

K-resolver: Towards Decentralizing Encrypted DNS Resolution

Nguyen Phong Hoang, Ivan Lin, Seyedhamed Ghavamnia, Michalis Polychronakis
Stony Brook University, New York, USA
{nhoang, ivlin, sghavamnia, mikepo}@cs.stonybrook.edu

Abstract—Centralized DNS over HTTPS/TLS (DoH/DoT) resolution, which has started being deployed by major hosting providers and web browsers, has sparked controversy among Internet activists and privacy advocates due to several privacy concerns. This design decision causes the trace of all DNS resolutions to be exposed to a third-party resolver, different than the one specified by the user’s access network. In this work we propose *K-resolver*, a DNS resolution mechanism that disperses DNS queries across *multiple* DoH resolvers, reducing the amount of information about a user’s browsing activity exposed to each individual resolver. As a result, none of the resolvers can learn a user’s entire web browsing history. We have implemented a prototype of our approach for Mozilla Firefox, and used it to evaluate the performance of web page load time compared to the default centralized DoH approach. While our *K-resolver* mechanism has some effect on DNS resolution time and web page load time, we show that this is mainly due to the geographical location of the selected DoH servers. When more well-provisioned anycast servers are available, our approach incurs negligible overhead while improving user privacy.

I. INTRODUCTION

Introduced in 1983, the domain name system (DNS) is a hierarchical naming system made up of name servers distributed across the Internet [34]. DNS maps human-memorable domain names (e.g., *example.com*) to their corresponding IP address, which are then used to route and deliver resources between connected devices. For that reason, DNS is involved in almost every online activity. The original DNS protocol, however, was not designed with security and privacy in mind. More specifically, DNS packets are not encrypted or authenticated, resulting in several security vulnerabilities and privacy risks. Man-on-the-side (MoTS) attackers can tamper with DNS responses to re-direct users to malicious hosts [1, 8, 24, 45], while state-level adversaries can passively monitor DNS traffic for surveillance or inject forged DNS responses to block unwanted connections for censorship purposes [2, 10, 12, 13, 37].

As a response to these threats, DNS over HTTPS (DoH) [16] and DNS over TLS (DoT) [19] were recently proposed and are under active development, with many web browsers and network operators already supporting them. Despite the use of different protocols to encapsulate DNS messages, these two techniques have the same purpose of preserving the integrity and

confidentiality of DNS resolutions against adversaries between a client and its recursive resolver (also referred to as recursor).

Although the benefits provided by DoH/DoT against threats “under the recursor” are clear, they come with the cost of trusting the DoH/DoT operator with the entire web browsing history of users. Recently, the design decision of Firefox [7]—one of the most popular web browsers—to centralize DNS resolutions to a single DoH server operated by Cloudflare, has led to controversy due to several privacy concerns [20, 21]. A similar plan by Google Chrome has also faced strong opposition [32].

As a step towards mitigating the privacy concerns stemming from the exposure of *all* DNS resolutions of a user—effectively the user’s entire domain-level browsing history—to an *additional* third party entity, in this work we propose *K-resolver*, a resolution mechanism in which DNS queries are dispersed across multiple (K) DoH servers, allowing each of them to individually learn only a *fraction* ($1/K$) of a user’s browsing history.

We evaluated the privacy benefit provided by this mechanism using 200 user profiles, with each profile containing 100 domain names sampled from the top one million popular websites [28]. In addition, we also evaluated the impact of our resolution scheme on the performance of web page load time. Although both DNS resolution and web page load time take slightly longer when browsing with our *K-resolver* mechanism, we show that this impact is due to the geographical location of the selected DoH recursors. If more well-provisioned anycast DoH/DoT servers are available, our proposed mechanism can be expected to provide comparable performance while also improving the online privacy of Internet users.

II. BACKGROUND AND MOTIVATION

DNS maps domain names to their IP address(es). The mapping between a domain and its associated IP address(es) is called a *resource record*, which is stored in a distributed fashion on the DNS hierarchy, as illustrated in Figure 1.

There are two traditional resolution methods to obtain the IP address(es) of a domain. The first way is via an iterative DNS resolution. For instance, to find the IP address of *example.com*, the client first queries the *root* name server to obtain the IP address(es) of the *.com* name server, then queries the *.com* name server for the IP address(es) of *example.com*, and so on. However, traversing the whole DNS hierarchy for every lookup can be costly and time consuming if the client needs to look up many different domains multiple times. Recursive DNS resolution was introduced to cope with this problem. In a recursive resolution, a *recursor* acts as a proxy between the

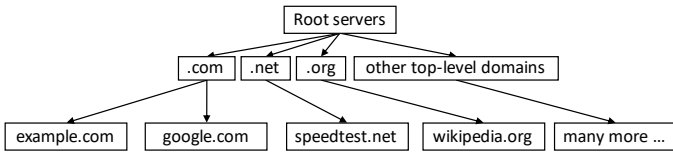


Fig. 1. The domain name system hierarchy.

client and the rest of the DNS hierarchy. When the recursor receives the query for *example.com* from the client, it will iteratively look up the IP address(es) in the same manner described above if the resource record of *example.com* is not in its cache. However, if the domain is queried by another client and was previously cached, the recursor will answer the query directly from its cache, saving the iterative lookup overhead.

