COMP0057 – Literature Review

Privacy Protections in Secure DNS Alternatives

based on *Oblivious DNS over HTTPS (ODoH):*
A Practical Privacy Enhancement to DNS [37]

Endong Liu[1]

University College London, endong.liu.20@ucl.ac.uk

This report is submitted as part requirement for the module COMP0057 Research in Information Security at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract. The traditional Domain Name System (DNS) protocol uses plaintext transmission over the Internet, which will leak sensitive information to third parties and resolution components. This literature review builds a threat model to the traditional DNS protocol. It evaluates the capabilities of the secure DNS alternatives for defending against passive and active adversaries based on this threat model. The finding of this review shows that none of these alternatives can fully protect user privacy in the entire resolver procedure. Finally, this review is concluded with a recommendation of using a combination scheme in the current stage and a few future research predictions.

1 Introduction

1.1 Motivation

Because many protocols on the Internet did not have security or privacy considerations when they were designed, researchers want to improve the related capabilities of these protocols. For example, Internet Protocol Security (IPSec) [3] is the network layer enhancement for IPv4, Transport Layer Security (TLS) [1] works over the transport layer, and Hypertext Transfer Protocol Secure (HTTPS) [32] encrypts the HTTP traffic. These secure protocols have been accepted by Internet Engineering Task Force (IETF) and have become an Internet standard. However, the Domain Name System (DNS) [28] [29] does not receive the same attention.

The DNS protocol maps a human-readable domain name to a machine-readable IP address. It is so fundamental to the Internet that each online activity needs DNS queries to find the service initially. Its high performance is based on the User Datagram Protocol (UDP) plaintext transmission, which is vulnerable to various attacks. Therefore, some recent research wants to design an efficient DNS protocol that can provide security and privacy. This review will investigate these DNS alternatives, evaluate their capabilities for protecting user privacy, and finally aim to understand the design principle of the recent research in this field.

1.2 Outline

This literature review will first give a reminder of the traditional DNS components and working procedure, and then build the corresponding threat model about both passive and active adversaries. Criteria for the acceptability of newly designed DNS protocols will also be defined. Based on the predefined threat model, Section 3 presents the evaluation of these DNS alternatives in the privacy design aspect, alongside the comparison and summary of their strength and weakness. Additionally, more potential issues encountered in practical use and deployment will be discussed. Finally, this review concludes the current situation of the secure and privacy-enhanced DNS protocols and predicts the possible research directions in the future.

2 Background

2.1 Traditional DNS

The traditional DNS (Do53) procedure [28] [29] involves three main components: stub resolvers, recursive resolvers, and name servers. The client-side component, stub resolvers, will send a DNS request that includes the query name (`QName`) and other query parameters (`QType`, `QClass` ...) in the `Queries` section to a recursive resolver. In a typical resolution process without caching, the recursive resolver will forward the query to name servers. Name servers are based on a tree structure such that the recursive resolver will query from the root (root servers) to the leaf (authoritative servers) recursively. The last authoritative server will return the associated IP address of the queried domain name in the `Answers` section to the recursive resolver. Finally, the recursive resolver will forward the answer to the stub resolver.

Without caching, the entire resolution procedure can be divided into two phases: communications between stub resolvers and recursive resolvers are called *Phase 1*, and communications between recursive resolvers and name servers are called *Phase 2*. An example of resolving `www.github.com` is illustrated in Figure 1.

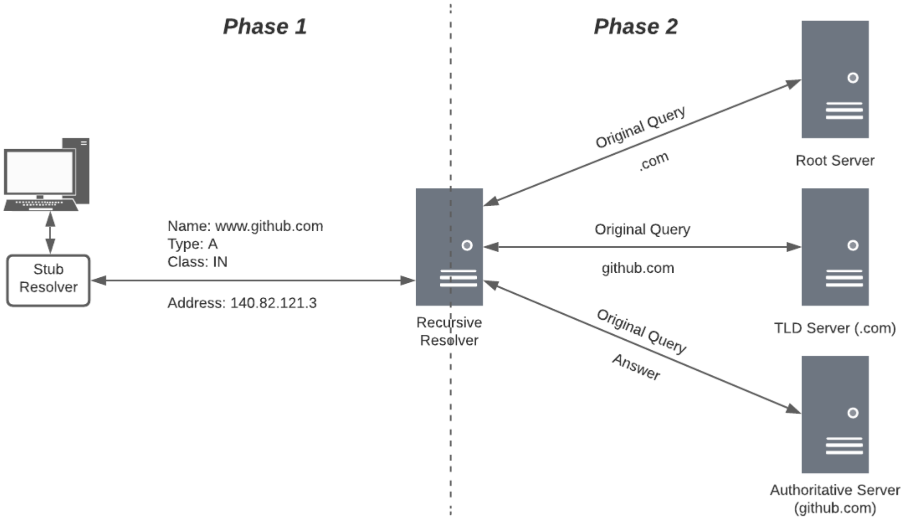

Fig. 1. Resolving www.github.com without Caching

2.2 Threat Models

To help each user across the Internet access the service when knowing the domain name, the initial design principle of Do53 indicates that the data of DNS should be public. Therefore, the implementation of Do53 did not consider either security or privacy, which means the traditional DNS traffic is in plaintext. However, with the popularity and development of the Internet, people are becoming more concerned about their personal privacy issues. Some protocols like HTTPS are developed to refine the ecosystem of secure and privacy-friendly Internet, but the DNS protocol might become one of the weakest points in this defensive circle. To fill the research gap in this area, Bortzmeyer proposes some DNS privacy considerations in RFC 7626 [4].

Privacy Threats Because an entire resolution process includes various information in different data sections, RFC 7626 [4] selects `QName` and `Source IP` as the top two parameters associated with user privacy.

The information leakage of these two parameters could happen both in *Phase 1* and *Phase 2*, but attack principals and their focuses might be different.

In *Phase 1*, third parties are the majority adversaries. There are two main reasons for their malicious behaviours in this phase. One is that the `Source IP` address currently becomes the client IP address, and another is that all DNS traffic is involved in this phase regardless of whether the caching is used (DNS performance relies heavily on caching). Therefore, adversaries can get the client information with their requested domain name (in `QName`) through passive eavesdropping. The domain name can be used to identify which service the user wants to ask for, and the user IP address would leak sensitive information like their dynamic geographic locations [14]. Adversaries would also be able to build a linkability relationship between them. This relationship will help analyze user behaviours and build their fingerprints. They can be used in large-scale surveillance or censorship from governments [17] [31] and for the profits of Internet Service Providers (ISPs) [38].

