制标识键值，并生成一个上传插入的请求到用户自身的节点上，插入请求的消息中包含 Unique Identifier (UID) 和 hops-to-live (HTL) 字段，UID 用于防止消息陷入环路，HTL 用于在文件被上传前确保无键值冲突的节点数量，防止消息无限转发[10]。节点收到上传请求的信息后，首先要证实本地记录里是否已经存在此键值。若有，节点会反馈该键值所对应的文件，用户就会得知此次请求产生一次键值冲突，必须使用另外一个计算得出的键值。若在本地没有检索到此键值，节点会在自己的路由表里查询到一个标识键值最接近的节点，并把请求转发给该节点。若该请求转发后发现了一个键值冲突，则该节点中冲突的键值所对应的文件会被原路返回，同时存储在所有之前请求经过的节点本地，这些节点的路由表中也会新增相应的路由表项。HTL 用于 IP 中的 time-to-live(TTL) 类似，在每个节点都进行-1 操作，当一个上传请求的 HTL 字段在转发到各节点的过程中减为零时都未发生键值冲突，“all clear”会以转发的路径返回到最开始发出申请的节点，意味着该节点能够以选择的键值上传相应文件。当 HTL 小于 15 的时候插入请求开始符合可缓存条件，此时经过的节点都会缓存一份文件内容，节点存储一份文件的条件是在满足可缓存的前提下，当前节点和其两跳范围内的节点相比距离文件对应的键的位置最近。回溯过程中消息只包含 UID 字段，在到达节点 G 时同样会将响应 2 变更回响应 1，最终转发回插入请求的发起节点，至此插入过程完成。

文件请求节点发起检索请求，首先计算出文件对应的键的位置，然后选择距离键位置最近的节点作为下一跳路由，持续此过程直到找到目标文件或者 HTL 减小为 0 检索失败，最后请求到达文件的存储节点，目标文件将按照 D→E→B→A 的顺序发送给节点 A，同时节点 E、B 和 A 也会缓存该文件。文件检索的过程分为搜索和回溯两步。搜索过程中消息具有 UID 和 HTL 字段，回溯过程中消息只包含 UID 字段，每一个代理请求的节点只知道其直接前驱和直接后继，不知道存储数据的节点和发起数据请求的节点，这一特性提供了对请求者的匿名性保护。在请求成功后，被请求文件将在网络中扩散。

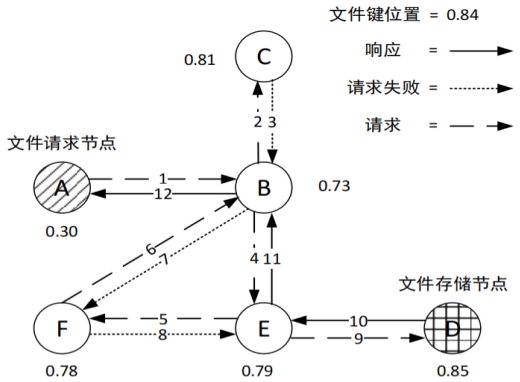

图 8 从 Freenet 中检索文件

2.3 具体用途

Freenet 的存储系统是由逻辑环上每个节点的部分存储空间共同构成的，每个节点可以根据自身的情况独立配置其存储容量。Freenet 中的文件存放在节点的数据存储目录中，由唯一的二进制文件的键值标识和管理。当其他节点需要获得某个文件时，会首先获得存储该文件的节点信息，并使用文件的键值作为标识。存储该文件的节点会分块发送给请求节点，优化传输效率和网络资源的使用，同时在网络中复制多个副本以提高系统的可靠性。这种机制不仅增加了数据的冗余度，还能在部分节点失效或离线时，确保数据的可访问性和完整性。

Freenet 不仅能够匿名的分享文件，还可以构建匿名的应用层服务，包括匿名 Web 站点 Freesite，匿名社交应用 Sone，匿名邮箱 Freemail，匿名 Internet Relay Chat (IRC) 应用 FLIP ，匿名论坛 FMS 等，由于 Freenet 的每个节点只与其相邻节点进行通信，并且不了解整个通信链路的全貌，这种设计有效地保护了用户的身份信息。节点之间的通信采用了复杂的路由算法，使得任何单一节点都无法掌握完整的网络拓扑和数据流向，从而提供了强有力的匿名性保障。这种设计不仅防止了外部攻击者的追踪，也