As mentioned earlier, these traditional DNS resolutions are not encrypted, exposing all DNS queries and responses in plaintext, and thus leading to numerous security and privacy problems. The DoH and DoT protocols were proposed as an effort to address these problems. While these two techniques can be deployed at all levels of the DNS hierarchy to encrypt DNS messages between the involved parties, there is recently a push for the deployment of DoH/DoT “under the recursor,” with major public DNS resolvers already supporting it (e.g., Google [50], Cloudflare [52]). Although the deployment of DoH/DoT “above the recursor” at authoritative name servers is also recommended by the Internet Engineering Task Force [29], it has not been widely adopted. To the best of our knowledge, at the time of writing this paper, only the name server of Facebook has collaborated with Cloudflare to secure DNS traffic between Cloudflare’s recursors and Facebook’s name servers [5].

Encrypting DNS messages to protect them from eavesdropping or manipulation is a desirable improvement from which all privacy-conscious Internet users would benefit. However, the recent controversy is over how that encryption should be done, especially when it comes to the cost of having to trust an *additional* entity with the user’s entire browsing history, in case the web browser is configured to centralize all DNS resolutions to only one recursor—which is currently the default configuration for Mozilla Firefox [7]. This motivated us to explore the *K-resolver* approach as a resolution mechanism to remedy this “single-point-of-trust” problem. More specifically, instead of sending all DNS resolution requests to only one recursor, our mechanism allows users to distribute their DNS queries among multiple K recursors. As a result, none of the recursors can learn the entire browsing history of the user.

III. THREAT MODEL

The bottom line of recent debates around DoH/DoT is about the decision of centralizing all DNS queries into a single DoH/DoT third-party recursor by major browser vendors (Mozilla Firefox from version 62 [7], and Google Chrome from version 79 [50]). While users can configure these browsers to use any DoH recursor of their preference, they still have to trust the selected recursor with their entire web browsing history, posing a severe threat to their privacy if the recursor maliciously uses the observed DNS queries for illicit purposes.

To mitigate this privacy threat, our *K-resolver* mechanism is proposed to limit the amount of browsing information known

by each recursor. Note that our proposal does not aim to anonymize a given user’s browsing activities by decoupling the link between users and their DNS queries, which is not the original purpose of DoH/DoT either. For anonymous DNS resolutions, users may use anonymous communication systems to route their DNS traffic [15]. For instance, DNS-over-HTTPS-over-Tor is one of the more complicated methods to secure DNS traffic, which has been already implemented by Cloudflare [40].

Our threat model is oblivious to the relationship between hosting providers and operators of the recursor, who may be the same entity. Nonetheless, we consider this relationship as an orthogonal problem due to the current state of web co-location, in which the vast majority of web servers are hosted by only a handful of hosting providers [42], among which Google and Cloudflare are dominant [14]. For example, with our *K-resolver* mechanism, the domain name of a website may be resolved by a recursor that does not belong to Google or Cloudflare. However, if the website or a third-party resource of the site happens to be hosted by Google or Cloudflare, then the fact that it was visited by a particular user cannot be hidden from these same companies.

Ultimately, with the *K-resolver* resolution mechanism in place, each among K selected recursors can only obtain at most $1/K$ part of the user’s browsing history, which is still a desirable outcome for privacy-conscious users.

IV. DESIGN

As most privacy concerns around DoH stem from the design decision of major web browsers to centralize all DNS queries into a single DoH recursor, we focused on implementing our proposed mechanism in a web browser as a proof of concept. Since Mozilla is one of the first vendors to support DoH in their browser, we chose to use Firefox and modify its source code to integrate our *K-resolver* mechanism into the browser. To that end, our implementation of the *K-resolver* mechanism is solely a client-end modification of the Firefox web browser without changing any other components of the DNS hierarchy or the DoH/DoT standards.

The design of our *K-resolver* approach is centered around two main design considerations, which we discuss below.

A. Domain-recursor Pair Fixation

A particular domain name, which is entered in the URL bar of the browser (parent domain), should always be resolved by the same recursor. One of the ways to distribute DNS queries is naively rotating queries among recursors in a round-robin fashion. If the number of visited websites is small while the number of recursors is large enough, a round-robin rotation approach may achieve the design goal of our *K-resolver* mechanism over a short period of time. Unfortunately, round-robin rotation is not a sustainable solution as all selected recursors will eventually observe all domain names visited over a long period of time, making the situation even worse. Therefore, a particular parent domain should always be resolved by the same recursor, preventing other recursors from learning the visit to that domain.

Given an indexed list of K DoH recursors selected by the user, a domain name is resolved by a recursor whose index is

calculated by taking the modulo of the hashed value of that domain name concatenated with a salt value, with respect to the K number of selected recursors. For each user, the salt value, which can be configured by and is only known by that user, is generated randomly so that two different users using the same list of recursors will not end up with selecting the same recursor to resolve the same domain.

B. Parent-domain-based Recursor Selection

All DNS queries generated from within the same web page should be resolved by the same recursor of the parent domain. Rendering a web page often requires the download of several third-party resources, including images, scripts, and style sheets, as observed by previous work [3, 9, 22, 27, 35, 38, 39]. DNS resolutions of these third-party resources can be used as an effective feature to fingerprint web sites. Even when being split across recursors, some unique third-party domain names can potentially be used to trace back to the parent domain. Therefore, all the DNS queries generated due to the resolution of third-party elements of a given page should also be resolved by the same recursor of the parent domain, instead of sending them to different resolvers.