In *Phase 2*, `QName` and `Source IP` are still significant targets. The differences are that the attacking method is changed from eavesdropping to observing due to the widely-used caching, and the subject of observation is recursive resolvers and name servers rather than third parties. Because recursive resolvers can monitor the DNS traffic in both phases, they have the capabilities to link each domain name query to the corresponding client IP address. For name servers, `QName` is indispensable for the query, but its entire contents are not necessary to expose to high-level name servers except for the leaf authoritative server. For instance, the recursive resolver can only send `.com` to the root server rather than the original content `www.github.com`. Moreover, `Source IP` seems not useful for name servers since it is currently the IP address of the recursive resolver. Still, some extension techniques can allow the recursive resolver to send queries with client information. EDNS Client Subnet (ECS) [9] is designed for better performance by choosing a Content Delivery Network (CDN) geolocationally near the client IP's subnet. Then, ECS requires the recursive resolver to leak sensitive user location information to name servers for this performance goal, raising some privacy concerns. Therefore, client information is still an essential consideration for improving privacy.

Furthermore, the attacks on DNS privacy are not limited to passive eavesdropping or observations. Some attacks might be based on traffic analysis, and one direct filter uses the port number. The port number of traditional DNS is 53, and if some DNS alternatives use a dedicated port number, their traffic can be easily distinguishable. Using a standard port number is still risky but requires more sophisticated analysis techniques [10] [36] . When adversaries can identify the secure DNS traffic, they (e.g., ISPs) could refuse or redirect them for censorship or profits.

A potential active attack, called association attacks, exists when `QName` and `Source IP` are decoupled by two different entities. This attack needs the cooperation of these two entities to link `QName` and `Source IP`, then threatens user privacy again. Although there are no known real-world attacks, this is still a possible threat, especially under the pressure of governments. Some active attacks do not have an impact on privacy directly but can compromise the availability of key components in the resolving process. Because the traditional DNS uses UDP, the Denial of Service (DoS) attacks (e.g., flooding and amplification) will prevent users from using privacy-enhancing services, which leads to invalidating efforts to protect privacy.

Finally, the acceptability of the protocol technologies has a critical impact on privacy. Suppose a new DNS protocol is accepted by IETF and applied by mainstream service providers. In that case, more developers are willing to join its ecosystem to improve the capability, but more research and attacks will aim at the potential vulnerabilities that threaten privacy. In contrast, less popular protocols have less attack investigation while their anonymity set will be smaller.

Assumptions When evaluating DNS alternatives in Section 3, it is assumed that all protocols are working in an environment where the caching is disabled and the ECS extension is enabled. Thus attacks like DNS cache poisoning are not considered. This assumption is aimed to judge the privacy capability of each newly-designed DNS protocol in both *Phase 1* and *Phase 2*. Moreover, to focus on privacy issues, the following problems are serious in the real world [8] [15] but will not be mentioned in this review:

- problems or attacks caused by the underlying protocols (e.g., HTTPS and TLS), and
- problems or attacks caused by the wrong implementations for protocols or codes.

3 Privacy Evaluation of DNS Alternatives

Singanamalla et al. describe some DNS alternatives in the Background & Related Work section and compare their performance with their proposed DNS protocol [37]. This section will cover most variants mentioned by Singanamalla et al. and introduce some new protocols in recent years. These alternatives can be divided into two categories: DNS over encryption and their privacy-enhanced versions. The privacy threats described in Section 2.2 will be used as criteria to evaluate their capability in improving user privacy.

3.1 DNS over Encryption

DNSCryptv2 The components and encryption layers of DNSCryptv2 [12] are illustrated in Figure 2.

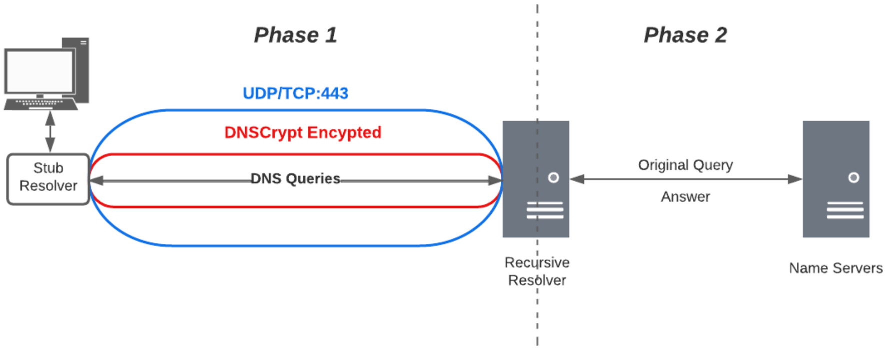

Fig. 2. DNSCryptv2

Users need to know the IP address and port number of one DNSCrypt-enabled recursive resolver to start a DNSCrypt session. The session is chosen to use UDP by default and switches to trying TCP when failures occur, which means DNSCrypt supports both UDP and TCP. Users should also know the name and public key of the service provider to verify a set of received certificates. Then, users can use the certificate to build a DNSCrypt encrypted channel for the following DNS queries and answers.

DNSCrypt uses a hybrid encryption scheme over the DNS traffic in *Phase 1*. The Curve25519 elliptic curve and the hsalsa20 hash function are used for key exchange, and the XSalsa20-Poly1305 are used as an authenticated encryption cipher [12]. Therefore, one significant parameter, `QName`, in DNS query requests cannot be eavesdropped by third parties. However, DNSCrypt does not provide any privacy protections in *Phase 2*. The recursive resolver can observe the `QName` and `Source IP` of each DNS query to build a mapping relationship. Name servers can also obtain `QName` and client subnet information when ECS is enabled.

DNSCrypt uses a padding mechanism defined in ISO/IEC 7816-4 [12] to defend against common traffic analysis attacks. It also selects 443 as its default port number over UDP and TCP. Port number 443 is a standard port number for HTTPS, and it is a challenge for adversaries to distinguish the HTTPS and DNSCrypt traffic. For defending against DNS flooding and amplification attacks, the session run over TCP will be better than over UDP [40].

DNSCrypt was published in 2015 and currently has different open-source implementations (e.g., Unbound, PowerDNS dnsdist, and AdGuard DNS) [12] on GitHub. However, its specification has not been submitted to the IETF. It also has not been applied by some sizeable public service providers (e.g., Google and Cloudflare) and is not embedded in some standard operating systems (e.g., Android and Linux). Therefore, the acceptability of DNSCrypt is questionable.

DNS over TLS (DoT) The components and encryption layers of DoT [19] are displayed in Figure 3.