防止了内部节点的窥探，确保了用户在使用 Freenet 时的高度隐私和安全。

参 考 文 献

[1] 陈欢, 苏马婧, 王学宾, 等. 匿名通信综述[J/OL]. 电子技术应用, 2021, 47(04): 46-53+58. DOI: 10.16157/j.issn.0258-7998.200995.

[2] 王良民赵蕙. 网络层匿名通信协议综述[J/OL]. 网络与信息安全学报, 2020, 6(1): 11. https://www.infocomm-journal.com/cjnis/CN/abstract/article_169945.shtml. DOI: 10.11959/j.issn.2096-109x.2020006.

[3] 罗军舟, 杨明, 凌辰, 等. 匿名通信与暗网研究综述[J]. 计算机研究与发展, 2019, 56(01): 103-130.

[4] PéREZ-GONZáLEZ F, TRONCOSO C, OYA S. A least squares approach to the static traffic analysis of high-latency anonymous communication systems[J/OL]. IEEE Transactions on Information Forensics and Security, 2014, 9(9): 1341-1355. DOI: 10.1109/TIFS.2014.2330696.

[5] CHAUM D L. Untraceable electronic mail, return addresses, and digital pseudonyms[J/OL]. Commun. ACM, 1981, 24(2): 84–90. https://doi.org/10.1145/358549.358563.

[6] EDMAN M, YENER B. On anonymity in an electronic society: A survey of anonymous communication systems [J/OL]. ACM Comput. Surv., 2009, 42(1). https://doi.org/10.1145/1592451.1592456.

[7] DINGLEDINE R, MATHEWSON N, SYVERSON P F. Tor: The second-generation onion router[C/OL]//USENIX Security Symposium. 2004. https://api.semanticscholar.org/CorpusID:8274154.

[8] CLARKE I, SANDBERG O, WILEY B, et al. Freenet: A distributed anonymous information storage and retrieval system[C/OL]//Workshop on Design Issues in Anonymity and Unobservability. 2000. https://api.semanticscholar.org/CorpusID:215762961.

[9] CHE H. Review: From p2p to web services and grids: Peers in a client-server world[J/OL]. Comput. J., 2005, 48(3): 379. https://doi.org/10.1093/comjnl/bxh081.

[10] 王晓箴, 刘宝旭. Freenet 综述及 P2P 技术应用探讨[C]//第 13 届全国计算机、网络在现代科学技术领域的应用学术会议论文集. 中国科学院高能物理研究所计算中心; 中国科学院高能物理研究所计算中心;, 2007: 241-245.

[11] 郭晗. 基于 Freenet 的暗网空间资源探测技术研究[D]. 上海交通大学, 2018.

[12] 许岩岳. 基于 Freenet 网络的通信防溯源性能分析[D]. 西安理工大学, 2020.

[13] ZHANG H, GOEL A, GOVINDAN R. Using the small-world model to improve freenet performance[C/OL]//Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies: Vol. 3. 2002: 1228-1237 vol.3. DOI: 10.1109/INFCOM.2002.1019373.

[14] WATTS D J, STROGATZ S H. Collective dynamics of ‘small-world’ networks[J/OL]. Nature, 1998, 393: 440-442. https://api.semanticscholar.org/CorpusID:3034643.

[15] KLEINBERG J M. Navigation in a small world[J/OL]. Nature, 2000, 406: 845-845. https://api.semanticscholar.org/CorpusID:4425543.

[16] HALVEMAAN K. Freenet darknet mapping[Z]. 2017.

[17] CONTI M, DE GASPARI F, MANCINI L V. Anonymity in an electronic society: A survey[M/OL]. Cham: Springer International Publishing, 2016: 283-312. https://doi.org/10.1007/978-3-319-32699-3_12.