V. EXPERIMENT SETUP

In this section, we discuss how we set up and conduct experiments to analyze the privacy benefit of our K -resolver mechanism and its impact on web page load time.

A. Domain Name Dataset

We first generate 200 user profiles with each of them containing 100 unique domain names. We approximate this number based on the result from a prior study showing that most Internet users visit an average of 89 domains per month [49]. We sample the domains from the top list of one million popular websites obtained from the Tranco project on December 12th, 2019 [28]. To simulate a real-world scenario, for each user profile, we select the domains based on their popularity ranking instead of randomly picking them from the top list. Hence, a domain with a higher rank is more likely to be selected.

B. Selection of DoH Recursors

Next, we use public DoH servers provided by the Curl project to curate a list of recursors [47]. At the time of conducting our study, there are 38 operators offering 53 recursors. However, many of these recursors integrate filtering capabilities, including parental control and advertisement blocking. Moreover, some recursors are located in countries that are notorious for Internet censorship—for instance, *Rubyfish.cn*'s recursor is located in China. To make sure that DNS resolutions are not biased and tampered with throughout our experiment, we only select those active recursors that do not filter or censor any content. After excluding recursors that are not suitable for our study, we were left with a curated list of 26 DoH recursors.

C. Automated Web Crawling and Performance Measurement

Finally, with the K -resolver mechanism integrated in the Firefox browser, we visited 100 domain names with each user profile generated above. For every domain name, we launch

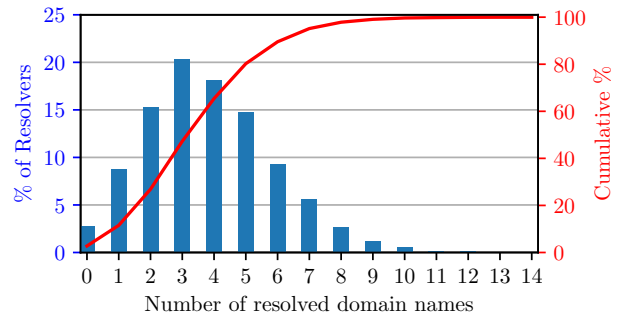


Fig. 2. CDF of domains resolved per DoH recursor.

a new Firefox session to clean cached contents of previously visited domains so that they will not have any impact on the performance metrics of the domain currently being visited. The whole process, from when the browser is launched until the web page is loaded, is handled by the Browsertime framework [44].

After completely loading the web page, Browsertime is configured to export an HTTP Archive (HAR file), containing several performance metrics which we later use for our analyses in §VI-B. Furthermore, to account for unavoidable factors that may affect our experiments, such as downgraded network performance of up-stream providers or web servers of tested domains being busy, we visited each domain at three different times, from the 18th to the 24th of December, 2019. We conducted our experiments on machines running Ubuntu 18.04.3 LTS with 32GB of RAM, 8 Intel Xeon X5450 processors, connected to a Gigabit educational network located in the East coast of the United States.

VI. DATA ANALYSIS

Using the data collected in §V, we analyze the privacy benefits provided by our K -resolver mechanism and its impact on web page load time, compared to the default setting of Cloudflare’s DoH recursor in the Firefox browser. For each of the 26 DoH recursors selected, we also investigate the effect that their geographical location has on DNS resolution time.

A. Privacy Benefit Analysis

As introduced in §III, the privacy benefit of our K -resolver mechanism is obtained by uniformly distributing one’s visited domains among K recursors, such that each recursor only learns a $1/K$ -portion of the user’s browsing history. For instance, given 100 domain names per user profile and 26 DoH recursors, each recursor should ideally resolve just 3–4 domain names.

However, in practice it is extremely hard to obtain an even distribution due to several uncertainties. For example, in a real-world scenario, we cannot know all domains that a user will visit in advance, and thus cannot guarantee that our bucket-hashing algorithm will evenly split all domains across the selected recursors. Figure 2 shows the distribution of the number of domains resolved per recursor in our experiment. The number of domains resolved by each recursor exhibits a skewed normal distribution, in which most DoH recursors (40%) resolve 3–4 domains. However, there are some corner cases in which 2.8% of recursors do not resolve any domains, while 2.2% of the recursors resolve from 8 to 14 domains.

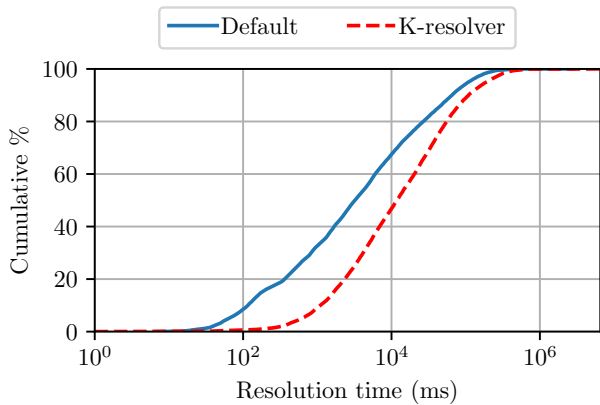


Fig. 3. CDF of the total DNS resolution time.