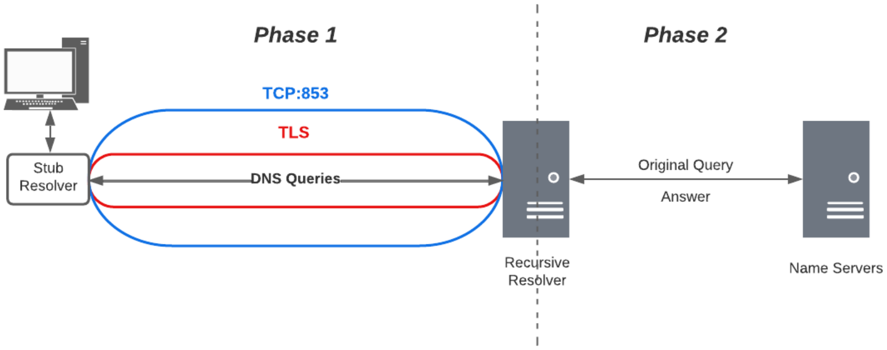

Fig. 3. DNS over TLS

DoT also protects the privacy of DNS traffic in *Phase 1*, but unlike DNSCryptv2, DoT is based on the mature TLS suites. Because attacks aimed at the underlying protocols are not considered, it can be believed that DoT hides the `QName` parameter in DNS query requests from third parties. Similar to DNSCryptv2, DoT does not support any privacy mechanisms in *Phase 2*. Both `QName` and client information would be exposed to recursive resolvers and name servers.

To balance the performance and privacy of DoT, designers provide two usage profiles for users to configure [19]. When choosing the "opportunistic" privacy profile, DoT provides privacy when necessary such that users might not validate the recursive resolver. This profile will improve the performance and availability but cannot defend against some on-path active adversaries (i.e., MITM). When applying the "out-of-band key-pinned" privacy profile, users should guarantee that the DNS traffic is transmitted to an authorized recursive resolver. After the TLS handshake, the authentication is based on X.509 certificates provided by a trusted certificate authority (CA). This choice could defend some basic active attacks but is slow to start.

DoT uses the EDNS(0) padding policy defined in RFC 8467 [27] to mitigate the traffic analysis. However, DoT selects a dedicated number 853 as its default port. The initial idea of not using original port 53 is to decrease the risk of downgrade attacks [19], but the unique port number increases the probability of being monitored or blocked. When mitigating DNS flooding and amplification attacks, the underlying transport layer protocol, TCP, has more sophisticated and diverse defences.

The specification of DoT was proposed in 2016 and have accepted by IETF. An increasing number of servers have deployed DoT, and the mainstream operating systems (e.g., Windows, macOS, Linux, iOS, and Android) also support DoT as an option for their users. Therefore, DoT has high acceptability for its ecosystem, but another alternative, DNS over HTTPS, challenges its popularity.

DNS over HTTPS ((DoH) The components and encryption layers of DoH [18] are shown in Figure 4.

DoH encodes the DNS query into the HTTP request as its content, and the client can request the DNS answer by using the HTTP `GET` or `POST` method. Then, the HTTP request is protected by TLS and requires X.509 certificates mandatorily to verify the recursive resolver, which is similar to the strict requirement of the "out-of-band key-pinned" option in DoT. In this vision, the encryption layers of DoH are a bit higher than DoT, which usually means DoH needs more bytes for the new encapsulation layer and might impact the latency. In the privacy aspect, because of the success of HTTPS, DoH can prevent outside access to the `QName` in *Phase 1* but still does not make any efforts on privacy in *Phase 2*.

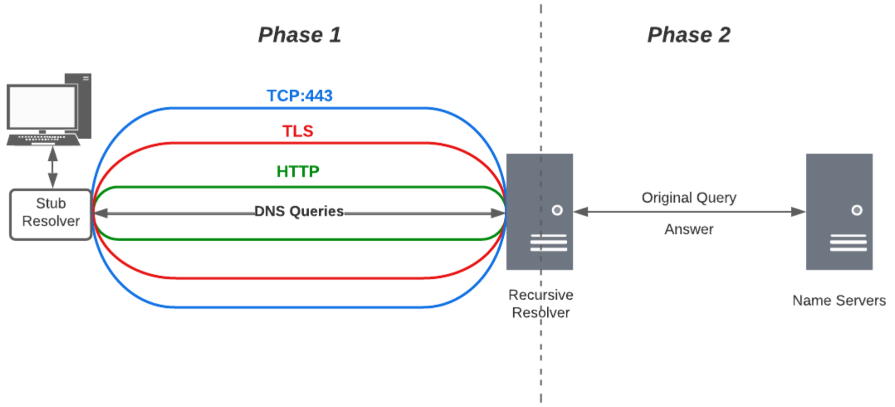

Fig. 4. DNS over HTTPS

Because the DNS query is nested in HTTPS, DoH traffic shares the same port number 443 with regular HTTPS traffic. Additionally, DoH uses EDNS(0) padding policy to reduce size-based correlation [27]. These contributions can help DoH escape from some basic traffic analysis. However, some recent research found that the above mechanisms cannot defend some side-channel traffic analysis (e.g., timing information) powered by Random Forest (RF) [36] or k-Nearest Neighbour (kNN) with Neural Networks [7]. These Machine-Learning (ML) based classification methods can distinguish DoT traffic from encrypted Web traffic, although they are more complex for adversaries to deploy. Because DoH's underlying protocols, TLS and TCP, are the same as DoT's, it can be assumed that they have the same capability to defend the DoS attacks.

RFC 8484 [18] defined the DoH protocol in 2018, and it is becoming the mainstream standard of secure DNS. Because of its HTTPS-based feature, DoH can be deployed in any HTTPS-supported Web servers and will share the same X.509 certificates with the existing websites. Moreover, the upgrade of DoH benefits from the newer Internet standard like TLS 1.3, HTTP/2 over TCP, or HTTP/3 over QUIC. Therefore, some giant server providers, Google and Cloudflare, have provided their own public DNS servers based on DoH. DoH is also supported by most browsers (e.g., Chrome, Firefox, Edge, and Opera) and regular operating systems (e.g., Windows, macOS, and iOS). With the increase in popularity, academia also focuses more on DoH. There were 30 peer-reviewed papers about DoH from 43 papers about DNS over encryption until 2022 January [26], some of them [7] [10] [36] might guide the privacy attack on DoH.

DNS over QUIC (DoQ) The components and encryption layers of DoQ [21] are illustrated in Figure 5.

QUIC [22] is a protocol designed to improve the performance and security of Internet service. The connection building process of HTTPS needs 1 Round Trip Time (RTT) in TCP handshake and 2 RTT in TLS 1.2 handshake, which increases its latency. QUIC improves this by using UDP and TLS 1.3 to support 0-RTT and 1-RTT data. Meanwhile, TLS 1.3 [33] provides security. QUIC also provides multiplexing, congestion control, and forward error corrections such that it can replace TCP in some application scenarios.

Because of QUIC's efficiency and security, DoQ is aimed to build a DNS that can balance performance and privacy. However, DoQ is designed only to protect privacy on `QName` in *Phase 1*. For performance, DoQ inherits most advantages of QUIC (e.g., 0-RTT data, multiplexing, and loss recovery). One extra privacy consideration is that 0-RTT data can be replayed by adversaries to find out the query name from the recursive resolver outgoing traffic. Although TLS 1.3 [33] recommends closing 0-RTT data by default, DoQ ignores this suggestion for significant performance gains [21]. And like DoT, DoQ also provides the same "Opportunistic" and "Strict" Usage Profiles. Furthermore, the padding policy can be selected from EDNS(0) [27] or QUIC [22], but both have some open privacy issues.

