

# 加密 DNS 调研

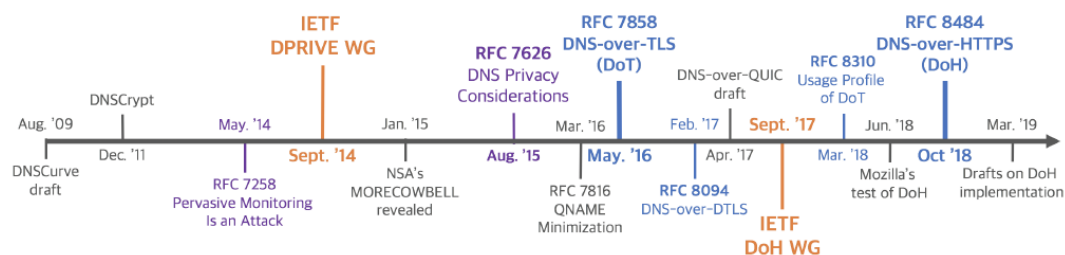
吴佳挺 李仁杰

## 目录

1	加密 DNS 背景.....	2
2	协议介绍.....	3
2.1	DNS-over-HTTPs.....	3
2.2	DNS-over-TLS .....	4
2.3	DNS-over-DTLS .....	4
2.4	DNSEncrypt.....	4
2.5	DNS-over-QUIC.....	5
3	研究方向.....	5
3.1	重点文献: .....	5
3.2	研究方向 .....	5
4	测量结果.....	6
4.1	《An Empirical Study of the Cost of DNS-over-HTTPS》 .....	6
4.1.1	DoH 现状.....	6
4.1.2	DNS 传输模式比较.....	8
4.1.3	基于 HTTP/2.0 的 DoH 和 UDP 的性能开销.....	9
4.1.4	DoH 对 DNS 解析时间和页面加载时间的影响.....	11
4.2	《An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?》 .....	12
4.2.1	服务端测量.....	12
4.2.2	客户端测量.....	14
4.2.3	真实世界中 DoE 流量测量 .....	15
4.3	《Encrypted DNS => Privacy? A Traffic Analysis Perspective》 .....	17
4.3.1	加密 DNS 指纹识别效果.....	17
4.3.2	加密 DNS 识别与审查.....	19
5	常用工具.....	20
5.1	常用的客户端部署工具 .....	20
5.1.1	DNSEncrypt-proxy.....	21
5.1.2	SecureDNS.....	22
5.1.3	DNSLookup .....	24
5.1.4	DNSProxy .....	26
5.2	DoH 相关工具 .....	27
6	公共加密 DNS 服务器.....	29
6.1	DNSEncrypt-proxy 提供的加密 DNS 服务器列表 .....	29
6.2	Curl 提供的 DoH 服务器列表 .....	30

# 1 加密 DNS 背景

自 1985 年 DNS 协议被提出以来, DNS 协议一直是互联网体系的重要组成部分。但是由于 DNS 协议在设计之初没有考虑安全问题, 采用明文传输 DNS 请求和响应, 因此产生了许多安全问题。例如, 网络审查、DNS 欺骗等。



图表 1 加密 DNS 发展历史

加密 DNS(DNS-over-Encryption)正是针对这些安全问题所提出的解决方案。加密 DNS 的发展历史如图表 1 所示。最早保护 DNS 通信的提议可以追溯到 2009 年, 即 DNSCurve 协议。2010 年 2 月 23 日, openDNS 宣布其递归解析器支持 DNSCurve。此后, DNSCurve 并未得到其他域名解析服务商的支持。2011 年 12 月 6 日, OpenDNS 宣布名为 DNSCrypt 的新协议, 它不基于标准 TLS, 而是使用 X25519-XSalsa20Poly1305 加密结构。DNSCrypt 作为加密 DNS 中最早的协议之一, 已获得数家大型公共解析器的支持, 包括 OpenDNS, Yandex 和 OpenNIC。2014 年 5 月, IETF 发布 RFC7258(Pervasive Monitoring), 将 Pervasive Monitoring 行为视为一种攻击。Pervasive Monitoring 是指规模巨大且不加区分地监控并收集信息的行为。因而对 DNS 的机密性提出了要求。同年 7 月, IETF 成立 dprive 工作组, 研究如何确保 DNS 查询的机密性以防止 Pervasive Monitoring。2015 年 1 月, NSA(美国国家安全局)的 MORECOWBELL 系统被揭露。MORECOWBELL 使美国 NTOC(Network and Telecommunications Operations Center)能够实时监控全球网站, 而 DNS 是该系统的核心。该事件进一步促进了加密 DNS 的研究。2015 年 8 月, IETF 发布 RFC7672(DNS Privacy Consideration), 描述了 DNS 相关的隐私考虑, 旨在分析当前 DNS 安全状况。2016 年 3 月, IETF 发布 RFC7816(QNAME Minimization), 核心思想是发送最少信息的 DNS 查询就能那降低用户隐私泄露的风险。同年 5 月, IETF 发布 RFC7626(DNS-over-TLS), 提出用 TLS 协议加密 DNS 信息, 防止中间人攻击。截至 2018 年, Cloudflare、Quad9

与 CleanBrowsing 均向大众提供支持 DNS over TLS 的公共 DNS 解析服务。2017 年 1 月，IETF 发布 RFC8094(DNS-over-DTLS)，使用 DTLS 来加密 DNS 请求。该协议作为 DoT 协议的后备选项，目前没有得到实际部署。同年 7 月，IETF 成立 DoH 工作组，旨在研究 DoH 协议。2018 年 3 月，IETF 发布 RFC8310(Usage Profile of DoT)，讨论 DoT 的配置文件，更新了 RFC7858。同年 2 月，Firefox 浏览器开启测试 DoH 协议。8 月，IETF 发布 RFC8484，确定了 DoH 协议标准。2019 年 3 月，DoH 服务器的部署草案被提出。

## 2 协议介绍

目前已经提出的主流加密 DNS 方案有 DNS-over-TLS(DoT)、DNS-over-HTTPS(DoH)、DNS-over-DTLS、DNS-over-QUIC(DoQ)、DNSEncrypt 等。其中 DNS-over-DTLS 是作为 DoT 的备选方案提出，在 DoT 成功应用后，也就很少出现，没有得到实际发展；而 DoQ 则依赖于 Google 提出的 QUIC 协议，由于 QUIC 协议还未得到广泛应用，DoQ 也没有得到实际发展。

### 2.1 DNS-over-HTTPS

DoH 由 RFC8484 标准化，本质是将 DNS 查询嵌入到 HTTPS 消息中，该消息受 TLS 保护。DoH 使用 URI 模板（例如 `https://dns.example.com/dns-query{?dns}`）来定位服务，如图表 2 所示，**DoH 数据包以 URI 参数（使用 GET）或 HTTP 消息正文（使用 POST）进行编码**。DoH 与 HTTPS 共享**端口 443**，该端口将 DoH 查询与其他 HTTPS 流量混合在一起，因此可以有效地阻止仅针对 DNS 的流量分析。根据设计，DoH 需要对 DNS 服务器进行加密和身份验证。与 DoT 相似，查询时间开销可能由连接建立和加密引起。**DoH 在 HTTPS 之上运行，因此特别适合于 Web 浏览器等应用程序**。通常，应用程序已经包含存根解析器，**因此 DNS 客户端使用 DoH 所需的初始配置相比于更新 OS 或安装其他软件来说很少**。Firefox 从版本 62 开始支持 DoH，并提供用于 DoH 配置的 UI。但是，对于 DNS 运营商，由于主流 DNS 软件不支持 HTTP 和 DNS 的组合，因此他们需要部署其他实施方案才能提供服务。当前，一些大型共有解析器支持 DoH，包括 Cloudflare，Google 和 Quad9。

```
GET /dns-query?dns=AAABAAABAAAAAAB2V4YW1wbGUDY29tAAABAAE HTTP/1.1
Host: dns.example.com
Accept: application/dns-message

POST /dns-query HTTP/1.1
Host: dns.example.com
Accept: application/dns-message
Content-Type: application/dns-message
Content-Length: 29
00 00 01 00 00 01 00 00 00 00 00 00 07 65 78 61
6d 70 6c 65 03 63 6f 6d 00 00 01 00 01
```

图表 2 DoH 数据中的 URI 模板

## 2.2 DNS-over-TLS

DoT 在 2016 年由 RFC7858 标准化，其概念很简单：客户端和服务端在 DNS 查找之前协商 TLS 会话，并使用它保护 DNS 查询。客户端和递归解析器可以交换加密的 DNS 消息（防止被动监视），并且解析器可以通过验证 SSL 证书进行身份验证（防止中间人攻击）。默认情况下，DoT 使用端口 853 进行通信。使用专用端口可以使 DoT 请求与其他流量区分开来。当前 DoT 已得到操作系统（如 Android 9），DNS 软件（如 Unbound 和 Stubby）和大型公共 DNS 解析器（如 Cloudflare，Google 和 Quad9）等的广泛支持。对于服务提供商而言，当前的部署方式降低了操作 DoT 解析器的成本，并且 SSL 证书易于通过 Let's Encrypt 等自动 CA 进行安装。但是，客户端使用 DoT 之前必须进行一些更改，包括切换到新的存根解析器（例如，通过更新操作系统或安装存根解析器（如 Stubby））和手动配置 DoT 解析器。

## 2.3 DNS-over-DTLS

DNS-over-DTLS 是 DoT 的一种变体，它可以在 UDP 上工作以获得更好的性能。尽管基于 DTLS 的 DNS 和 DoT 共享大多数属性，但它仅是作为 DoT 的备份建议而设计的，当前基于 DTLS 的 DNS 尚无实际实现，包括存根和递归解析器，因此，它对客户端的可用性和对 DNS 运营商的可部署性均很差。

## 2.4 DNSCrypt

DNSCrypt 在 2011 年提出，它不基于标准 TLS，而是使用 X25519-XSalsa20Poly1305 加密结构。DNSCrypt 消息是通过端口 443 传输的，该端口也

与 HTTPS 流量混合在一起，并且可以在 **UDP 和 TCP 上使用**。作为加密 DNS 中最早的协议之一，DNSCrypt 已获得数家大型公共解析器的支持，包括 OpenDNS，Yandex 和 OpenNIC。要使用 DNSCrypt，客户端需要安装其他软件（如 DNSCrypt-proxy），**服务器需要提供专用的证书**。自提议以来，DNSCrypt 从未被 IETF 标准化。