To examine whether our bucket-hashing algorithm can evenly disperse *sensitive* domains across recursors, we consider some domain categories as “sensitive” and check if the set of sensitive domains within the same user profile are resolved by different DoH recursors (the desirable behavior). More specifically, we use Fortiguard’s Web Filter Lookup [11] to categorize the domain names selected in §V, then consider the following categories as sensitive: pornography, armed forces, gambling, health and wellness, illegal or unethical, malicious websites, medicine, other adult materials, plagiarism, proxy avoidance, spam URLs, and weapons (sales).

Among the 200 user profiles generated in §V, we have 182 profiles containing from 1 to 6 sensitive domains each. Among these profiles, there are only 23 profiles having sensitive domains resolved by different recursors, while 159 profiles have more than one sensitive domain resolved by the same recursor.

One may consider that having more than one sensitive domain resolved by one DoH recursor is harmful for privacy. However, this is a result of the uncertainty regarding the sensitivity of a given website visited by a user, of which we do not have any control. Moreover, the sensitivity of a website can vary from site to site, depending on who, when, and from where is visiting the site [12]. Therefore, our current design of the *K-resolver* mechanism is oblivious to the sensitivity of the resolved domains. Future work may take this sensitivity aspect into consideration when assigning (hashing) the domain name to a given recursor.

B. Web Page Load Time Evaluation

Next, we assess the impact of the *K-resolver* mechanism on i) the DNS resolution time, and ii) the web page load time.

Figure 3 shows the CDF of the total amount of time (in milliseconds) that it takes for all DNS names per website visit to be resolved. The default (continuous) line represents the time when browsing with the default DoH setting of Firefox, while the *K-resolver* (dashed) line represents the time when browsing with our *K-resolver* mechanism enabled.

Böttger et al. conducted a measurement study in April 2019 to compare the resolution time between the traditional DNS and DoH for the 1,000 most popular Alexa sites [3]. Their result shows that more than 50% of web page visits take longer than 5,000ms in total to resolve all domains with Cloudflare’s

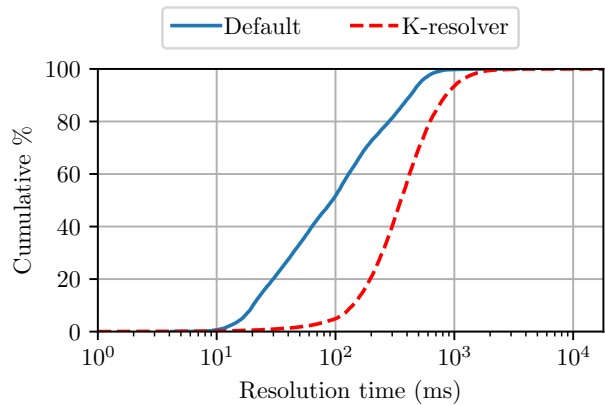


Fig. 4. CDF of the average DNS resolution time.

DoH recursor. Our experiment with the default DoH setting of Firefox, however, shows that the total resolution time has been reduced. Specifically, more than 50% of our web page visits require less than 3,600ms in total to resolve all domains.

Our *K-resolver* mechanism has a longer total resolution time compared to the original DoH setting. More than 50% of all visits have a total DNS resolution time longer than 10s, about three times more than the default DoH setting of Firefox. This is an expected result as not all DoH recursors we selected in §V support anycast, while many of them are located in a different continent than our vantage point (USA). We further investigate the impact of the geographical location of these DoH recursors on the resolution time in §VI-C.

When rendering a website, the DNS queries of involved domains can be issued in a parallel fashion [33, 46], depending on various uncontrollable factors (e.g., the underlying operating system, network conditions). Moreover, websites often have a different number of resources hosted on numerous domains. Therefore, in many cases, focusing only on the total amount of time for all DNS queries to be resolved will not reflect the actual resolution time in real-world scenarios. Since Böttger et al. [3] only report the total resolution time per website, we also capture the response time of each individual DNS resolution under both settings, i.e., the default DoH setting and the *K-resolver* mechanism.

The CDF in Figure 4 shows the average time that each individual DNS resolution takes when resolving with the default DoH setting and with our *K-resolver* mechanism. When using Cloudflare’s DoH recursor in the default Firefox setting, we observe that 50% of DNS resolutions take less than 100ms, while almost all DNS queries are resolved within 1,000ms. In contrast, when resolving with our *K-resolver* mechanism, less than 5% of the resolutions finish within 100ms, while more than 10% take longer than 1,000ms to be resolved. We further explain this discrepancy in §VI-C.

DNS resolution time directly impacts the web page load time, as domains embedded in the web page need to be resolved before the web browser can fetch resources hosted on these domains. Next, we analyze the web page load time of domains in the 200 user profiles generated in §V. The web page load time is defined as the time until the “onLoad” event is triggered.

Due to the longer DNS resolution time as shown in Figures 3

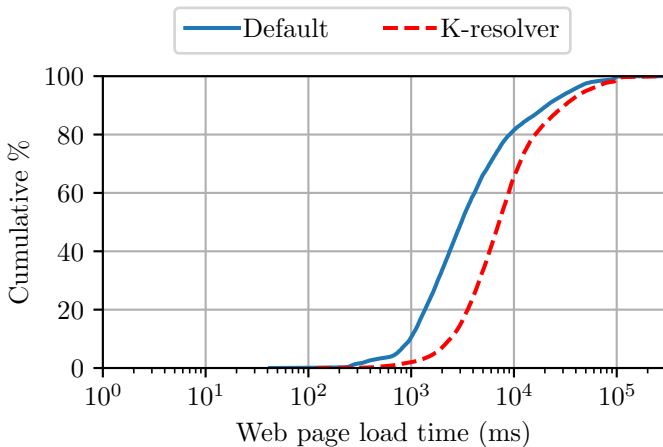


Fig. 5. CDF of the web page load time.

and 4, it is expected that most websites would take a longer time to be loaded when browsing with our *K-resolver* mechanism. As shown in Figure 5, 80% of the visits finish loading the web page within 10,000ms under the default DoH setting, while only 60% of the visits are completely loaded within the same amount of time for our approach.

