# Fingerprinting DNS over HTTPS (DoH)

John Choi

Georgia Institute of Technology

# Abstract

DNS over HTTPS (DoH) is a new form of DNS encryption where DNS requests are no longer in plaintext but are sent over Port 443, which is the port meant for HTTPS. The focus of this paper is mainly on determining if fingerprinting can decrypt DoH queries because DoH is built to protect and allow for DNS queries to be confidential and secure meaning not be left in plaintext. If fingerprinting methods can decrypt DoH queries, the whole premise would be invalid since an adversary could easily use fingerprinting to extract the DoH query data and make it just as weak as the current role of DNS queries now. The use of Fingerprinting methods such as ja3 and ja3s allows for the testing of fingerprinting techniques. Determining whether there are clear signs to differentiate web pages hosted on the same server is essential. Under DoH, there is enough obfuscation that differentiating web pages should not be possible. Leading to protecting the confidentiality of the specific web page a client is trying to reach. We are using the fingerprinting methods of ja3 and ja3s because all DoH requests require a TLS handshake, and even under the new TLS standard TLS 1.3, the initial handshake is in plaintext meaning the initial handshake is readable while the other handshakes after are not. The analysis will see if the specific content and web pages are readable rather than just the generic server information detailed during the initial handshake. The study will see how easy or difficult it is to identify each set of requests and compare it to other requests that are made. Using ja3 and ja3s and the results will help determine if minimal fingerprinting methods are valid in identifying and differentiating between certain web pages hosted on the same server. From the analysis, though the connected server information is public, there is no definite way to identify precisely which web page on the server a client is visiting using the MD5 hash. Since DoH only connects the web browser to the server, no specific information regarding the web page and its contents will be available to view.

# Table of Contents

# Introduction

DNS over HTTPS (DoH) called Domain Name System over Hypertext Transfer Protocol Secure is a new protocol that became a standard through Request For Comments, 8484 (RFC 8484), which is an internet standard for revealing and discussing new topics to introduce to the internet and other technology platforms.[5] DoH is used to send DNS queries to recursive DNS providers, such as  Cloudflare and Quad9 or local/national Internet Service Providers (ISPs). DoH builds upon other protocols, such as DNS over TLS (Transfer Layer Security) (DoT) and the original DNS query standard (Do53).[6] Do53 also referred to as legacy in figure 1. The current industry standard is Do53 due to its simplicity and relative ease of use and history, which includes many security flaws, one such flaw being that Do53 sends information through plaintext. The implementation of the new protocols offers additional safeguards to Internet users. Using DoH, DNS queries are sent via HTTPS, making the data confidential, thus improving security from a user's perspective. However, the usage of DoH may potentially lead to different issues. Currently, Mozilla, Google, Cloudflare, and a handful of other companies deploy DoH.[10] Because of the large market share of some of these companies, there is a potential for market concentration. For instance, requests may be sent to one of two major DoH service providers, meaning they will collect the vast majority of information from users' DNS queries. When using legacy DNS, ISPs are the primary source to receive DNS requests, but now not only do the ISPs still receive the requests, the recursive resolvers (RR) also receive the DNS request through their specialized Recursive Resolver for DoH.

The scope of this paper is to look into answering under the constraint that ISPs do not implement DoH and only have a select few DoH resolvers. Is it possible for ISPs to create simple methods of identifying/fingerprinting, such as implementing ja3/ja3s to correctly identify and differentiate the differences between web pages under the same server and negate the confidentiality of the information? From basic web sleuthing to stateless fingerprinting, are ISPs still able to collect information regarding their users? Fingerprinting is a method where certain identifiable information such as source/destination IP address or other DNS-related queries helps determine the web page being reached. Using methods such as ja3 and ja3s, identify what is being communicated between the source and destination servers and identify what the user is trying to reach. ja3/ja3s take the initial TLS handshake to identify who is sending the request and accept the request. Some methods include certificate lookups to find identifiable content sent via DoH on port 443 and reverse DNS lookups. These methods can help to identify the web page that is requested. Even if the specific web page is not determinable, limiting web pages that need to be checked against allows for a more robust check. Fingerprinting and having a smaller pool of possible candidates make it easier to identify what the user is trying to reach. Using such fingerprinting methods, we can understand the strengths and weaknesses of DoH and determine the feasibility of using DoH to protect our requests from being identified. Until recently, few papers focused on the visibility and identifiability of DoH packets. Most papers focused on the assumptions rather than focusing on the actual risk of having DoH packets over port 443.