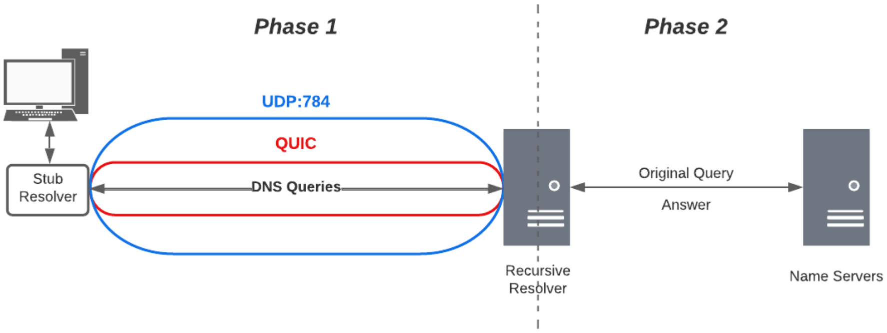

Fig. 5. DNS over QUIC

Another vulnerability is the selected port number. Older versions of DoQ [20] used number 784, which was not assigned at that time, for experimentations. The latest version (v11 access in March 2022) indicates that the port number of DoQ is to be determined (TBD), but contributors prefer to use a dedicated port number. This decision will make DoQ traffic be filtered or blocked by some access control lists (e.g., in firewall). Moreover, the underlying protocol of QUIC is UDP which cannot defend against DoS attacks effectively.

Finally, both QUIC and DoQ are newborn protocols for the Internet, and the specification of DoQ in IETF is still at active draft status. Therefore, few open-source DoQ implementations can be found online.

3.2 Privacy Enhanced DNS

QName Minimization The mitigation process of QName Minimization [5] is shown in Figure 6

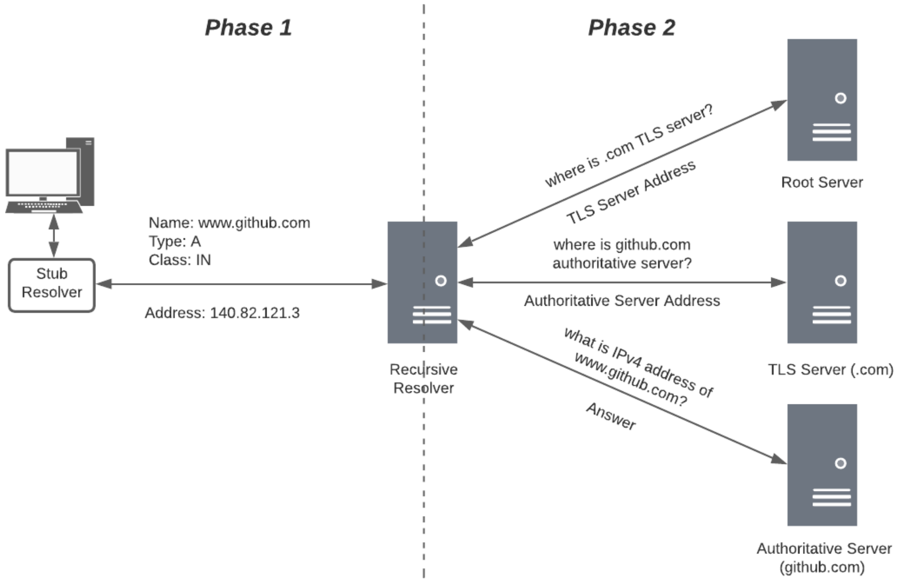
Phase 1
Phase 2
Stub Resolver
Name: www.github.com
Type: A
Class: IN
Address: 140.82.121.3
Recursive Resolver
where is .com TLS server?
TLS Server Address
Root Server
where is github.com authoritative server?
Authoritative Server Address
TLS Server (.com)
what is IPv4 address of www.github.com?
Answer
Authoritative Server (github.com)

Fig. 6. QName Minimization

QName Minimization is a specific protocol designed for improving privacy in *Phase 2*. The main goal is to provide the minimal information that is necessary to name servers at higher levels. As illustrated in Figure 6, the root server only knows that the user wants to find the address of the top-level domain name server for `.com`, and only the last authoritative server will know the full content of the queried domain name.

A new privacy consideration proposed by QName Minimization is that both `QName` and `QType` are sensitive information. Therefore, recursive resolvers should not provide the original `QType` to name servers. The design principle suggests using `NS` (Authoritative Name Server) to fill the `QType` field. However, many broken (or illegal) name servers would not give a correct response to `QType=NS`. The alternative solution is to replace `NS` with `A` (IPv4 Host Address) such that most name servers will return the correct response.

QName Minimization is an experimental protocol categorized by IETF in 2016. It provides a different option for users and recursive resolvers to protect their privacy, although it might reduce the performance when resolving deep domain names (like `www.host.group.department.example.com`) or meeting query loops. Currently, Cloudflare provides QName Minimization supports in its public DNS servers, but few open-source implementations are available online. Developers prefer to use another mechanism to protect `QName` and client information, which will be discussed in the following parts of this section.

Oblivious DNS (ODNS) The involved components of ODNS [35] are illustrated in Figure 7.