C. Impact of DoH Recursors’ Geographical Location

Next, we analyze each individual DoH recursor to examine the impact of its geographical location on the resolution time. The box plot (excluding outliers) in Figure 6 shows the distribution of the resolution time of the 26 DoH recursors selected in §V. Both Cloudflare’s DoH servers are within the top five servers with the best performance, with a median resolution time ranging from 300ms to 330ms. The rest of the recursors have a median resolution time ranging from 330ms to 450ms. This discrepancy in performance among these recursors is primarily due to their geographical location. More specifically, a closer recursor to our experimental location (East Coast, USA) is more likely to provide a lower response latency.

Table I shows the geographical location, the ping time, and the median DNS resolution time of the 26 recursors used in our experiments. Except for those recursors that do not respond to our ping (ICMP) packets (whose ping time is marked as N/A), there is a correlation between the geographical location and the ping time. Notably, most recursors that are hosted on anycast IP address(es) have the shortest ping time, while unicast recursors have a longer ping time.

While the ping time to each recursor can be a good indicator for its response latency, the DNS resolution time also depends on how well-provisioned the recursor is. As shown in Table I, the top four anycast recursors with the best resolution time also have relatively small ping time. However, there are other four anycast recursors also have relatively small ping time, but are not among the top recursors with shortest resolution time. This result shows that the impact on the latency of resolution time and web page load time is mainly because of the geographical location of the recursors that we select. Assuming an idealistic future scenario in which more well-provisioned anycast DoH/DoT recursors become available, our proposed mechanism can be expected to provide comparable

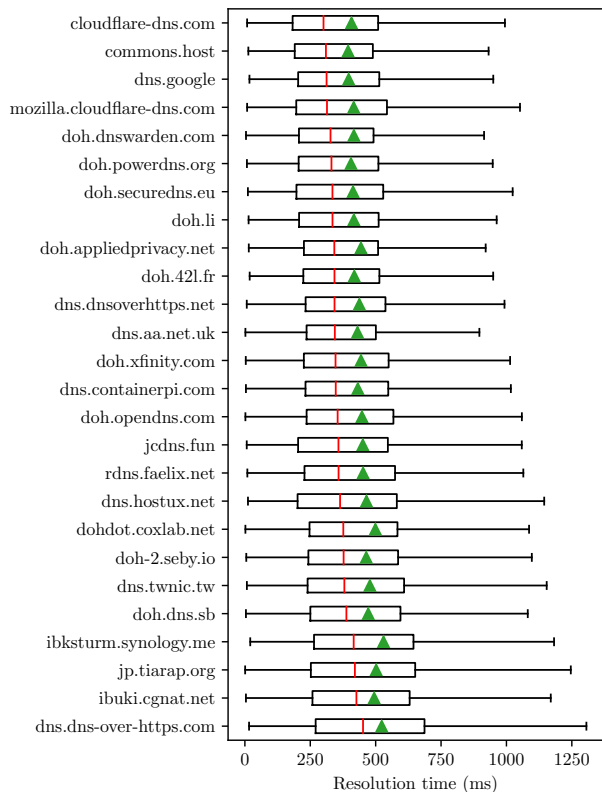


Fig. 6. The distribution of DNS resolution time of the 26 DoH recursors.

performance to the existing status quo, while also improving the online privacy of Internet users.

VII. RELATED WORK

The rapid development and deployment of DoH/DoT has attracted many researchers to study these new protocols. Recent studies mainly investigate two aspects of DoH/DoT: privacy [4, 14, 18, 36] and performance [3, 6, 30].

In terms of user privacy, researchers aim to tackle two types of adversaries. The first type includes “nosy” on-path observers, who have the privilege to monitor the encrypted traffic between the user and the DoH/DoT server, trying to predict the website being visited by the user using different traffic fingerprinting techniques [4, 14, 18]. The second type of adversary is located at the recursor, such as compromised or malicious recursors that have illicit intentions to use of the user’s online history [36].

Bushart et al. [4] find that, even with encryption, DNS traffic is still susceptible to traffic fingerprinting attacks based on packet length and count. To address this problem, several packet padding strategies have been implemented [31]. Siby et al. [43] developed a model for classifying and predicting encrypted DoH traffic using packet sizes as a key feature. The authors show that recently implemented padding strategies are not effective and still cannot cope with traffic analysis attacks.

Currently, client-side implementations of DoH/DoT still use a single recursor. This design, however, turns the selected DoH/DoT recursor into a single point of failure, but also a single point of trust. For instance, several failures of Cloudflare’s

TABLE I. GEOGRAPHICAL LOCATION INFORMATION OF THE 26 DoH RECURSORS AND THE MEDIAN OF THEIR RESOLUTION TIME.