## 2.5 DNS-over-QUIC

DNS-over-QUIC 提供与 DoT 类似的隐私属性，但性能与 UDP 上 DNS 类似。根据目前的草案，它旨在最小化延迟并解决 TCP 的行头阻塞之类的问题。为了获得更好的可用性，它还提供了一种后备机制，当 QUIC 连接失败时，可以使用 DoT 或明文 DNS。DNS-over-QUIC **计划使用专用端口 784**。但是，**还没有针对 DNS 客户端或运营商的实际实现**。

## 3 研究方向

### 3.1 重点文献：

- a) 《An Empirical Study of the Cost of DNS-over-HTTPS》，IMC 2019
- b) 《An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?》，IMC 2019，段海新
- c) 《Encrypted DNS--> Privacy? A Traffic Analysis Perspective》，NDSS 2020

### 3.2 研究方向

- a) 基于 HTTP 和 TLS 调查并描述了当前 DoH 的现状，探究了 DoT 和 DoH 对延迟的敏感性，量化比较基于 HTTP/2.0 的 DoH 和 UDP 的性能开销，探究 DoH 对 DNS 解析时间和页面加载时间的影响。
- b) 对加密的 DNS 进行端到端的大规模分析的，全面地分析 DNS 加密协议的发展现状、可访问性和延时性。针对 Server 端，识别 DoT 和 DoH 解析器及证书有效性；针对 Client 端，检测客户端到 DoH 解析器的可达性和查询时延；探索 DNS 加密使用率及趋势。
- c) 研究加密 DNS 指纹。对比数据清洗前后，DNS 指纹攻击的效果；对比 web

指纹与 DNS 指纹的效果；对比训练数据规模对 DNS 指纹识别的影响；对比训练数据收集的时间对 DNS 指纹识别的影响；对比不同基础设施对收集的数据对 DNS 指纹识别的影响；评估不同抗加密流量识别技术对 DNS 指纹识别的影响

## 4 测量结果

### 4.1 《An Empirical Study of the Cost of DNS-over-HTTPS》

#### 4.1.1 DoH 现状

如错误!未找到引用源。所示，截至论文发表时，curl 项目 (<https://github.com/curl/curl/wiki/DNS-over-HTTPS>) 已经收集了很多公开的 DoH 解析服务地址，包括主流企业如 Google、Cloudflare、IBM (Quad9) 等。可以看到，DoH 解析服务对应一个 URL，大多数的 URL 路径为 /dns-query，少部分为 /family-filter 或 /，而 Google 提供了两个不同的路径：/resolve 和 /dns-query。（注：2020 年 2 月更新，Google 已删除了 /resolve）。根据 curl 项目中给出的最新的列表，对图表 3 中涉及到的 DoH 服务地址更新如图表 4。可以看到 Google 已删除了 /resolve；Cloudflare 新增了 3 个 URL 并且增加了对 Tor 的支持；Quad9 新增了两个 URL，包括一个不安全的以及一个支持安全 w/ECS 的；SecureDNS 已被移除。

Provider	DoH URL	MK
Google (i)	<a href="https://dns.google.com/resolve">https://dns.google.com/resolve</a>	G1
Google (ii)	<a href="https://dns.google.com/dns-query">https://dns.google.com/dns-query</a>	G2
Cloudflare	<a href="https://cloudflare-dns.com/dns-query">https://cloudflare-dns.com/dns-query</a>	CF
Quad9	<a href="https://dns.quad9.net/dns-query">https://dns.quad9.net/dns-query</a>	Q9
CleanBrowsing	<a href="https://doh.cleanbrowsing.org/doh/family-filter">https://doh.cleanbrowsing.org/doh/family-filter</a>	CB
PowerDNS	<a href="https://doh.powerdns.org/">https://doh.powerdns.org/</a>	PD
Blahdns	<a href="https://doh-ch.blahdns.com/dns-query">https://doh-ch.blahdns.com/dns-query</a>	BD
	<a href="https://doh-jp.blahdns.com/dns-query">https://doh-jp.blahdns.com/dns-query</a>	
	<a href="https://doh-de.blahdns.com/dns-query">https://doh-de.blahdns.com/dns-query</a>	
SecureDNS	<a href="https://doh.securedns.eu/dns-query">https://doh.securedns.eu/dns-query</a>	SD
Rubyfish	<a href="https://dns.rubyfish.cn/dns-query">https://dns.rubyfish.cn/dns-query</a>	RF
Commons Host	<a href="https://commons.host/">https://commons.host/</a>	CH

图表 3 DoH 解析服务器（MK 为表 2 中的不同类别）

Provider	DoH URL
Google	https://dns.google/dns-query DNS64: https://dns64.dns.google/dns-query
Cloudflare	https://cloudflare-dns.com/dns-query also available via Tor onion service Mozilla: https://mozilla.cloudflare-dns.com/dns-query Block Malware: https://security.cloudflare-dns.com/dns-query Block Malware and Adult Content: https://family.cloudflare-dns.com/dns-query DNS64: https://dns64.cloudflare-dns.com/dns-query
Quad9	Recommended: https://dns.quad9.net/dns-query Secured: https://dns9.quad9.net/dns-query Unsecured: https://dns10.quad9.net/dns-query Secured w/ECS Support: https://dns11.quad9.net/dns-query
CleanBrowsing	https://doh.cleanbrowsing.org/doh/family-filter/
PowerDNS	https://doh.powerdns.org
Blahdns	Finland: https://doh-fi.blahdns.com/dns-query Japan: https://doh-jp.blahdns.com/dns-query Germany: https://doh-de.blahdns.com/dns-query
SecureDNS	无
Rubyfish	https://dns.rubyfish.cn/dns-query
Commons Host	https://commons.host

图表 4 DoH 解析服务器（截至 2020.06.13）

尽管从技术上讲，有关 DoH 的 RFC-8484 中并未规定要使用的特定路径，而是将其留给服务运营商，但大多数服务仍使用/dns-query 路径，这是 RFC 中所有示例中使用的路径，使用不同的 URL 容易出错而备受争议，这也解释了为什么现在 Google 仅保留了 RFC 推荐的/dns-query 路径。



同时，不同的解析器具有不同的特征，如图表 5 所示。各个服务所支持传输内容类型不同，根据 RFC-8484，DoH 服务器必须支持 `application/dns-messagecontent` 类型；另一种主流类型为 `application/dns-json`。随着 TLS1.3 成为官方 RFC 以及 TLS1.0 等较低版本被发现安全漏洞，大部分 DoH 服务不再支持 TLS1.0 和 TLS1.1。相反，所有 DOH 服务都支持 TLS1.2，大部分支持 TLS1.3。除了 Google、Cloudware、IBM 等部分大厂支持 DoT 外，DoT 在其他服务商似乎并不受欢迎。同时，只有 Google 在支持 QUIC。DoH 依赖于 PKI 证书体系以确认 DNS 解析器身份，为了弥补 PKI 体系的已知缺点，Certificate Transparency (CT, 证书透明)、Certificate Authority Authorization (CAA, 证书颁发机构授权)、Online Certificate Status Protocol (OCSP, 在线证书状态协议)被提出。对这三者的调查发现，所有的 DoH 服务证书都支持 CT，而 Google 还提供了 CAA 记录，但没有服务支持 OCSP MS。

Feature	G1	G2	CF	Q9	CB	PD	BD	SD	RF	CH
dns-message	X	✓	✓	✓	✓	✓	✓	✓	✓	✓
dns-json	✓	X	✓	✓	X	X	✓	X	✓	X
TLS 1.0	X	X	✓	X	X	✓	X	✓	✓	X
TLS 1.1	X	X	✓	X	X	✓	X	✓	✓	X
TLS 1.2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TLS 1.3	✓	✓	✓	✓	X	✓	✓	✓	X	✓
CT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DNS CAA	✓	✓	X	X	X	X	X	X	X	X
OCSP MS	X	X	X	X	X	X	X	X	X	X
QUIC	✓	✓	X	X	X	X	X	X	X	X
DNS-over-TLS	✓	✓	✓	✓	✓	X	X	X	X	X
Traf. Steering	DL <sup>*</sup>	DL <sup>*</sup>	AC <sup>+</sup>	AC <sup>+</sup>	AC <sup>+</sup>	UC <sup>±</sup>	UC <sup>±</sup>	UC <sup>±</sup>	UC <sup>±</sup>	AC <sup>+</sup>

<sup>\*</sup> DNS Load Balancing    <sup>+</sup> Anycast    <sup>±</sup> Unicast

图表 5 不同服务的特征

4.1.2 DNS 传输模式比较

该论文对 DoT 和 HTTP/1.1 和 HTTP/2.0 两种协议下的 DoH 传输效果进行了对比。在受控环境下，通过部署本地 CoreDNS 解析器，并排除了传输协议无关的因素（如 DNS 缓存、查询速率分布、域名分布等）后，对 UDP、TLS、HTTP/1.1、



HTTP/2.0 四种模式的解析时间进行观察分析。测试结果如图表 6 所示。Baseline 是第一次测量所得的没有延迟情况下的解析性能基线；Delayed 为第二次测量，每 25 个查询中选择一个延迟 1000ms 的情况，用来观察解析时间的延迟是否会影响后续解析。

实验结果表明，没有延迟的基准情况下，UDP 和 TLS 都在不到 1ms 的时间内提供了查询响应；HTTP/2.0 始终在不到十毫秒的时间内；而 HTTP/1.1 由于浏览器对于流水线管道支持的问题，解析时间存在明显的波动。

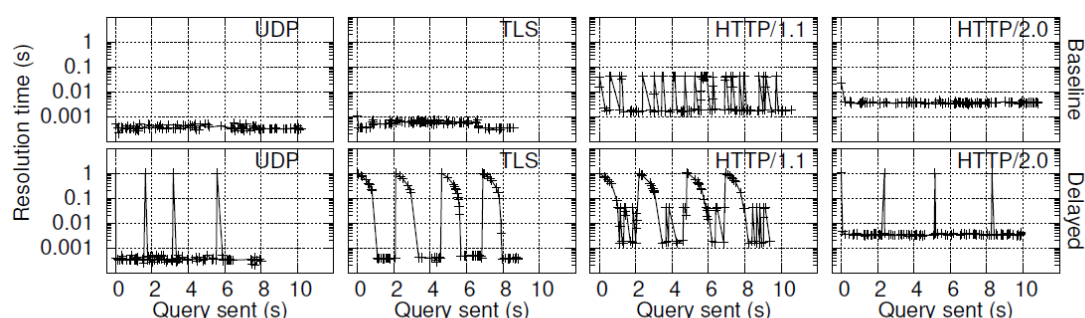


Figure 2: Impact of head-of-line-blocking on resolution times for DNS over different transport protocols. The upper charts depict the baseline and the lower ones the effect of a delay (1000ms for one in 25 queries).