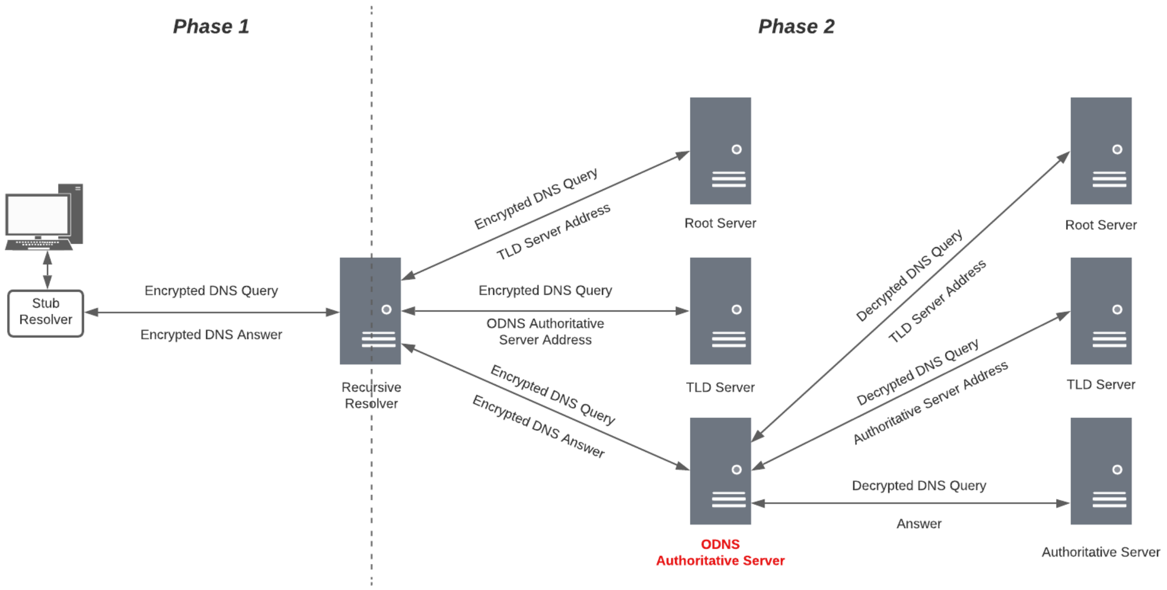

Fig. 7. Oblivious DNS

The main idea of ODNS is to decouple the `QName` from the client information. The ODNS-modified stub resolver encrypts the `QName` in DNS queries using a session key k and encrypts the session key with a previously obtained public key pk of one ODNS authoritative server: $\{www.github.com\}_k \;||\; \{k\}_{pk}$. A typical encrypted ODNS domain name is like `12kj56mns0sowd.odns.xxx.` Moreover, ODNS forcibly removes the support for ECS from the beginning of the design. Therefore, third parties cannot eavesdrop the `QName` parameter during *Phase 1* and recursive resolvers can only observe the encrypted domain name. During the first recursive resolution, the encrypted DNS query will be redirected to the ODNS authoritative server. The ODNS authoritative server will use its private key sk to get the session key k and the `QName` parameter but it cannot get the client information. Then, ODNS authoritative server will act as a new recursive resolver to finish the second recursive resolution. After it receives the answer from the final authoritative server, the answer will be encrypted by the session key k: $\{140.82.121.3\}_k$ and then sent back to the recursive resolver and the user. Therefore, recursive resolvers can only observe the client IP information and name servers (including ODNS authoritative server) can obtain the `QName`, which finally breaks the linkability.

From the client side, ODNS will use the encrypted domain name and session key to fill the `QName` field. This behaviour will cause some problems. The length of `QName` field limits the selection of symmetric key scheme and length. The ODNS designers choose AES-128, which is currently secure but has limited upgrading space. One possible solution is to include the session key into the "Additional Information" section of a DNS query, but this method still has a problem since most recursive resolvers would drop this part before querying to name servers [13]. Another vulnerability is that this slight modification would allow ODNS to use the port number 53 of the traditional DNS or reserve an unassigned port number, and use UDP as its underlying transport layer protocol. Although these details are not described by the design of ODNS, the traffic analysis attacks and DoS attacks are serious threats to ODNS.

ODNS was designed in 2018 [35], and its specification draft on IETF (v0 access in March 2022) [13] has not been updated for a long time, so some necessary details for the real-world implementations are missing. But in general, ODNS protects the privacy in the entire DNS resolving process from a new perspective.

Anonymized DNSCrypt The components of Anonymized DNSCrypt [11] are shown in Figure 8.

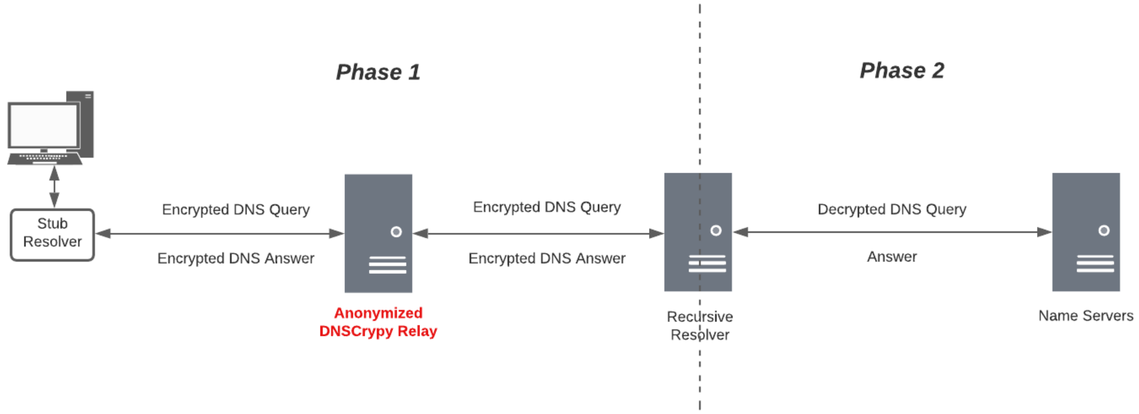

Fig. 8. Anonymized DNSCrypt

ODNS puts the obfuscation infrastructure between recursive resolvers and name servers to decouple `QName` from the client information. This design has a performance flaw that two recursive resolving processes are required without the help of caching. Anonymized DNSCrypt solves this problem by deploying the relay between the stub resolver and recursive resolvers. This relay only needs to check the encrypted DNS queries and responses' validity and then forward these messages. It does not need to implement a new recursive-resolver-like server similar to the ODNS authoritative server.

The stub resolver needs to modify the basic DNSCrypt DNS query by adding a prefix information `<anon-magic>=0xff0xff0xff0xff0xff0xff0xff0xff0x000x00` before the server address. When the relay receives the anonymized DNSCrypt query, it will decode the IPv6 server address (IPv4 address should be mapped to IPv6) and the port number. Then, the relay will check their ranges and forward the unmodified query to the corresponding recursive resolver if they are valid. For query responses, recursive resolvers can select to reply a standard DNSCrypt response or an anonymized DNSCrypt response with the prefix information `<resolver-magic>=0x720x360x660x6e0x760x570x6a0x38`. The relay also needs to check the query response to ensure that the response size is smaller than the query size. Such a checking mechanism mitigates the DNS flooding and amplification attacks to some extent.

Anonymized DNSCrypt inherits the advantages of basic DNSCrypt in *Phase 1* and enhances the capability to protect privacy in *Phase 2*. However, the involvement of the relay component leads to a new security issue. If the relay and the recursive resolver are controlled by the same entity, called association attacks, the obfuscation function of the relay will be meaningless. Because a DNSCrypt server can simulate a relay on the same IP address and port number [11] (potential self-collusion), users should consider this threat when they want to join the DNSCrypt ecosystem. Finally, the acceptability of anonymized DNSCrypt is similar to DNSCryptv2, which indicates that this protocol is currently not widely applied and not accepted by IETF.

Oblivious DNS over HTTPS (ODoH) The involved components of ODoH [37] are shown in Figure 9.