DoH Server	Location	Ping (ms)	Resolution (ms)
cloudflare-dns.com	anycast	3.14	300
commons.host	anycast	3.99	310
dns.google	anycast	2.93	312
mozilla.cloudflare-dns.com	anycast	3.19	314
doh.dnswarden.com	DE	87.50	327
doh.powerdns.org	NL	77.90	331
doh.securedns.eu	NL	85.30	335
doh.li	UK	74.90	335
doh.appliedprivacy.net	AT	96.60	342
doh.42l.fr	FR	76.20	343
dns.dnsverhttps.net	US	73.00	343
dns.aa.net.uk	UK	71.40	344
doh.xfinity.com	US	23.53	347
dns.containerpi.com	JP	162.10	347
doh.opendns.com	anycast	3.90	355
jdns.fun	NL	85.23	358
rdns.faelix.net	anycast	82.70	358
dns.hostux.net	LU	81.60	364
dohdot.coxlab.net	US	N/A	376
doh-2.seby.io	AU	N/A	377
dns.twnic.tw	TW	230.00	380
doh.dns.sb	anycast	3.20	388
ibksturm.synology.me	CH	105.80	416
jp.tiarap.org	anycast	3.28	421
ibuki.cgnat.net	BR	137.10	426
dns.dns-over-https.com	JP	N/A	452

recursor have caused large-scale disruptions in web browsing activities [51]. To protect user data from compromised resolvers, Nakatsuka et al. [36] developed an architecture based on trusted execution environments and remote attestation to allow users to verify the resolution software running on the recursor.

Schmitt et al. [41] propose a resolution mechanism to anonymize users’ DNS queries by decoupling the link between users and their DNS queries. More specifically, the mechanism introduces an entity, called an Oblivious DNS (ODNS) resolver, residing between the user and the traditional recursor. This ODNS resolver hides the identity of the user from the traditional resolver, while not knowing any information about the domain names being queried.

While it might be expected that DoT/DoH resolutions will take longer than traditional DNS due to the need to establish a secure connection, Hounsel et. al [17] conducted a measurement study and found that this is not always the case. Similarly to our findings in VI-C, the authors find that the resolution time can vary from resolver to resolver depending on the implementations of both the DNS client and the resolver. In addition to specific resolvers, the authors find that encrypted DNS can perform just as well or even better than traditional DNS in certain network conditions. Overhead costs of establishing a TCP connection could be amortized on large loads, and TCP allows for faster retransmissions on lossy networks [17].

VIII. DISCUSSION

In our current prototype, we use bucket hashing to disperse DNS resolutions across DoH servers. However, we have shown in §VI that there are still some corner cases that would benefit from future work that will investigate more case-specific domain resolution distribution algorithms. More specifically, we observed cases in which several “sensitive” domains are

resolved by the same DoH recursor, which is not optimal as long as there are other resolvers in the pool that have not received any sensitive resolution at all. However, as mentioned earlier, the sensitivity of a domain depends on who, when, and from where visited a given website. Therefore, another domain name distribution approach that future work can investigate is to take the website category into consideration, such that domains in the same category are resolved by the same recursors.

As discussed in §IV, all subsequent DNS queries generated by rendering a web page are also resolved by the same DoH recursor that resolves the website’s parent domain. However, DNS prefetching could have a negative impact on the privacy benefit provided by our *K-resolver* mechanism. For example, consider a user who relies on a search engine to look up websites of interest. If DNS prefetching is enabled, the recursor that resolves the search engine domain name (e.g., *bing.com*, *google.com*) would also observe all domain names of interest that the user could potentially visit later. To that end, we suggest that DNS prefetching should be disabled when using *K-resolver*, as also is recommended by prior studies [25, 26].

While we opt to implement our *K-resolver* mechanism in a web browser since this is the root cause of recent debates surrounding centralized DoH/DoT, DNS resolution is generally handled by the operating system (OS). Although there have been some efforts to support DoH/DoT resolution at the OS level [23, 48], all queries are still sent to a single “trusted” resolver. We hope that our proposed mechanism will also prompt OS vendors to take a major role in adjusting the OS resolver so that multiple DoH/DoT recursors can be configured at the OS level.

IX. CONCLUSION

For more than 35 years, DNS has been one of the core protocols of the Internet. Its original unencrypted design does not take security and privacy into consideration, allowing any on-path entity between a user and the DNS recursor to learn the user’s browsing history. The DoH/DoT protocols aim to solve this very problem by transmitting DNS queries over an encrypted HTTPS/TLS channel. However, the recent design decision of major web browsers to centralize all DNS queries into one DoH/DoT recursor has sparked controversy among Internet activists and online privacy advocates.

In this work, we introduced *K-resolver*, a resolution mechanism in which DNS queries are dispersed across multiple recursors, exposing this way only a *fraction* of a user’s browsing history to each recursor. Although the impact of this resolution mechanism on web page load time is obvious, we have shown that this is primarily due to the lack of well-provisioned anycast recursors. We hope that our work will inspire further privacy-enhancing improvements to the default DoH configuration of major browsers, as well as operating systems. We believe that, in an idealistic future where the number of independent, well-provisioned anycast recursors will be larger, distributing DNS queries across multiple recursors is a promising approach for improving the online privacy of Internet users.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their thorough feedback on earlier drafts of this paper. We also thank Peter Hedenskog, the creator of sitespeed.io, for helping us with debugging the Browsertime framework in a timely manner.