图表 6 基于四种传输模式的 DNS 解析性能对比

在存在延迟的情况下，可以看到基于 UDP 的 DNS 后续解析几乎不受延迟的影响，这是由于 UDP 的无连接使得各个查询独立；TLS 协议下的延迟对后续查询具有连锁效应，这是因为 TLS 连接的按序交付意味着仅在对延迟查询的答复之后才发送对后续查询的答复。而实现无序传输的 DoT 十分复杂，支持的服务商也非常少。HTTP/1.1 与 TLS 类似具有连锁效应，因为 RFC 中要求按序交付请求；而 HTTP/2.0 与 UDP 类似，延迟对后续解析没有影响。这解释了为什么 DNS-over-HTTPS/2.0 比 DNS-over-TLS 更受欢迎，并且 RFC 中也将 HTTP/2.0 最为最低标准。

### 4.1.3 基于 HTTP/2.0 的 DoH 和 UDP 的性能开销

DNS-over-HTTPS/2.0 拥有显著优势，但其会引入附加层，因此增加了头部和开销。为了研究真实环境 DoH 的开销，论文针对 Alexa 全球排名前 100,000 网页，收集了在访问这些网站的过程中解析的所有域名，共发送了 2,178,235 个 DNS 解析查询，对应 281,414 个唯一的域名，并在本地存根解析器记录所有查询。之

后分别使用 Google 和 Cloudflare 的解析器，通过基于 UDP 的常规 DNS 和 DNS-over-HTTPS/2.0 解析这些域名。在 HTTP/2.0 中分别使用了持久连接（HP）和非持久连接（H）。

如图表 7 所示，基于 UDP 的 DNS 和 DoH 两种 DNS 请求的大小以及数据包数量区别明显，UDP-based DNS 请求仅消耗了 182 bytes 和 2 个数据包；而单 DoH 解析（H）的中位数在 Cloudflare 和 Google 分别为 5737 bytes、27 个数据包和 6941 bytes、31 个数据包。持久连接（HP）可以分摊许多已发送请求的开销，但开销仍然比 UDP-based DNS 的 4 倍还多。其中 Google 的开销比 Cloudflare 多是因为 Google 服务器的交易量更大，需要维护更多 TLS 连接，且 Google 的证书更大。

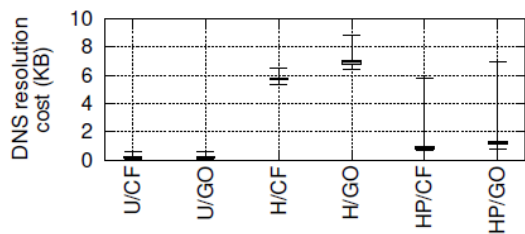


Figure 3: Total bytes per resolution. Domain names were resolved via UDP-DNS (U), DNS-over-HTTPS without persistent connection (H) and with a persistent connection (HP). The DNS servers of Cloudflare (CF) and Google (GO) were used. Whiskers span the full range of values.

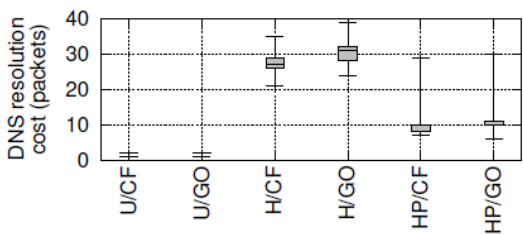


Figure 4: Total packets per resolution. Domain names were resolved via UDP-DNS (U), DNS-over-HTTPS without persistent connection (H) and with a persistent connection (HP). The DNS servers of Cloudflare (CF) and Google (GO) were used. Whiskers span the full range of values.

图表 7 UDP-based DNS 和 DoH 开销对比

如图表 8 所示，将 DoH 开销分解到协议各层。可以看到 4 个 DoH 服务的 body 层开销分布类似，尽管在极端情况下谷歌倾向于发送稍微大一些的 body。每额外增加一层，就会增加至少与原始 DNS 有效负载相同大小的开销。值得注意的是，即使单单 TLS 、TCP 的开销就相当于 UDP-based DNS 的全部开销。对于 HTTP/2.0 可以看到持久连接可以显著减少各层开销，这是由 HTTP/2 的差分标头功能引起的，在顺序请求和答复中，该功能保证仅传输已更改的标头。

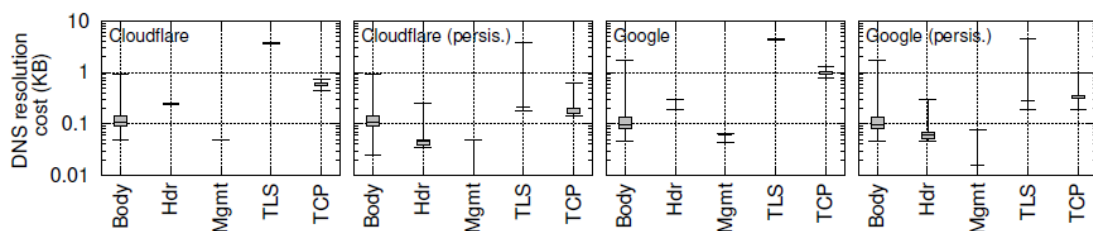


Figure 5: Overheads per DNS resolution for DNS-over-HTTPS/2. First two columns show sizes for (HTTP) bodies and headers exchanged. Mgmt refers to messages being exchanged to maintain the HTTP/2 connection like settings and windows updates. TLS and TCP refer to sizes of the respective layers.

图表 8 DoH 开销分解

#### 4.1.4 DoH 对 DNS 解析时间和页面加载时间的影响

基于 Firefox 和 sitespeed.io，对 Alexa top1000 网页进行测试，衡量 DNS 解析时间和网页的加载时间。设计了 5 种场景：本地基于 UDP 的 DNS (U/LO)、Cloudflare UDP-based DNS (U/CF) 和 DoH (H/CF)、Google UDP-based DNS (U/GO) 和 DoH (H/GO)。结果如图表 9 所示，右边两个图为在 PlanetLab 上进行相同实验的结果。可以看到，通过 UDP 请求 Google 和 Cloudflare 比请求本地解析器用时短；引入 DoH 后确实会增加解析时间，而 Cloudflare 的速度也比 Google 快一点。然而页面加载时间之间几乎没有区别。需要注意的是，整体页面加载时间比 DNS 解析时间快，这因为浏览器并行发送请求，而 DNS 图中显示的累积 DNS 解析时间没有并行性。

实验表明，切换到 DoH 并不会显著增加页面加载时间。这意味着在不牺牲用户体验的情况下，使用 DoH 来获得更好的安全性是可行的

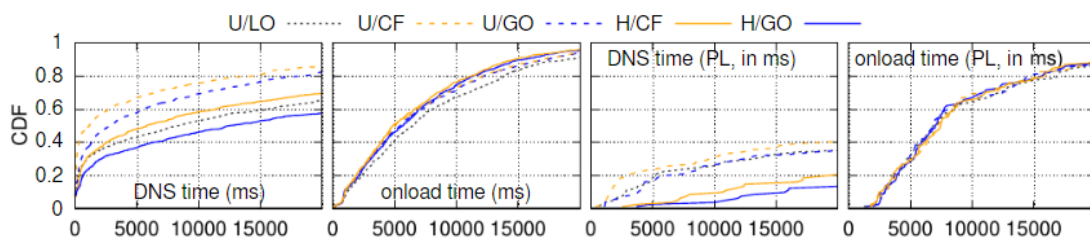


Figure 6: CDF of DNS resolution and page load times (time of onload event): U/ indicates legacy resolver, H/ indicates resolution via DoH, /LO indicates local resolver, /GO indicates Google and /CF indicates Cloudflare.

图表 9 页面加载时间对比

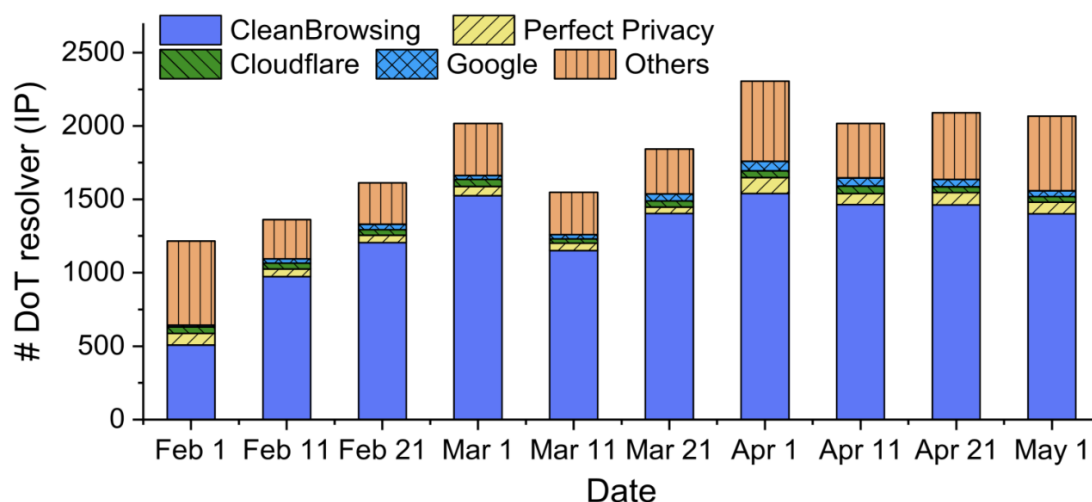
## 4.2 《An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?》

### 4.2.1 服务端测量

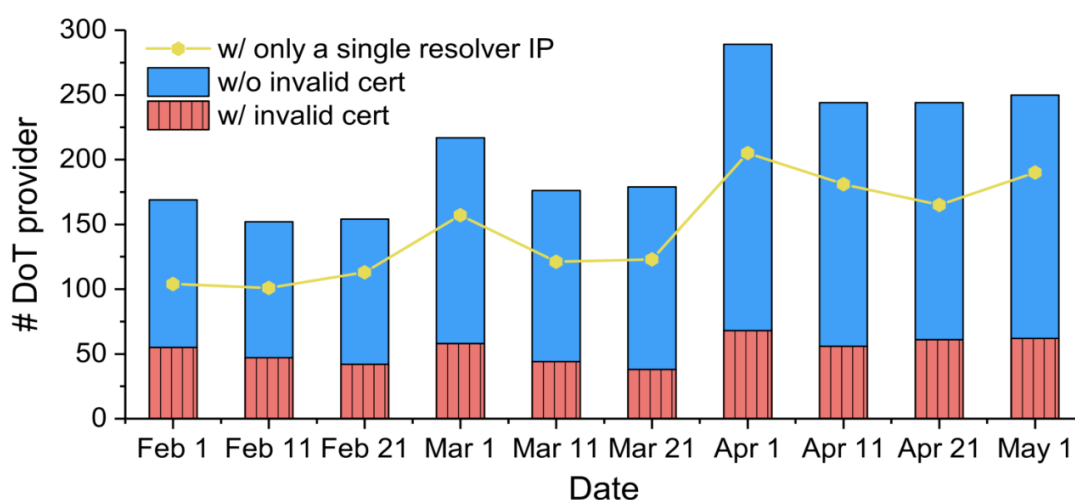
**发现公开的 DoT 解析服务器：**使用 ZMap 扫描所有 IPv4 地址的 853 端口（使用 `zmap -p 853` 命令），然后使用 `getdns` API 用 DoT 查询探查这些地址。使用 OpenSSL 来验证证书的有效性。2019 年 2 月 1 日至 2019 年 5 月 1 日，每隔 10 天扫描一次，测量结果如图表 10 所示。