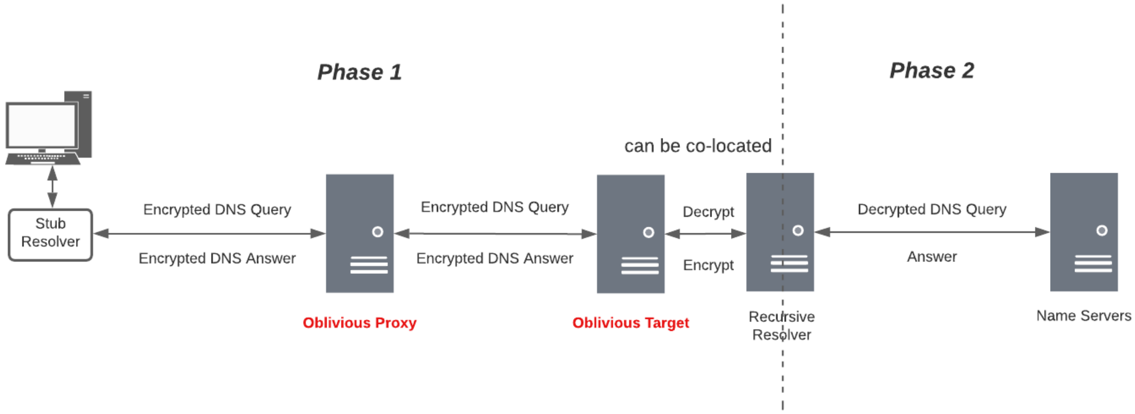

Fig. 9. Oblivious DNS over HTTPS

ODoH segments the function of each component in the resolution process. Different from DoH which supports HTTP `GET` and `POST` methods, ODoH requires the stub resolver only to use the `POST` method. ODoH also defines a new Content-Type HTTP header: `"application/oblivious-dns-message"` [24]. The ODoH-modified stub resolver needs to add this header into HTTP metadata to identify the message and contains the Uniform Resource Identifiers (URIs) of the Oblivious Proxy and the Oblivious Target in HTTPS requests. Their URI templates are shown as follow:

```
https://dnsproxy.example/dns-query{?targethost,targetpath}

https://dnstarget.example/dns-query
```

The Oblivious Proxy will forward the ODoH messages like the relay in Anonymized DNSCrypt, but it does not provide any checking functions. The Oblivious Target plays the decryption and encryption function. It also needs to verify the correctness of the Content-Type HTTP header and add this header in the ODoH query responses. This separation of duties will shift the pressure of defending against DNS DoS attacks from the recursive resolver to the target, and then reduce the resource consumption of the recursive resolver. Moreover, the Oblivious Target will apply rate-limiting and blacklist mechanisms to mitigate these attacks.

Generally, ODoH successfully decouples `QName` from the client information in *Phase 1* and uses HTTPS to protect the confidentiality of the ODoH traffic. Recursive resolvers and name servers in *Phase 2* will not get useful client information and will bring the ECS mechanism back to improve the resolving performance.

However, the latest specification draft of ODoH on IETF (v11 access in March 2022) does not specify the padding policy, and contributors have noted that ODoH will not defend ML-based traffic analysis attacks [24]. Like Anonymized DNSCrypt, the association attacks (between Oblivious Target and Proxy) are still possible. Finally, the acceptability of ODoH needs to stand the test of time. Still, the good news is that researchers have already provided the Golang and Rust implementations of Oblivious Target and Proxy and keep updating the specification draft for the IETF standardization process.

DNS over HTTPS over Tor (DoHoT) The infrastructures of DoHoT [30] are illustrated in Figure 10.

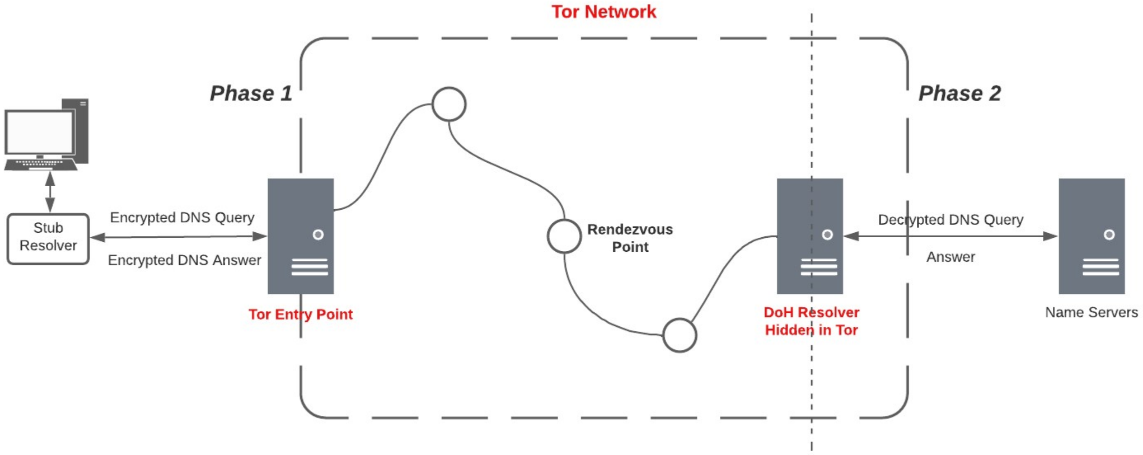

Fig. 10. DNS over HTTPS over Tor

The advocates of DoHoT believe that the single obfuscation layer (one relay or one proxy) cannot be resilient to the surveillance from both sides. Thus, DoHoT is aimed to mitigate these association attacks in Anonymized DNSCrypt and ODoH by The Onion Router (Tor). Tor uses layered encryption to protect confidentiality and integrity, and builds an unlinkability of input and output flows based on the traffic content. Cloudflare provided its DNS resolver for Tor in 2018 [34], and a typical onion service will use at least six hops to provide anonymous communication to both users and service providers. Therefore, DoHoT will support a more powerful obligatory architecture to defend the association attacks.

In general, DoHoT uses HTTPS and Tor to establish a double protection to the sensitive parameter `QName` in the DNS query. Several hops in the Tor Network will decouple the domain name from the client information, and mitigate the association attacks. The recent research [36] found that the Tor Network can be a good privacy improvement for the DoH traffic to defend against the traffic analysis attacks, although the Tor Network itself cannot efficiently mitigate traffic analysis attacks on Web traffic. Furthermore, the Tor Network is based on TCP which is regarded to be resilient to DNS flooding and amplification attacks.

However, DoHoT also includes some drawbacks of the Tor Network since most entry points and exit points of are public on the Internet. Entry points can be blocked to impact the availability of DoHoT, and the traffic from exit points is not encrypted and can be monitored. Moreover, the performance of DoHoT will be influenced by the Tor Network and the public DoH recursive resolver, this might not be accepted by normal users. With the limited population, DoH recursive resolvers can know the DNS queries must come from a small set of Tor users, which increases the probability to analyze user behaviours and build fingerprints, and eventually becomes a threat to their privacy.

3.3 Summary

For an intuitive comparison of DNS alternatives that evaluated in this review, Table 1 is shown below.