REFERENCES

- [1] N. Alexiou, S. Basagiannis, P. Katsaros, T. Dashpande, and S. A. Smolka, "Formal Analysis of the Kaminsky DNS Cache-Poisoning Attack Using Probabilistic Model Checking," in *2010 IEEE 12th International Symposium on High Assurance Systems Engineering*, 2010, pp. 94–103.
- [2] S. Aryan, H. Aryan, and J. A. Halderman, "Internet Censorship in Iran: A First Look," in *USENIX FOCI 2013*, Washington, D.C.
- [3] T. Böttger, F. Cuadrado, G. Antichi, E. L. a. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An Empirical Study of the Cost of DNS-over-HTTPS," in *IMC '19*. New York, NY, USA: ACM, 2019, pp. 15–21.
- [4] J. Bushart and C. Rossow, "Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS," *arXiv e-prints*, p. arXiv:1907.01317, Jul 2019.
- [5] M. Chantra, "DNS over TLS: Encrypting DNS end-to-end," <https://engineering.fb.com/security/dns-over-tls/>, 2018, accessed: October 2019.
- [6] C. Deccio and J. Davis, "DNS Privacy in Practice and Preparation," in *ACM CoNEXT '19*, 2019, pp. 138–143.
- [7] S. Deckelmann, "What's next in making encrypted DNS-over-HTTPS the default," <https://blog.nightly.mozilla.org/2018/06/01/improving-dns-privacy-in-firefox/>, 2019, accessed: October 2019.
- [8] H.-X. Duan, N. Weaver, Z. Zhao, M. Hu, J. Liang, J. Jiang, K. Li, and V. Paxson, "Hold-On: Protecting Against On-Path DNS Poisoning," in *the Conference on Securing and Trusting Internet Names*, 2012.
- [9] M. Falahraghegar, H. Haddadi, S. Uhlig, and R. Mortier, "Anatomy of the third-party web tracking ecosystem," 2014.
- [10] O. Farnan, A. Darer, and J. Wright, "Poisoning the Well: Exploring the Great Firewall's Poisoned DNS Responses," in *Workshop on Privacy in the Electronic Society*. New York: ACM, 2016, pp. 95–98.
- [11] FortiNet, "Fortiguard labs web filter," <https://fortiguard.com/>.
- [12] N. P. Hoang, Y. Asano, and M. Yoshikawa, "Your Neighbors Are My Spies: Location and other Privacy Concerns in GLBT-focused Location-based Dating Applications." *Transactions on Advanced Communications Technology (TACT)*, vol. 5, no. 3, pp. 851–860, May 2016.
- [13] N. P. Hoang, S. Doreen, and M. Polychronakis, "Measuring I2P Censorship at a Global Scale," in *USENIX FOCI*, Santa Clara, CA, 2019.
- [14] N. P. Hoang, A. A. Niaki, N. Borisov, P. Gill, and M. Polychronakis, "Assessing the Privacy Benefits of Domain Name Encryption," 2019.
- [15] N. P. Hoang and D. Pishva, "Anonymous communication and its importance in social networking," in *16th International Conference on Advanced Communication Technology*, ser. ICACT 2014. IEEE, Feb 2014, pp. 34–39.
- [16] P. Hoffman and P. McManus, "DNS queries over HTTPS (DoH)," Internet Requests for Comments, IETF, RFC 8484, October 2018.
- [17] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Analyzing the Costs (and Benefits) of DNS, DoT, and DoH for the Modern Web," *arXiv e-prints*, p. arXiv:1907.08089, Jul 2019.
- [18] R. Houser, Z. Li, C. Cotton, and H. Wang, "An Investigation on Information Leakage of DNS over TLS," in *ACM CoNEXT '19*. New York, NY, USA: ACM, 2019, pp. 123–137.
- [19] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for DNS over transport layer security (TLS)," Internet Requests for Comments, IETF, RFC 7858, May 2016.
- [20] B. Hubert, "Centralised DoH is bad for Privacy, in 2019 and beyond," https://labs.ripe.net/Members/bert_hubert/centralised-doh-is-bad-for-privacy-in-2019-and-beyond, 2019, accessed: October 2019.
- [21] —, "DoH: (anti-)competitive and network neutrality aspects," <https://blog.powerdns.com/2019/12/03/doh-anti-competitive-and-network-neutrality-aspects>, 2019, accessed: December 2019.
- [22] M. Ikram, R. Masood, G. Tyson, M. A. Kaafar, N. Loizon, and R. Ensafi, "The Chain of Implicit Trust: An Analysis of the Web Third-Party Resources Loading," in *WWW '19*. New York, NY, USA: ACM, 2019.
- [23] T. Jensen, I. Pashov, and G. Montenegro, "Windows will improve user privacy with DNS over HTTPS," <https://techcommunity.microsoft.com/t5/networking-blog/windows-will-improve-user-privacy-with-dns-over-https/ba-p/1014229>, 2019, accessed: November 2019.
- [24] D. Kaminsky, "Catching up with Kaminsky," *Network Security*, vol. 2008, no. 9, pp. 4 – 7, 2008.
- [25] S. Krishnan and F. Monrose, "An empirical study of the performance, security and privacy implications of domain name prefetching," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems Networks (DSN)*, June 2011, pp. 61–72.
- [26] S. Krishnan and F. Monrose, "DNS Prefetching and Its Privacy Implications: When Good Things Go Bad," in *LEET '10*. USA: USENIX, 2010, p. 10.