测量到 1500 个开放的 DoT 解析器，如图表 10 所示，主要由大型服务提供商拥有，但也有一些小型提供商，它们在公共解析器列表中鲜为人知。每次扫描中发现远超 1.5K 个主机的 853 端口处于打开状态（例如 2 月 1 日为 356M，5 月 1 日为 230M），但是其中绝大多数不提供 DoT。如图表 10，在每次扫描中发现了超过 1.5K 个开放式 DoT 解析器，远远超过了公共解析器列表。从地理位置上看，解析度最高的前 10 个国家中，爱尔兰，巴西和俄罗斯的 DoT 解析器在三个月内增长了一倍，而美国的 DoT 解析器增长了四倍。相比之下，发现中国的 DoT 解析器数量大幅下降（-84%），关闭的解析器大多属于云托管平台。

25%的服务提供商存在使用无效 SSL 证书的 DoT 解析器的情况，如图图表 11。在最新的扫描结果（2019 年 5 月 1 日）中，有 62 个提供商的 122 个解析器使用了无效证书，包括 27 个过期的证书，67 个自签名的证书和 28 个无效证书链。在 27 张已过期的证书中，有 9 张已在 2018 年过期，表明它们可能不再维护。作者还发现，有 47 个解析器使用 FortiGate（Fortinet 的防火墙）的自签名默认证书，充当 DoT 代理，将检查来自 DNS 客户端的加密查询。



图表 10 每次扫描识别到的 DoT 解析器数量



图表 11 公开 DoT 解析器的服务提供商证书有效性

**发现公开的 DoH 解析服务器：**利用合作伙伴提供的 URL 数据集（数十亿个 URL），通过已知的 DoH 的 URL 模板（例如，/dns-query 和/resolve 等）进行匹配。

相比开放的 DoT 解析器，测量到的 DoH 解析器的数量很少。在 URL 数据集中匹配到了 61 个有效的 URL，这些 URL 具有常见的 DoH 路径（例如/dns-query 和/resolve）。对于每个 URL，通过手动进行 DoH 查询来检查其可用性。最终共发现了 17 个可用的 DoH 解析器，远少于测量发现的 DoT 解析器的数量。对于这 17 个 DoH 解析器检查其证书的有效性发现，它们的 443 端口上的证书都是有效的。

注：根据 curl 项目给出的最新结果，公开的 DoH 解析器已经不止有 17 个，截至

2020 年 6 月 14 日，共有 71 个，符合/dns-query 模板的有 61 个，符合/resolve 模板已经不存在了。

### 4.2.2 客户端测量

利用 ProxyRack 在全球 150 多个国家和地区建立 600,000 个 SOCKS 代理测绘点。在获得全球视野的同时，作者还通过芝麻代理的 SOCKS 代理（位于中国）探查网络审查地区的 DoE 流量拦截情况。考虑到测试效率，作者将测试范围缩小到三个大型且有代表性的公众解析器：Cloudflare，Google 和 Quad9。为了进行比较，研究团队也自建了一个解析器，它支持明文 DNS，DoT 和 DoH。测量结果如下。

#### 1) 可达性分析

测试结果如图表 12 所示，超过 99% 的全球用户可以正常访问大型的 DoE 服务器，而不到 1% 的客户端遇到由 IP 冲突，网络审查和 TLS 拦截问题。

Platform	Type	Cloudflare			Google			Quad9			Self-built		
		Correct	Incorrect	Failed	Correct	Incorrect	Failed	Correct	Incorrect	Failed	Correct	Incorrect	Failed
ProxyRack (Global)	DNS	83.46%	0.08%	16.46%	84.12%	0.08%	15.80%	99.78%	0.11%	0.11%	99.90%	0.06%	0.04%
	DoT	98.84%	0.02%	1.14%	n/a <sup>2</sup>	n/a	n/a	99.78%	0.06%	0.15%	99.90%	0.05%	0.05%
	DoH	99.91%	0.04%	0.05%	99.85%	0.00%	0.15%	85.99%	13.09%	0.92%	99.93%	0.02%	0.05%
Zhima (Censored, China)	DNS	84.86%	0.00%	15.14%	98.91%	0.01%	1.08%	99.76%	0.01%	0.23%	99.90%	0.05%	0.05%
	DoT	84.90%	0.00%	15.10%	n/a	n/a	n/a	99.47%	0.02%	0.51%	99.81%	0.02%	0.18%
	DoH	99.74%	0.00%	0.25%	0.01%	0.00%	99.99%	99.25%	0.15%	0.60%	99.92%	0.00%	0.08%

<sup>1</sup> Failed: clients receive no DNS response packets. Incorrect: we only see SERVFAIL responses and responses with 0 answers.

<sup>2</sup> At the time of experiment, Google DoT was not announced.

图表 12 可达性测试结果

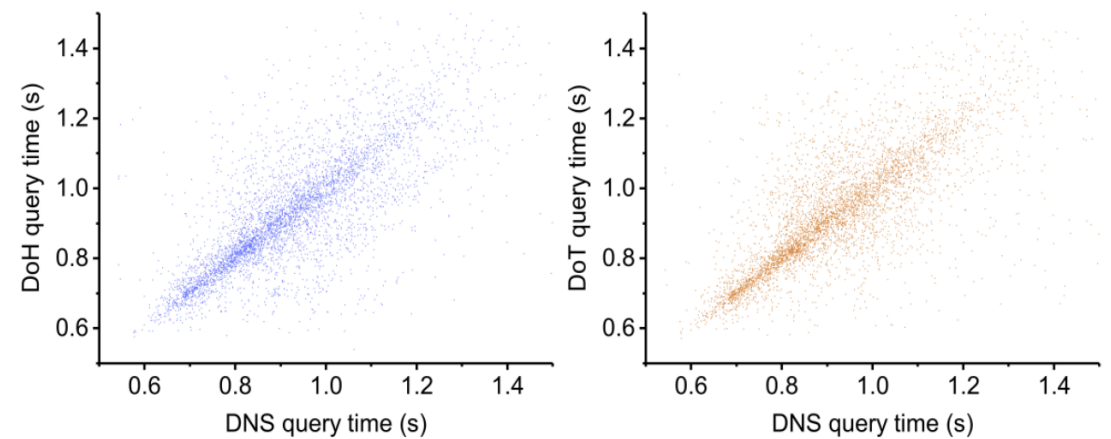
由于网络审查制度的存在，Google DoH 无法被中国的用户所使用。在整个中国的数据集中，有 99.99% 的客户端无法使用 Google 的 DoH 服务。此外，发现 Quad9 DoH 存在一个配置问题，导致查询失败率有 13%。

#### 2) 性能分析

使用复用连接时，对 DNS 查询进行加密会给全局客户端的查询等待时间带来可承受的开销，并且可以像明文 DNS 一样良好。平均而言，具有复用连接的 DoE 查询延迟比传统 DNS 查询多几毫秒。通过比较 Cloudflare 的明文 DNS，DoT 和 DoH 的查询延迟，获得了 5ms / 9ms（对于 DoT）和 8ms / 6ms（对于 DoH）的“平均数/中位数”的性能开销。如果我们查看各个客户端，图表 13 显示了其明文 DNS 和 DNS 加密之上的查询性能。大多数客户端分布在 y = x 线附近，这

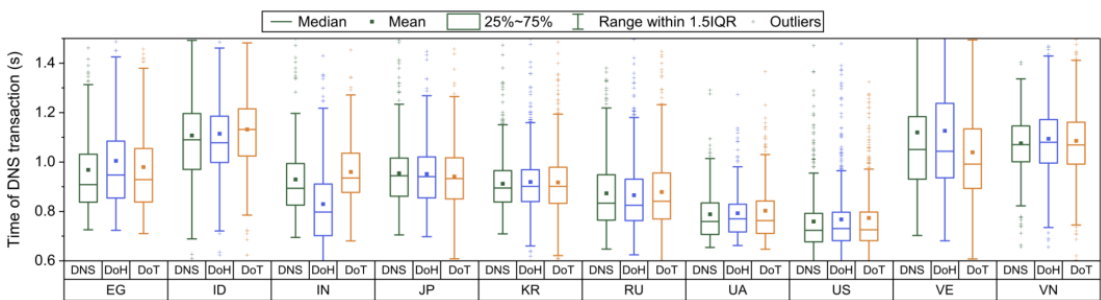


表明他们并没有遭受明显的性能降级。相反，如果每个查询建立一个完整的新 TCP 和 TLS 会话（即不使用复用连接），则性能开销可能会很大，尤其是在解析器离客户端较远的情况下，无复用连接的性能开销可能高达数百毫秒。



图表 13 在单个代理客户端上 DNS 和 DoH（左）、DoT（右）的查询时间

此外，DoE 的性能在不同国家和地区间也会有波动。如图表 14 所示，尽管全球来看性能开销很小，但也能发现 DoE 延迟高于平均水平的国家/地区。例如，在使用 Cloudflare 的 DoT 时，在印度尼西亚的 504 个客户端的开销的平均数/中位数分别为 25ms / 42ms。相比之下，对于某些客户端，DoE 甚至可以比明文 DNS 更快。例如，与明文 DNS 相比，使用 Cloudflare DoH 时，在印度的 282 个客户端的平均数/中位数分别降低了 99ms / 96ms。其他测试也表明了使用 DoE 可能实现的性能改进。这很可能是由于查询采用的任播或路由不同导致的，并且不同地域的解析器对域名解析服务器的延迟不同。