Table 1. Privacy Evaluation of 10 DNS Alternatives

DNS Alternatives	*Phase 1*		*Phase 2*		Traffic Analysis	DoS	Association	Acceptability
	`QName`	`Source IP`	`QName`	Client Info				
Do53	○	○	○	○	○	○	N/A	●
DNSCryptv2	●	●	○	○	◐	◐	N/A	◐
DoT	●	●	○	○	○	◐	N/A	●
DoH	●	●	○	○	◐	◐	N/A	●
DoQ	●	●	○	○	○	○	N/A	◐
QName Minimization	○	○	◐	○	○	○	○	◐
ODNS	●	●	○	●	◐	◐	○	○
Anonymized DNSCrypt	●	●	○	●	◐	◐	○	○
ODoH	●	●	○	●	◐	◐	○	◐
DoHoT	●	●	○	●	●	◐	●	○

N/A do not need to consider the attack

○ cannot protect the parameter / cannot defend against the attack / has been accepted rarely

◐ can partially protect the parameter / can partially defend against the attack / has been accepted

● can totally protect the parameter / can totally defend against the attack / has been accepted widely

Most DNS alternatives use encryption schemes to protect all sensitive information in *Phase 1* from passive eavesdropping. However, the work focus in *Phase 2* turns to limiting service providers' access such that they can only observe the necessary `QName` at most. Decoupling the client information from the queried domain name usually needs the obfuscation layer (a relay or a proxy), which will import the association attacks.

Before one secure DNS protocol replaces the traditional Do53 as the new mainstream, choosing a dedicated port number will simplify the surveillance and blocking. Moreover, advanced traffic analysis attacks like ML-based, and DNS DoS attacks are still open issues for the public and protocol designers. Finally, acceptability also impacts privacy. Better performance and lower practical deployment difficulties (e.g., fewer modifications, mature underlying protocols, and fewer learning costs) will make the newly-designed protocol more acceptable. However, too much attack research and too few users will negatively influence privacy.

4 Discussion

The privacy evaluation presented in Section 3 emphasizes passive attacks on sensitive information and active attacks on availability. It is under an ideal environment to investigate the privacy capability of the protocol design itself. In this discussion, more real-world influences that threaten user privacy will be covered.

Trade-off between Performance and Privacy. Most DNS protocols will compare their own performance with other variants to verify that the user experience is acceptable. Lu et al. [25] test and compare the service quality of DNS over encryption on a large scale, and find that most of them have tolerable performance. However, the comparison should not be limited to selecting which DNS alternatives. For example, DoT and DoQ provide a fallback mechanism ("strict" and "opportunistic" usage profile), while DoH and ODoH allow the client stub resolver to apply HTTPS reuse. These options inside each DNS protocol will directly influence privacy and performance. Unfortunately, the protocol designers usually prefer to give a description of this privacy threat not a statistical analysis.

Service Centralization. The traditional Do53 is a decentralized system, but the situation differs because of the lower popularity of these DNS variants. Most protocols need trust recursive resolvers to decrypt DNS queries, but the current service are provided by a few large companies (e.g., Google and Cloudflare). This centralization problem is harmful to the client sensitive information. It can be mitigated by decoupling this information from the queried domain through relays or by the K-resolver mechanism designed by Hoang et al. [16] to decrease the exposed data for each resolver. Still, service centralization will also be a challenge to availability and performance. For example, DoHoT suggests using more than eight DoH recursive resolvers for loading balance and resilience to DoS attacks.

Ethical and Regulatory Concerns. Although these secure and privacy-enhanced DNS schemes are not designed as some censorship bypassing techniques, they actually raise some ethical and regulatory concerns. Current Internet filtering and blocking mechanisms are usually used to limit access to illegal activities (e.g., child pornography) and protect users inside the network from harmful content (e.g., phishing websites and spam). Secure DNS alternatives make applying these techniques much harder, especially combining encryption and obfuscation. Similar to the Tor network, this problem makes some governments restrict the large-scale application of secure DNS [23]. Moreover, many researchers use passive DNS traffic analysis for good things (e.g., malware detection [2] [39]), although this behaviour hurts DNS privacy. The secure DNS schemes reduce the effectiveness of existing detection and trackback approaches and, finally, may cause the misuse of malicious behaviours [6] [26].

5 Conclusion

The traditional Do53 protocol is still the most popular way for domain name queries on the Internet. Because this significant infrastructure component transmits messages in plaintext, more and more attention has been paid to designing a secure and privacy-enhanced DNS protocol in recent years. In this review, the threat model of the traditional DNS protocol on privacy aspect is identified, and several newly designed protocols are evaluated based on the threat model. The finding is that most DNS alternatives can protect sensitive data from passive attacks between the client stub resolver and the recursive resolver. However, none of them can fully protect user privacy in the entire resolution process. Additionally, active attacks like traffic analysis and DoS attacks are still open issues for privacy aspects. Finally, it can be observed that the more complex and complete the privacy protection scheme, the less likely it is to be accepted at this stage.

Because of the design limitations found in this review, one practical method for current users who want to protect their privacy thoroughly is choosing a service provider that can support the combination of these DNS alternatives. For future research directions, the client stub resolver component is a missing area that has not been widely discussed and studied. Additionally, making the encrypted DNS traffic resilient to ML-based traffic analysis might be a severe challenge. Finally, some external influences can also impact the acceptability of these DNS schemes. They exceed the scope of privacy-enhancing techniques such that some human-based factors (e.g., profit-driven promotion and user experience) might play an important role.