- [27] T. Lauinger, A. Chaabane, S. Arshad, W. Robertson, C. Wilson, and E. Kirda, "Thou Shalt Not Depend on Me: Analysing the Use of Outdated JavaScript Libraries on the Web," in *NDSS*, Feb. 2017.
- [28] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhooob, M. Korczyński, and W. Joosen, "Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation," in *NDSS*, Feb. 2019.
- [29] J. Livingood, A. Mayrhofer, and B. Overeinder, "DNS Privacy Requirements for Exchanges between Recursive Resolvers and Authoritative Servers," <https://datatracker.ietf.org/doc/draft-lmo-dprive-phase2-requirements/>, 2019, accessed: November 2019.
- [30] C. Lu, B. Liu, Z. Li, S. Hao, H. Duan, M. Zhang, C. Leng, Y. Liu, Z. Zhang, and J. Wu, "An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?" in *IMC '19*. New York, NY, USA: ACM, 2019, pp. 22–35.
- [31] A. Mayrhofer, "Padding Policies for Extension Mechanisms for DNS (EDNS(0))," Internet Requests for Comments, IETF, RFC 8467, 2018.
- [32] J. D. McKinnon and R. McMillan, "Google Draws House Antitrust Scrutiny of Internet Protocol," <https://www.wsj.com/articles/google-draws-house-antitrust-scrutiny-of-internet-protocol-11569765637>, 2019, accessed: October 2019.
- [33] MDN Web Docs, "X-DNS-Prefetch-Control," <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-DNS-Prefetch-Control>, accessed: October 2019.
- [34] P. Mockapetris, "Domain Names - Concepts and Facilities," Internet Requests for Comments, IETF, RFC 882, November 1983.
- [35] T. Mueller, D. Klotzsche, D. Herrmann, and H. Federrath, "Dangers and prevalence of unprotected web fonts," in *International Conference on Software, Telecommunications and Computer Networks*, Sep. 2019.
- [36] Y. Nakatsuka, A. Paverd, and G. Tsudik, "PDofT: Private DNS-over-TLS with TEE Support," in *ACSAC '19*. New York, NY, USA: ACM, 2019.
- [37] A. A. Niaki, S. Cho, Z. Weinberg, N. P. Hoang, A. Razaghpanah, N. Christin, and P. Gill, "ICLab: A Global, Longitudinal Internet Censorship Measurement Platform," in *IEEE S&P*, May 2020.
- [38] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. Van Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "You are what you include: Large-scale evaluation of remote javascript inclusions," in *CCS '12*. New York, NY, USA: ACM, 2012, p. 736–747.
- [39] S. Patil and N. Borisov, "What Can You Learn from an IP?" in *ANRW '19*. New York, NY, USA: ACM, 2019, pp. 45–51.
- [40] M. Sayrafi, "Introducing DNS resolver for Tor," <https://blog.cloudflare.com/welcome-hidden-resolver/>, 2018, online; accessed September 2018.
- [41] P. Schmitt, A. Edmundson, A. Mankin, and N. Feamster, "Oblivious DNS: Practical Privacy for DNS Queries: Published in PoPETS 2019," in *ANRW '19*. New York, NY, USA: ACM, 2019, pp. 17–19.
- [42] C. A. Shue, A. J. Kalafut, and M. Gupta, "The Web is Smaller Than It Seems," in *IMC '07*. New York, NY, USA: ACM, 2007, pp. 123–128.
- [43] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS: Privacy? A Traffic Analysis Perspective," *arXiv e-prints*, Jun 2019.
- [44] Sitespeed, "Browsertime introduction," <https://www.sitespeed.io/>, 2019, accessed: October 2019.
- [45] J. Szurdi, B. Kocso, G. Cseh, J. Spring, M. Felegyhazi, and C. Kanich, "The long "tail" of typosquatting domain names," in *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, 2014, pp. 191–206.
- [46] The Chromium Projects, "Developer Design Documents - DNS Prefetching," <https://www.chromium.org/developers/design-documents/dns-prefetching>, 2019, accessed: November 2019.
- [47] The curl project, "Publicly available DoH servers," <https://github.com/curl/curl/wiki/DNS-over-HTTPS>, 2019, accessed: December 2019.
- [48] The DNS Privacy Project, "DNS Privacy Daemon - Stubby," <https://dnspriacy.org/wiki/display/DP/DNS+Privacy+Daemon+-+Stubby>, 2019, accessed: October 2019.
- [49] The Nielsen Consumer Survey, "How the world spends its time online," <https://web.archive.org/web/20130714132137/https://www.creditloan.com/blog/how-the-world-spends-its-time-online/>, 2010, accessed: November 2019.
- [50] M. Vale and P. Sood, "Google public DNS now supports DNS-over-TLS," <https://security.googleblog.com/2019/01/google-public-dns-now-supports-dns-over.html>, 2019, accessed: October 2019.
- [51] S. J. Vaughan-Nichols, "Cloudflare stutters and the internet stumbles," <https://www.zdnet.com/article/cloudflare-stutters-and-the-internet-stumbles/>, 2019, accessed: October 2019.
- [52] Ólafur Guðmundsson, "Introducing DNS resolver, 1.1.1.1," <https://blog.cloudflare.com/dns-resolver-1-1-1-1>, 2018, accessed: October 2019.