图表 14 在不同每个国家和地区的性能表现

4.2.3 真实世界中 DoE 流量测量

**测量 DoT:** 使用了由位于中国的大型 ISP 骨干路由器收集的 18 个月的

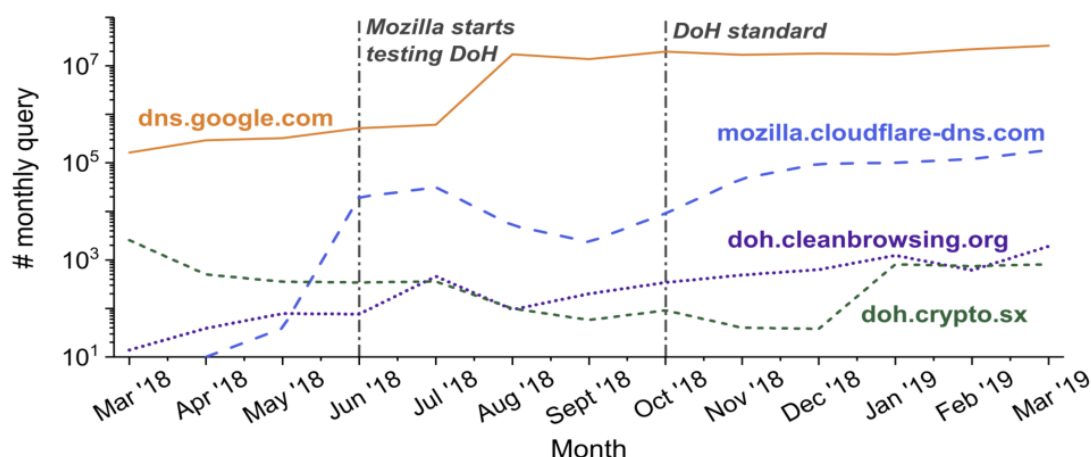


NetFlow 数据集（2017 年 7 月至 2019 年 1 月）来观测 DoT 流量。如果一个流是由客户端发送到 DoT 解析器的 TCP 端口 853，则将其视为 DoT 流量。

**测量 DoH:** DoH 查询与 HTTPS 流量混合在一起，因此从 NetFlow 等流量数据集中观察它们是不可行的。因此通过检查被动 DNS 数据集中解析器域名的查询量来评估 DoH 使用情况。DNSDB 和 360 PassiveDNS 是分别由 Farsight Security 和 Qihoo 360 维护的两个大型被动 DNS 数据集，它们都包含给定域的聚合统计信息，包括其第一个查询和最后一个查询的时间戳以及历史查询数。

测量结果如下：

- 1) 尽管 DoT 与传统 DNS 相比规模仍然很小，但可以观察到 DoT 服务的实际流量在近几个月来使用量的增长。
- 2) 大型服务提供商在所有 DoH 服务中占主导地位，并且其使用量还在增长。根据 DNSDB，在发现的 17 个公共 DoH 解析程序中只有 4 个域的月查询数超过 1 万。由于其他解析器的流量并不大，因此我们重点关注四个流行的 DoH 解析器（即 Cloudflare，Google，CleanBrowsing 和 crypto.sx）的查询趋势。根据 360 PassiveDNS 数据集，图表 15 显示了 4 个流行的 DoH 域的每月查询量。作为最早出现且最受欢迎的 DoH 解析程序，Google DoH 收到的查询量比其他域名多几个数量级。由于 Firefox 对 DoH 的支持以及在 Firefox Nightly 上进行的 DoH 测试，Cloudflare 的 DoH 也收到了大量流量。除此之外，DoH 解析程序的查询量都在增长，例如，从 2018 年 9 月（记录了 200 条查询）到 2019 年 3 月（记录了 1,915 条查询），CleanBrowsing DoH 的查询量增加了近 10 倍。



图表 15 热门 DoH 域的月查询量变化趋势

### 4.3 《Encrypted DNS => Privacy? A Traffic Analysis Perspective》

#### 4.3.1 加密 DNS 指纹识别效果

在一个封闭的网站列表内，记录访问每个网站所产生的 DoH 流量。根据产生的 DoH 流量学习得到一个模型。然后，再次访问某个网站，记录产生的 DoH 流量。根据这些 DoH 流量和学习到的模型推测出该网站是何网站。这样的过程被称为加密 DNS 指纹识别。如图表 16 所示，文章在一个 1500 个网站的封闭列表中，访问每个网站所产生的 DoH 数据取样 200 次，然后利用随机森林算法学习得到一个模型。该模型在数据集中的准确率、召回率、F 值基本高于 90%。

Scenario	Precision	Recall	F1-score
Curated traces	0.914 ± 0.002	0.909 ± 0.002	0.908 ± 0.002
Full dataset	0.904 ± 0.003	0.899 ± 0.003	0.898 ± 0.003
Combined labels	0.940 ± 0.003	0.935 ± 0.003	0.934 ± 0.003

图表 16 加密 DNS 指纹识别效果

文章对比了传统 web 指纹识别方法在加密 DNS 指纹识别中表现和本文方法在 web 网页指纹识别中的效果。如图表 17 所示，本文提出的均优于传统 web 指纹识别方法。

	DoH-only	Web-only	DoH + Web
n-grams	0.87	0.99	0.88
k-Fingerprinting [18]	0.74	0.95	0.79
CUMUL [16]	0.75	0.92	0.77
DF [19] <sup>6</sup>	0.51	0.94	0.75

图表 17

文章还考察了训练样本数对加密 DNS 指纹识别的影响。如图表 18，当每个网站训练样本数为 10 时，准确率达到 0.873。当样本数增加到 40 时，准确率达到 0.908，增长了 0.035，准确率随着样本数的增多增长较为明显。当样本数达到 100 时，准确率为 0.912，增长了 0.004，准确率随着样本数的增多增长较为很少，可见训练样本数为 40 时大概已经到达合适的训练样本数。

Number of samples	Precision	Recall	F1-score
10	0.873	0.866	0.887
20	0.897	0.904	0.901
40	0.908	0.914	0.909
100	0.912	0.916	0.913

图表 18

文章接着考察了训练数据与测试数据的间隔时间对加密 DNS 指纹识别率的影响。如图表 19，列表示训练样本数据采集时间，行表示测试样本数据采集时间。我们可以看到，用本周训练样本得到的模型来测试本周的数据的识别率较高，分别为 0.880、0.921、0.910、0.876、0.906。时间相隔越远，识别效果越差，用第 0 周训练数据去测试 2、4、6、8 周收集的测试数据，识别率分别为 0.886、0.868、0.775、0.770。识别率逐渐降低。

F1-score	0 weeks old	2 weeks old	4 weeks old	6 weeks old	8 weeks old
0 weeks old	0.880	0.827	0.816	0.795	0.745
2 weeks old	0.886	0.921	0.903	0.869	0.805
4 weeks old	0.868	0.898	0.910	0.882	0.817
6 weeks old	0.775	0.796	0.815	0.876	0.844
8 weeks old	0.770	0.784	0.801	0.893	0.906

图表 19

文章进一步考察了收集数据的基础设计对于 DNS 指纹识别的影响。如图表 20 左 1 图，用同一位置收集到的训练数据测试收集的测试数据效果最好。随着位置相隔距离边远，识别效果逐渐下降。如图表 20 左 2 图，使用 Google 和 Cloudflare 的 DoH 服务器收集的数据相互测试，发现 Google DoH 服务器收集到训练数据能较好识别 Google 和 Cloudflare DoH 服务器收集的测试数据。但 Cloudflare DoH 服务器收集到训练数据只能较好识别 Cloudflare DoH 服务器收集的测试数据，但不能识别 Google DoH 服务器收集到的测试数据。如图表 20 右 2，发现桌面平台和 RPI 平台收集到的数据无法相互识别。同理，如图表 20 右 1，不同 DoH 客户端收集到的数据收集的数据也无法相互识别。

Location	LOC1	LOC2	LOC3	Resolver	GOOGLE	CLOUD	Platform	DESKTOP	RPI	Client	CLOUD	CL-FF	LOC2
LOC1	0.906	0.712	0.663	GOOGLE	0.880	0.129	DESKTOP	0.8802	0.0003	CLOUD	0.885	0.349	0.000
LOC2	0.748	0.908	0.646	CLOUD	0.862	0.885	RPI	0.0002	0.8940	CL-FF	0.109	0.892	0.069
LOC3	0.680	0.626	0.917							LOC2	0.001	0.062	0.908

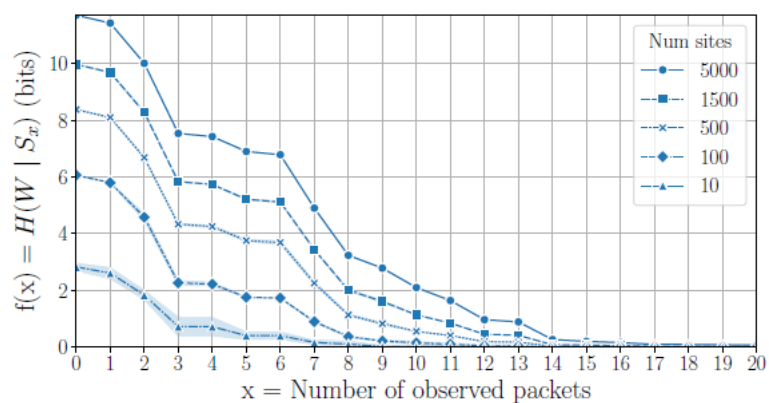
图表 20

### 4.3.2 加密 DNS 识别与审查

文章定义一个评价访问网站产生的 DoH 流量序列的唯一性函数：

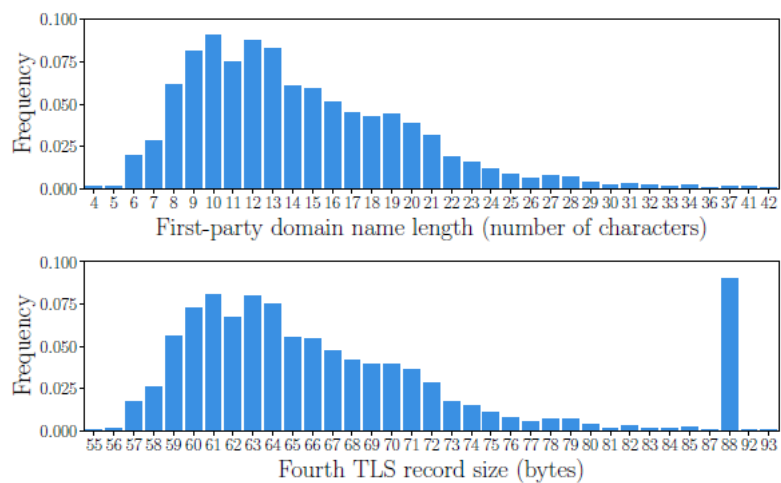
$$H(W | S_l) = \sum_{\forall o \in \Omega_{S_l}} \Pr[S_l = o] H(W | S_l = o).$$