References

1. Allen, C., Dierks, T.: The TLS Protocol Version 1.0. RFC 2246 (Jan 1999), https://www.rfc-editor.org/info/rfc2246
2. Antonakakis, M., Perdisci, R., Lee, W., II, N.V., Dagon, D.: Detecting malware domains at the upper DNS hierarchy. In: 20th USENIX Security Symposium (USENIX Security 11). USENIX Association, San Francisco, CA (Aug 2011), https://www.usenix.org/conference/usenix-security-11/detecting-malware-domains-upper-dns-hierarchy
3. Atkinson, R.: Security Architecture for the Internet Protocol. RFC 1825 (Aug 1995), https://www.rfc-editor.org/info/rfc1825
4. Bortzmeyer, S.: DNS Privacy Considerations. RFC 7626 (Aug 2015), https://www.rfc-editor.org/info/rfc7626
5. Bortzmeyer, S.: DNS Query Name Minimisation to Improve Privacy. RFC 7816 (Mar 2016), https://www.rfc-editor.org/info/rfc7816
6. Bumanglag, K., Kettani, H.: On the impact of dns over https paradigm on cyber systems. In: 2020 3rd International Conference on Information and Computer Technologies (ICICT). pp. 494–499 (2020)
7. Bushart, J., Rossow, C.: Padding ain't enough: Assessing the privacy guarantees of encrypted DNS. CoRR abs/1907.01317 (2019), http://arxiv.org/abs/1907.01317
8. Chung, T., van Rijswijk-Deij, R., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B.M., Mislove, A., Wilson, C.: A longitudinal, End-to-End view of the DNSSEC ecosystem. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 1307–1322. USENIX Association, Vancouver, BC (Aug 2017), https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/chung
9. Contavalli, C., van der Gaast, W., Lawrence, D.C., Kumari, W.A.: Client Subnet in DNS Queries. RFC 7871 (May 2016), https://www.rfc-editor.org/info/rfc7871
10. Csikor, L., Singh, H., Kang, M.S., Divakaran, D.M.: Privacy of dns-over-https: Requiem for a dream? In: 2021 IEEE European Symposium on Security and Privacy (EuroS P). pp. 252–271 (2021)
11. Denis, F.: Anonymized dnscrypt specification. https://github.com/DNSCrypt/dnscrypt-protocol/blob/master/ANONYMIZED-DNSCRYPT.txt (2020)
12. Denis, F.: Dnscrypt version 2 protocol specification. https://github.com/DNSCrypt/dnscrypt-protocol/blob/master/DNSCRYPT-V2-PROTOCOL.txt (2020)
13. Edmundson, A., Schmitt, P., Feamster, N., Mankin, A.: Oblivious DNS - Strong Privacy for DNS Queries. Internet-Draft draft-annee-dprive-oblivious-dns-00, Internet Engineering Task Force (Jul 2018), https://datatracker.ietf.org/doc/html/draft-annee-dprive-oblivious-dns-00, work in Progress
14. Guha, S., Francis, P.: Identity trail: Covert surveillance using dns. In: International Workshop on Privacy Enhancing Technologies. pp. 153–166. Springer (2007)
15. Heninger, N., Durumeric, Z., Wustrow, E., Halderman, J.A.: Mining your ps and qs: Detection of widespread weak keys in network devices. In: 21st USENIX Security Symposium (USENIX Security 12). pp. 205–220. USENIX Association, Bellevue, WA (Aug 2012), https://www.usenix.org/conference/usenixsecurity12/technical-sessions/presentation/heninger
16. Hoang, N.P., Lin, I., Ghavamnia, S., Polychronakis, M.: K-resolver: Towards decentralizing encrypted DNS resolution. CoRR abs/2001.08901 (2020), https://arxiv.org/abs/2001.08901
17. Hoang, N.P., Niaki, A.A., Dalek, J., Knockel, J., Lin, P., Marczak, B., Crete-Nishihata, M., Gill, P., Polychronakis, M.: How great is the great firewall? measuring china's DNS censorship. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 3381–3398. USENIX Association (Aug 2021), https://www.usenix.org/conference/usenixsecurity21/presentation/hoang
18. Hoffman, P.E., McManus, P.: DNS Queries over HTTPS (DoH). RFC 8484 (Oct 2018), https://www.rfc-editor.org/info/rfc8484
19. Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., Hoffman, P.E.: Specification for DNS over Transport Layer Security (TLS). RFC 7858 (May 2016), https://www.rfc-editor.org/info/rfc7858
20. Huitema, C., Dickinson, S., Mankin, A.: DNS over Dedicated QUIC Connections. Internet-Draft draft-ietf-dprive-dnsoquic-05, Internet Engineering Task Force (Jun 2018), https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-05, work in Progress
21. Huitema, C., Dickinson, S., Mankin, A.: DNS over Dedicated QUIC Connections. Internet-Draft draft-ietf-dprive-dnsoquic-11, Internet Engineering Task Force (Mar 2022), https://datatracker.ietf.org/doc/html/draft-ietf-dprive-dnsoquic-11, work in Progress
22. Iyengar, J., Thomson, M.: QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000 (May 2021), https://www.rfc-editor.org/info/rfc9000

23. Jin, L., Hao, S., Wang, H., Cotton, C.: Understanding the impact of encrypted dns on internet censorship. In: Proceedings of the Web Conference 2021. p. 484–495. WWW '21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3442381.3450084
24. Kinnear, E., McManus, P., Pauly, T., Verma, T., Wood, C.A.: Oblivious DNS Over HTTPS. Internet-Draft draft-pauly-dprive-oblivious-doh-11, Internet Engineering Task Force (Feb 2022), https://datatracker.ietf.org/doc/html/draft-pauly-dprive-oblivious-doh-11, work in Progress
25. Lu, C., Liu, B., Li, Z., Hao, S., Duan, H., Zhang, M., Leng, C., Liu, Y., Zhang, Z., Wu, J.: An end-to-end, large-scale measurement of dns-over-encryption: How far have we come? In: Proceedings of the Internet Measurement Conference. p. 22–35. IMC '19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3355369.3355580
26. Lyu, M., Gharakheili, H.H., Sivaraman, V.: A survey on DNS encryption: Current development, malware misuse, and inference techniques. CoRR abs/2201.00900 (2022), https://arxiv.org/abs/2201.00900
27. Mayrhofer, A.: Padding Policies for Extension Mechanisms for DNS (EDNS(0)). RFC 8467 (Oct 2018), https://www.rfc-editor.org/info/rfc8467
28. Mockapetris, P.: Domain names - concepts and facilities. RFC 1034 (Nov 1987), https://www.rfc-editor.org/info/rfc1034
29. Mockapetris, P.: Domain names - implementation and specification. RFC 1035 (Nov 1987), https://www.rfc-editor.org/info/rfc1035
30. Muffett, A.: Dns over https over tor. https://github.com/alecmuffett/dohot (2021)
31. Pearce, P., Jones, B., Li, F., Ensafi, R., Feamster, N., Weaver, N., Paxson, V.: Global measurement of dns manipulation. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 307–323 (2017)
32. Rescorla, E.: HTTP Over TLS. RFC 2818 (May 2000), https://www.rfc-editor.org/info/rfc2818
33. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018), https://www.rfc-editor.org/info/rfc8446
34. Sayrafi, M.: Introducing DNS Resolver for Tor (May 2018), https://blog.cloudflare.com/welcome-hidden-resolver/
35. Schmitt, P., Edmundson, A., Feamster, N.: Oblivious DNS: practical privacy for DNS queries. CoRR abs/1806.00276 (2018), http://arxiv.org/abs/1806.00276
36. Siby, S., Juárez, M., Díaz, C., Vallina-Rodriguez, N., Troncoso, C.: Encrypted DNS -> privacy? A traffic analysis perspective. CoRR abs/1906.09682 (2019), http://arxiv.org/abs/1906.09682
37. Singanamalla, S., Chunhapanya, S., Vavrusa, M., Verma, T., Wu, P., Fayed, M., Heimerl, K., Sullivan, N., Wood, C.A.: Oblivious DNS over HTTPS (odoh): A practical privacy enhancement to DNS. CoRR abs/2011.10121 (2020), https://arxiv.org/abs/2011.10121
38. Weaver, N., Kreibich, C., Paxson, V.: Redirecting dns for ads and profit. In: USENIX Workshop on Free and Open Communications on the Internet (FOCI 11) (2011)
39. Zhao, G., Xu, K., Xu, L., Wu, B.: Detecting apt malware infections based on malicious dns and traffic analysis. IEEE Access 3, 1132–1142 (2015)
40. Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., Somaiya, N.: Connection-oriented dns to improve privacy and security. In: 2015 IEEE Symposium on Security and Privacy. pp. 171–186 (2015)