该函数的自变量是 DoH 流量序列长度，如图表 21，随着访问网站参数的 DoH 数据包的增多，函数值逐渐减少，DoH 序列的唯一性逐渐增加。当产生的 DoH 数据包数量达到 15 个后，函数值低于 1，拥有可能唯一辨识的可能。



图表 21

同时，文章发现访问网站产生 DoH 序列的第 4 个数据包的大小与网站对应域名的长度成正相关，如图表 22。这说明 DoH 序列的第 4 个数据包是网站域名的 DNS 解析请求。



图表 22

## 5 常用工具

### 5.1 常用的客户端部署工具

参考链接: <https://dnscrypt.info/implementations/>

工具名称	作者	支持的协议	支持的系统	工具的 实现语 言
DNSCrypt-Proxy	Frank Denis (@jedisct1)	DNSCrypt , DoH and Anonymized DNSCrypt	Linux , BSD , Windows , macOS , Android and more	Golang
SecureDNS	Texnomic (@Texnomic)	DNSCrypt , DoH , DoT , and Anonymized DNSCrypt	Linux , Windows , macOS and more	C#
DoH-proxy	Facebook	DoH	Linux , BSD , Windows , macOS and more	Python
Pcap_DNSProxy		DNSCrypt	Windows , Linux , macOS and OpenWrt/LEDE	C++
YourFriendlyDNS		DNSCrypt	Linux , Windows , macOS and Android	C++
Simple DNSCrypt	Christian Hermann	DNSCrypt and DoH	Windows	C#

dnscrypt-proxy switcher	Frank Denis	DNSEncrypt	macOS	Shell
DNSCloak	Sergey @s-s	DNSEncrypt	iOS	
DNSEncrypt proxy on Android		DNSEncrypt	Android	
DNSLookup	Andrey Meshkov	DNSEncrypt , DoH and DoT	Linux , BSD , Windows , macOS and more	Go
DNSProxy	Adguard team	DNSEncrypt , DoH and DoT	Linux , BSD , Windows , macOS and more	Go
YogaDNS	Initex	DNSEncrypt and DoH	Windows	

### 5.1.1 DNSEncrypt-proxy

- 项目链接: <https://github.com/DNSEncrypt/dnscrypt-proxy>
- 功能: 用作加密 DNS 客户端, 可以作为系统 DNS 使用, 监听系统 DNS 请求, 并转换为加密 DNS 请求转发到加密 DNS 服务器。
- 支持 DNSEncrypt 和 DoH 两种协议
- 维护了一个[加密 DNS 服务器列表](#)。列表中的每个服务器均提供一个 sdns 字符串, 里面包含了服务器类别(DoH/DNSEncrypt)以及对服务器发起请求所需的参数。

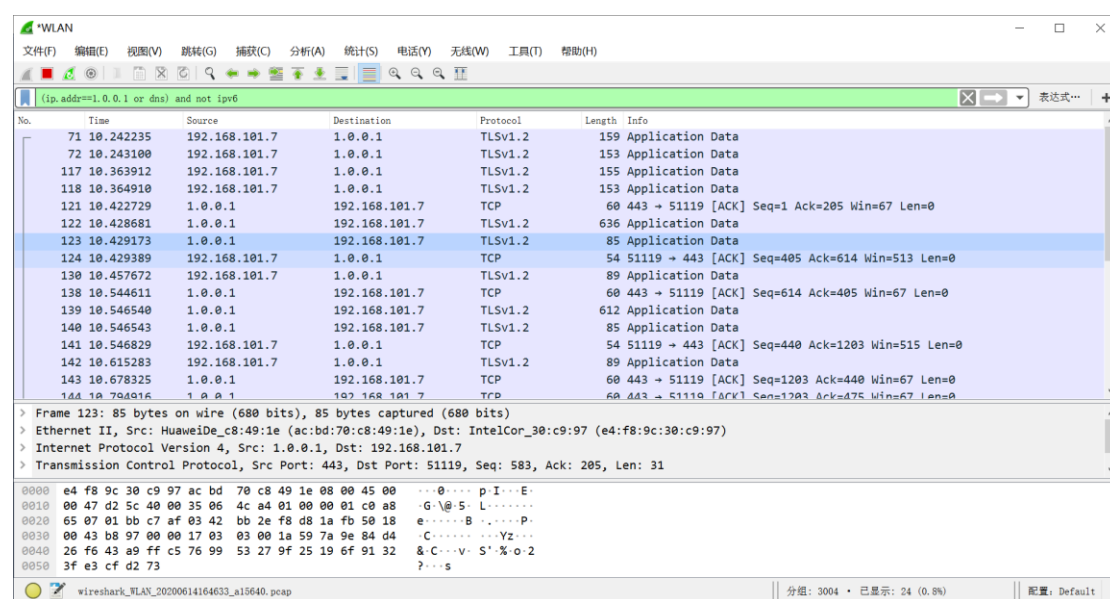
#### 5.1.1.1 使用方法:

- 1) 修改配置文件(dnscrypt-proxy.toml), 将 DNSEncrypt-proxy 监听端口设为 127.0.0.1:53。
- 2) 修改配置文件, 设置使用的加密 DNS 服务器
- 3) 设置系统 DNS 地址为 127.0.0.1

#### 4) 运行 DNSCrypt-proxy

### 5.1.1.2 测试:

- 系统: win10
  - 工具: dnscrypt-proxy-win64-2.0.42、wireshark
- 1) 参照 5.1.1.1 节方法开启加密 DNS 服务器。这是使用的是 cloudflare(1.0.0.1) 做加密 DNS 服务器, 该服务器为 DoH 服务器。
  - 2) 在浏览器中浏览网页
  - 3) Wireshark 捕获 DNS 数据包和 IP 地址为 1.0.0.1 的数据包
  - 4) 测试结果: 如图表 23。没有任何明文 DNS 请求, 全部是发往 1.0.0.1 的加密 DNS 请求。证明 DNS 已经被成功加密。



The image shows a Wireshark network capture window. The top toolbar includes icons for file operations, editing, viewing, jumping, capturing, analyzing, statistics, network interfaces, and help. The filter bar at the top shows the filter '(ip.addr==1.0.0.1 or dns) and not ip6'. The packet list on the left shows a series of packets, with packet 123 selected. The packet details pane on the right shows the structure of the selected packet: Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane at the bottom shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
71	10.242235	192.168.101.7	1.0.0.1	TLSv1.2	159	Application Data
72	10.243100	192.168.101.7	1.0.0.1	TLSv1.2	153	Application Data
117	10.363912	192.168.101.7	1.0.0.1	TLSv1.2	155	Application Data
118	10.364910	192.168.101.7	1.0.0.1	TLSv1.2	153	Application Data
121	10.422729	1.0.0.1	192.168.101.7	TCP	60	443 → 51119 [ACK] Seq=1 Ack=205 Win=67 Len=0
122	10.428681	1.0.0.1	192.168.101.7	TLSv1.2	636	Application Data
123	10.429173	1.0.0.1	192.168.101.7	TLSv1.2	85	Application Data
124	10.429389	192.168.101.7	1.0.0.1	TCP	54	51119 → 443 [ACK] Seq=405 Ack=614 Win=513 Len=0
130	10.457672	192.168.101.7	1.0.0.1	TLSv1.2	89	Application Data
138	10.544611	1.0.0.1	192.168.101.7	TCP	60	443 → 51119 [ACK] Seq=614 Ack=405 Win=67 Len=0
139	10.546540	1.0.0.1	192.168.101.7	TLSv1.2	612	Application Data
140	10.546543	1.0.0.1	192.168.101.7	TLSv1.2	85	Application Data
141	10.546829	192.168.101.7	1.0.0.1	TCP	54	51119 → 443 [ACK] Seq=440 Ack=1203 Win=515 Len=0
142	10.615283	192.168.101.7	1.0.0.1	TLSv1.2	89	Application Data
143	10.678325	1.0.0.1	192.168.101.7	TCP	60	443 → 51119 [ACK] Seq=1203 Ack=440 Win=67 Len=0
144	10.704016	1.0.0.1	192.168.101.7	TCP	60	443 → 51119 [ACK] Seq=1203 Ack=475 Win=67 Len=0

> Frame 123: 85 bytes on wire (680 bits), 85 bytes captured (680 bits)  
> Ethernet II, Src: HuaweiDe\_c8:49:1e (ac:bd:70:c8:49:1e), Dst: IntelCor\_30:c9:97 (e4:f8:9c:30:c9:97)  
> Internet Protocol Version 4, Src: 1.0.0.1, Dst: 192.168.101.7  
> Transmission Control Protocol, Src Port: 443, Dst Port: 51119, Seq: 583, Ack: 205, Len: 31

0000 e4 f8 9c 30 c9 97 ac bd 70 c8 49 1e 08 00 45 00 ...0... p-I...E:  
0010 00 47 d2 5c 40 00 35 06 4c a4 01 00 00 01 c0 a8 -G...@5-L...:  
0020 65 07 01 bb c7 af 03 42 bb 2e f8 d8 1a fb 50 18 e.....B.....P:  
0030 00 43 08 97 00 00 17 03 03 00 1a 59 7a 9e 84 d4 .C.....Vz...:  
0040 26 f6 43 a9 ff c5 76 99 53 27 9f 25 19 6f 91 32 &C...v..S'X-o-2  
0050 3f e3 cf d2 73 ?...s

图表 23

### 5.1.2 SecureDNS

- 项目链接: <https://github.com/Texnomic/SecureDNS>
- 功能: 用作加密 DNS 客户端, 可以作为系统 DNS 使用, 监听系统 DNS 请求, 并转换为加密 DNS 请求转发到加密 DNS 服务器。
- 支持 DNSCrypt、DoH、DoT 等多种协议。



### 5.1.2.1 使用方法

- 1) 运行 Texnomic.SecureDNS.Terminal.exe。生成一个名为 AppSettings.json 的配置文件。
- 2) 修改配置文件设置加密 DNS 服务器及加密 DNS 协议。
- 3) 修改配置文件，设置监听端口为 127.0.0.1:53。
- 4) 设置系统 DNS 地址为 127.0.0.1
- 5) 关闭 Texnomic.SecureDNS.Terminal.exe 后再次运行。

### 5.1.2.2 测试方法

- 系统：win10
  - 工具：SecureDNS、wireshark
- 1) 修改配置文件设置 DoH 服务器(<https://1.0.0.1/dns-query>)，设置加密 DNS 协议为 HTTPS(DoH)。Note：这里设置为我们的 DoH 服务器则无法访问网页。
  - 2) 在浏览器中浏览网页
  - 3) Wireshark 捕获 DNS 数据包和 IP 地址为 1.0.0.1 的数据包
  - 4) 测试结果：如图表 24。仅有少量明文 DNS 请求，应该是解析加密 DNS 服务器参数的请求，大部分是发往 1.0.0.1 的加密 DNS 请求。证明 DNS 加密成功。

No.	Time	Source	Destination	Protocol	Length	Info
813	53.278218	192.168.101.7	1.0.0.1	TCP	54	52716 → 443 [ACK] Seq=702 Ack=713 Win=131072 Len=0
814	53.387294	192.168.101.7	1.0.0.1	TCP	66	[TCP Retransmission] 52715 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1...
815	53.987208	192.168.101.7	1.0.0.1	TCP	66	[TCP Retransmission] 52717 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1...
817	54.006797	192.168.101.7	8.8.8.8	DNS	70	Standard query 0x3899 A google.com
818	54.026878	8.8.8.8	192.168.101.7	DNS	96	Standard query response 0x3899 A google.com A 59.24.3.174
819	54.046720	8.8.8.8	192.168.101.7	DNS	86	Standard query response 0x3899 A google.com A 172.217.27.142
820	54.086497	1.0.0.1	192.168.101.7	TCP	66	443 → 52717 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1200 SACK_P...
821	54.087032	192.168.101.7	1.0.0.1	TCP	54	52717 → 443 [ACK] Seq=1 Ack=1 Win=131840 Len=0
822	54.088206	192.168.101.7	1.0.0.1	TLSv1.2	393	Client Hello
824	54.267219	1.0.0.1	192.168.101.7	TCP	60	443 → 52717 [ACK] Seq=1 Ack=340 Win=67584 Len=0
825	54.267965	1.0.0.1	192.168.101.7	TLSv1.2	169	Server Hello, Change Cipher Spec, Encrypted Handshake Message
826	54.271421	192.168.101.7	1.0.0.1	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
827	54.272610	192.168.101.7	1.0.0.1	TLSv1.2	364	Application Data
829	54.450814	1.0.0.1	192.168.101.7	TCP	60	443 → 52717 [ACK] Seq=116 Ack=391 Win=67584 Len=0
830	54.451196	1.0.0.1	192.168.101.7	TCP	60	443 → 52717 [ACK] Seq=116 Ack=701 Win=68608 Len=0
831	54.456193	1.0.0.1	192.168.101.7	TLSv1.2	650	Application Data

图表 24

## 5.1.3 DNSLookup

- 项目链接: <https://github.com/ameshkov/dnslookup>
- 功能: 构建加密 DNS 请求, 发起一次性加密 DNS 请求, 无法监听系统 DNS。
- 支持 DoH、DoT、DNSCrypt 协议

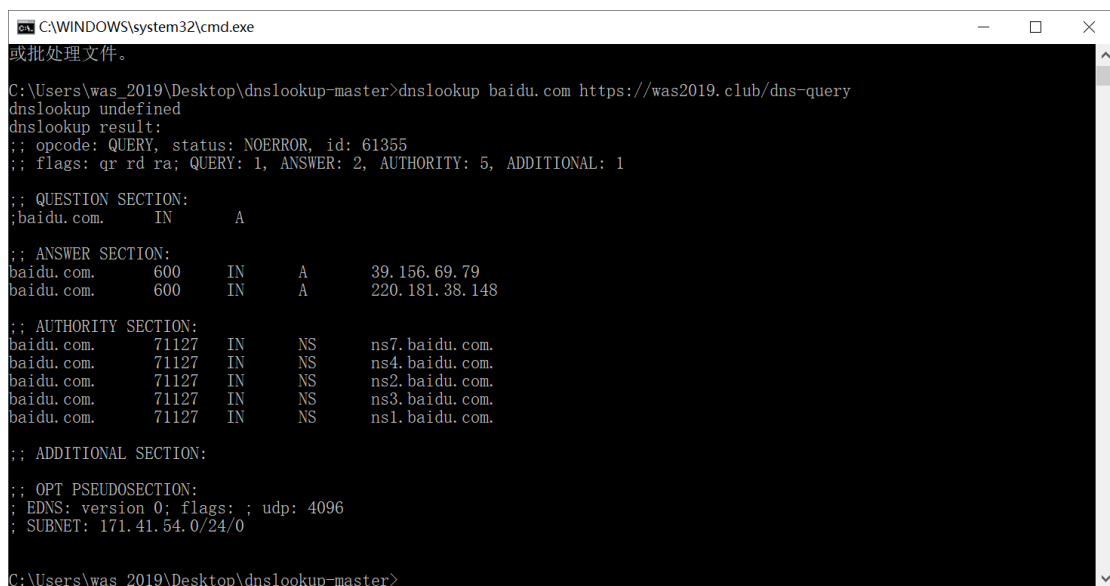
### 5.1.3.1 使用方法

- 构造明文请求: `./dnslookup example.org 176.103.130.130`
- 构造 DoT 请求: `./dnslookup example.org tls://dns.adguard.com`
- 构造 DoT 请求: `./dnslookup example.org tls://dns.adguard.com 176.103.130.130`
- 构造 DoH 请求: `./dnslookup example.org https://dns.adguard.com/dns-query`
- 构造 DoH 请求: `./dnslookup example.org https://dns.adguard.com/dns-query 176.103.130.130`
- 构造 DNSCrypt 请求: `./dnslookup example.org sdns://AQIAAAAAAAAAAFD E3Ni4xMDMuMTMwLjEzMDo1NDQzINER JS3PLCu_iZEIbq95zkSV2LFsig xDIuUso_OQhzIjluZG5zY3J5cHQzZGVmYXVsdC5uczEuYWRndWFyZC5jb2 0`

### 5.1.3.2 测试方法

- 系统: win10
- 工具: DNSlookup 源码、wireshark

- 1) ./dnslookup baidu.com <https://was2019.club/dns-query> 49.233.140.93
- 2) 结果: 如下图。成功收到域名解析结果。抓取的数据包中全是加密的数据包。  
证明成功构造了 DoH 请求, 并收到响应。



```
C:\WINDOWS\system32\cmd.exe
或批处理文件。

C:\Users\was_2019\Desktop\dnslookup-master>dnslookup baidu.com https://was2019.club/dns-query
dnslookup undefined
dnslookup result:
;; opcode: QUERY, status: NOERROR, id: 61355
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 1

;; QUESTION SECTION:
;baidu.com.      IN      A

;; ANSWER SECTION:
baidu.com.      600     IN      A       39.156.69.79
baidu.com.      600     IN      A       220.181.38.148

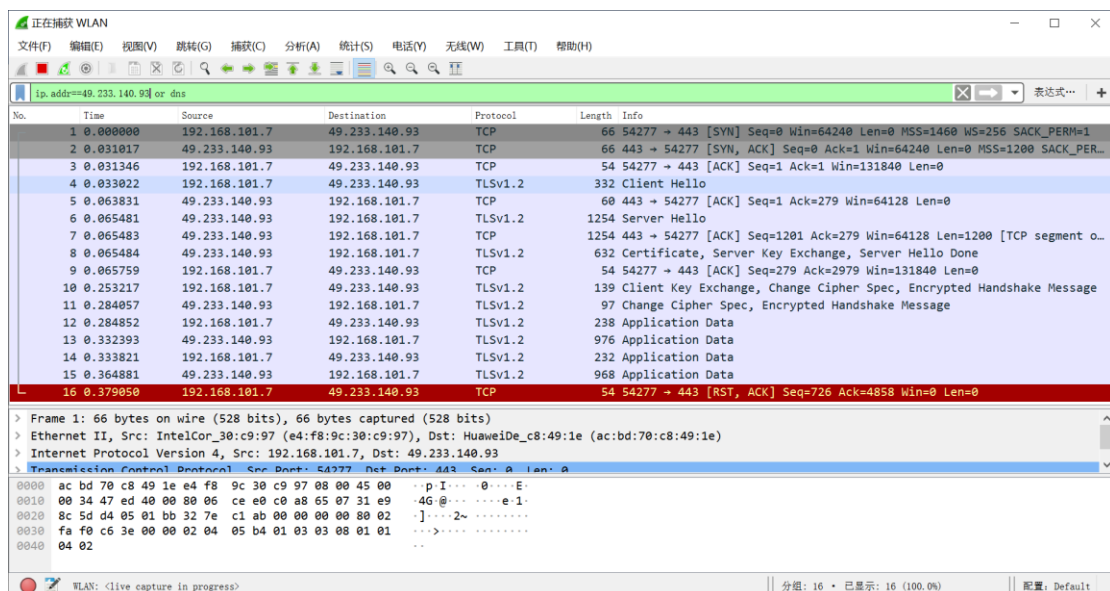
;; AUTHORITY SECTION:
baidu.com.      71127   IN      NS      ns7.baidu.com.
baidu.com.      71127   IN      NS      ns4.baidu.com.
baidu.com.      71127   IN      NS      ns2.baidu.com.
baidu.com.      71127   IN      NS      ns3.baidu.com.
baidu.com.      71127   IN      NS      ns1.baidu.com.

;; ADDITIONAL SECTION:

;; OPT PSEUDOSECTION:
; EDNS: version 0; flags: ; udp: 4096
; SUBNET: 171.41.54.0/24/0

C:\Users\was_2019\Desktop\dnslookup-master>
```

图表 25



图表 26

## 5.1.4 DNSProxy

- 项目链接: <https://github.com/AdguardTeam/dnsproxy>
- 功能: 用作加密 DNS 客户端或加密 DNS 服务器。当做客户端时可以作为系统 DNS 使用, 监听系统 DNS 请求, 并转换为加密 DNS 请求转发到加密 DNS 服务器。
- 支持 DoH、DoT、DNSEncrypt 等多种加密 DNS 协议。

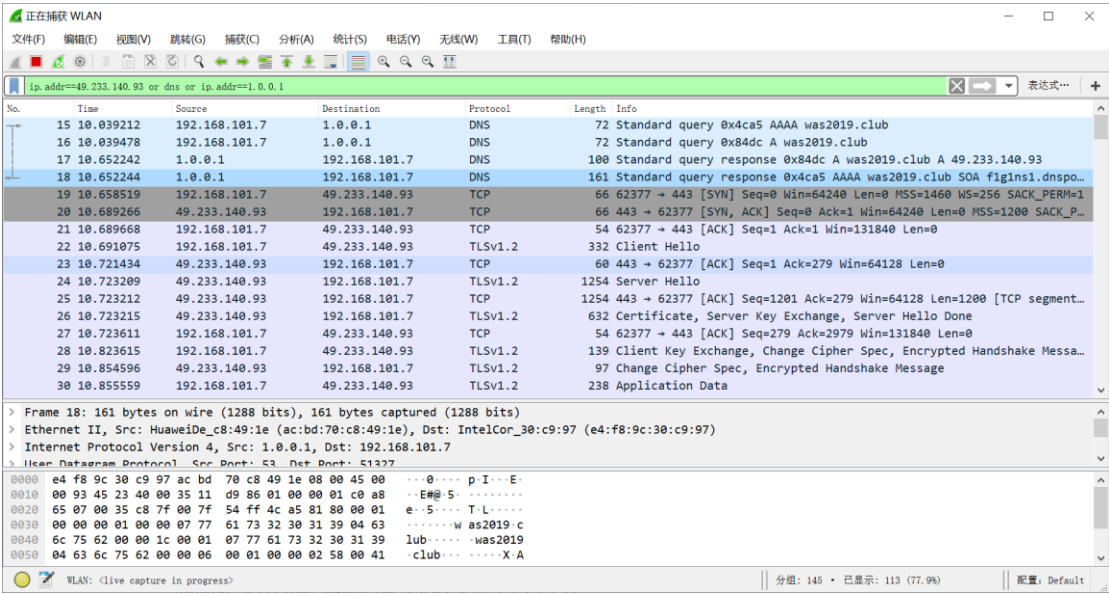
### 5.1.4.1 使用方法

- 设置系统 DNS 地址为 127.0.0.1
- 用做明文系统 DNS 客户端: `./dnsproxy -u 8.8.8.8:53`
- 用作系统 DoH 客户端: `./dnsproxy -u https://dns.adguard.com/dns-query -b 1.1.1.1:53`
- 用作系统 DoT 客户端: `./dnsproxy -u tls://dns.adguard.com`
- 用作系统 DNSEncrypt 客户端: `./dnsproxy -u sdns://AQIAAAAAAAAAAFDE3Ni4xMDMuMTMwLjEzMDo1NDQzINErR_JS3PLCu_iZEIbq95zkSV2LFsigxDIuUso_OQhzIjIuZG5zY3J5cHQvZGVmYXVs dC5uczEuYWVmdWVhZC5jb20`
- 用作系统 DoH 客户端 (DNS stamp): `./dnsproxy -u sdns://AgcAAAAAAAAABzEuMC4wLjGgENk8mGSIIIfMGXMOIIICcKvq7AVgrZxtjon911-ep0cg63U1-I8NIFj4GplQGb_TTLiczclX57DvMV8Q-JdjgRgSZG5zLmNsb3VkZmxhcUuY29tCi9kbmMtcXVlcnk`

### 5.1.4.2 测试方法

- 系统: win10
  - 工具: DNSProxy 源码、wireshark
- 1) 使用 5.1.4.2 小节方法开启加密 DNS 客户端, 这里选择使用 DoH 协议: `./dnsproxy -u https://was2019.club/dns-query -b 1.0.0.1:53`
  - 2) 使用浏览器随意访问页面, 并捕包分析

3) 结果：捕获的数据包中，仅在查询 DoH 服务器时使用明文 DNS，后面的查询均为加密 DNS。证明成功开启 DoH 客户端，保证了 DNS 加密性。



图表 27

5.2 DoH 相关工具

参考链接：<https://github.com/curl/curl/wiki/DNS-over-HTTPS>

Name	Author/Organization	Comments
coredns	Cloudflare	CoreDNS is a DNS server/forwarder, written in Go from the Cloud Native Computing Foundation.
doh-proxy	Facebook	tools for DoH
dns2doh	Daniel	tool for generating DOH responses and questions.
doh-proxy	Frank Denis	server-side proxy in rust
doh-php-client	Daniel Cid	can be used to test and run DoH requests via PHP applications.
doh-js-	Peter Lai	client-side implementation of DoH, can be

client		used in nodejs backend.
jDnsProxy	Travis Burtrum	DNS proxy and cache, implementing DNS-over-TLS, DNS-over-HTTPS, and Serve-Stale
dns-over-https	Star Brilliant	server-side and client-side implementation, written in Golang
dnsdist	PowerDNS	supports doh, see <a href="https://dnsdist.org/guides/dns-over-https.html">https://dnsdist.org/guides/dns-over-https.html</a>
dnss	Alberto Bertogli	daemon written in Go which acts as a proxy (the most common use case), and as a server (in case you want end-to-end control).
nss-tls	Dima Krasner	a daemon that makes gethostbyname(), getaddrinfo(), etc. happen through DoH, without any change to applications, thus transparently migrating all applications that don't use their own resolver (like some browsers) from DNS to DoH.
dealdoh	Maxime Elomari	a middleware to proxy DoH requests to different DNS upstreams, written in PHP.
Encrypted-DNS	Siujoeng Lau	DNS-over-HTTPS forwarder written in Python
RouteDNS	Frank Olbricht	a flexible stub resolver, proxy, and router with support for DoH, DoT, and plain DNS written in Go.
h2odoh	Max Kostikov	an implementation with H2O HTTP/2 server using embedded mruby.
Encrypted DNS Server	Frank Denis	can serve DNSCrypt and DoH traffic simultaneously, written in Rust.
dnscrypt-	Frank Denis	dnscrypt-proxy 2 - A flexible DNS proxy,

proxy		with support for encrypted DNS protocols.
quart-doh	Matthieu Treussart	HTTP/2 server who serves a DOH proxy written in Python, with Quart Python web microframework.
EasyDoH	ElevenPaths	a simple add-on for Firefox that allows one to easily activate DNS over HTTPS and its working mode with just one click.
dohjs	BYU IMAAL	Client DoH JavaScript library for accessing DNS information from web applications. Can be tested at dohjs.org
Technitium DNS Server	Technitium	A FOSS, cross-platform DNS Server written in C# that can consume as well as host DNS-over-HTTPS (DoH) and DNS-over-TLS (DoT) services.

## 6 公共加密 DNS 服务器

### 6.1 DNSCrypt-proxy 提供的加密 DNS 服务器列表

- 链接: <https://dnscrypt.info/public-servers/>
- 特点: 包含 DoH、DNSCrypt 两种协议。服务器均使用 DNS stamp 加密访问服务器所需要的参数。
- 共计 230 个服务器
- 提供每个服务器的名称、IP、域名、stamp、描述、所支持的协议、是否记录解析日志、是否支持 DNSSEC 等信息。



Name ↑	Description	Protocol	Logging	DNSSEC
a-and-a	Non-logging DoH server in the UK operated by Andrews & Arnold Ltd, a company providing Internet connectivity and VoIP in the UK. <a href="https://www.aa.net.uk/legal/dohdot-disclaimer/">https://www.aa.net.uk/legal/dohdot-disclaimer/</a>	DoH		🔒
aafalo-me	DNS-over-HTTPS server running dns-over-https with PiHole for Adblocking in NL. Non-logging, AD-filtering, supports DNSSEC. Hosted in Netherlands on a RamNode VPS. 🚫	DoH		🔒
aafalo-me-gcp	Same as aafalo-me-nyc. Use aafalo-me-nyc. Kept for backward compatibility with people use this server. 🚫	DoH		🔒
aafalo-me-nyc	DNS-over-HTTPS server running dns-over-https with PiHole for Adblocking in NYC, USA. Non-logging, AD-filtering, supports DNSSEC. Hosted in New York on a RamNode Cloud instance. 🚫	DoH		🔒
abmb-sg-doh-ipv4	Non-logging and Non-filtering DoH server. Support DNSSEC. Both IPv4 & IPv6. Hosted in Singapore.	DoH		🔒
abmb-sg-doh-ipv6	Non-logging and Non-filtering DoH server. Support DNSSEC. Both IPv4 & IPv6. Hosted in Singapore.	DoH		🔒
abmb-sg2-doh-ipv4	Non-logging and Non-filtering DoH server. Support DNSSEC. Both IPv4 & IPv6. Hosted in Singapore.	DoH		🔒
abmb-sg2-doh-ipv6	Non-logging and Non-filtering DoH server. Support DNSSEC. Both IPv4 & IPv6. Hosted in Singapore.	DoH		🔒
adfree.usableprivacy.net	Public non-logging DoH server with advertising and tracker filtering. Hosted in Austria/Europe, details see: <a href="https://docs.usableprivacy.com/">docs.usableprivacy.com</a> <a href="https://docs.usableprivacy.com/">https://docs.usableprivacy.com</a> 🚫	DoH		🔒

图表 28

## 6.2 Curl 提供的 DoH 服务器列表

- 链接：<https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers>
- 共计 83 个服务器
- 提供服务器所有者、DoH 服务器请求路径、描述等信息。

### Publicly available servers

Who runs it	Base URL	Comment
A		
aafalo.me	Server US: <a href="https://dns-nyc.aafalo.me/dns-query">https://dns-nyc.aafalo.me/dns-query</a> Server EU: <a href="https://dns.aafalo.me/dns-query">https://dns.aafalo.me/dns-query</a>	Runs on Star Brilliant's <a href="#">dns-over-https</a> Both servers check for DNSSEC and block advertising
AdGuard	Default: <a href="https://dns.adguard.com/dns-query">https://dns.adguard.com/dns-query</a> Family protection: <a href="https://dns-family.adguard.com/dns-query">https://dns-family.adguard.com/dns-query</a>	Default provides ad-blocking at DNS level, while Family protection adds adult site blocking.
Alibaba Public DNS	<a href="https://dns.alidns.com/dns-query">https://dns.alidns.com/dns-query</a>	DoH/DoT/DNS Json API, Best DoH/DoT server in China
Andrews & Arnold	<a href="https://dns.aa.net.uk/dns-query">https://dns.aa.net.uk/dns-query</a>	no logging (see <a href="#">DNS Disclaimer</a> )
alekberg	Spain: <a href="https://dnses.alekberg.net/dns-query">https://dnses.alekberg.net/dns-query</a> Holland: <a href="https://dnsnl.alekberg.net/dns-query">https://dnsnl.alekberg.net/dns-query</a> Sweden: <a href="https://dnsse.alekberg.net/dns-query">https://dnsse.alekberg.net/dns-query</a>	DoH Servers in Spain, Holland and Sweden. No logging, no filtering, DNSSEC support.
armadillodns.net	<a href="https://doh.armadillodns.net/dns-query">https://doh.armadillodns.net/dns-query</a>	No source IP logging.
Association of...	<a href="#">https://doh.4215/dns-query</a>	DNSSEC, not logging queries' content, uses <a href="#">doh-proxy</a> and <a href="#">edgedns</a> for caching. Queries proxied through <a href="#">SSNN</a> network using DNS

图表 29