# Background

DNS queries currently use protocol Do53, which sends DNS queries via the client's Operating System (OS), which routes the queries to firewalls if set up by an organization or person and finally sent to an ISP to complete the request.[12] As we can see in figure 1, the main difference is that DoH skips all the in-between steps from the OS's DNS table and jumps straight to the specified RR via the web browser. The biggest issue with the current model is that all information we send out is in plaintext. Do53 sends DNS queries in a human-readable format for anyone who has access to the network to view.[11] A network observer can siphon information in a DNS query, creating fear of a Man in the Middle attack (MitM) due to the weak security that Do53 offers on its DNS queries. MitM is an attack where a person sits between the client and a destination server who reads the data being sent and received. DoH solves this by adding an increased layer of security on the data by sending all DNS query information over port 443, which is for HTTPS data.[3] The importance of port 443 is the security and the ability for web pages to be secured via port 443. DoH queries are created in the web browser (currently available on Firefox, Google Chrome, and Microsoft Edge). Then the DNS queries are sent straight to the RR with no input from the client's OS DNS table and any firewall set up on the network. Setting up firewalls is usually done by companies or more tech-savvy individuals to ensure traffic sent to and from the client's router is secure. Setting up firewalls ensures that the client has another safety feature that can help to protect the client and all the machines on the specific network. Using DoH is also an excellent way to circumvent many forms of DNS-based censorship. DNS-based censorship can be seen when corporations or nation-states white/blacklist certain DNS servers to ensure only content they want to view is viewed. However, with the help of DoH, an individual can go through the web browser and use port 443 to circumvent certain censorship methods. However, if DoH is widely adopted, we may see a rise in blocking of complete IP addresses instead of just DNS blocking.[4]

When discussing fingerprinting methods, we are looking to use ja3 and ja3s. They are both a fingerprinting method created at salesforce and have successfully fingerprint a client's communication with the server and the server's response to the client.[16] They work on the protocol of TLS, where it reads the handshake that tells both parties they are willing and able to talk to one another. Since even in TLS 1.3, most information sent is in cleartext, it is easy to identify when a handshake is being done and allows for ja3/ja3s to identify where the request is sent on a high level. The fingerprinting techniques can not specifically view what is being sent and to where. Fingerprinting methods such as ja3/ja3s use the TLS version, the ciphers, extensions, and other information that allow for the fingerprinting methods to understand what is sent in the query where.

Research done on the performance of DoH compared to Do53, DoT shows that there may be a cost penalty in terms of query speed when using DoH.[6,14] The studies done by Hounsel have flaws in how they collect data and the type of network environments that the experiments are run on. Another complicating factor is the types of sites attempting to be reached and tracked. Some sites run on older protocols, namely HTTP (Hypertext Transfer Protocol), the non-secure version of HTTPS. Complications with HTTP are that the port number used for HTTP is port 80, not port 443, which HTTPS and DoH use to move packets. Using HTTP leads to packets being easily identifiable when sent since there would be the chance that packets are sent only via port 80, but enabling DoH would send packets via port 443. Using this leads to issues where DoH is a complicated protocol to run for non-HTTPS sites, which we can knock out of our research question since these sites do not run HTTPS.

Much work has been done since DoH's mention in 2018. There still needs to be a way to assess the standard and its deployment empirically. DoH can protect ISPs knowing all of the specific content we are requesting and mitigate MitM attacks; there is concern that simple fingerprinting techniques such as ja3 and ja3s may reduce the confidentiality of the new DNS query standard. Some advocates say that DoH would affect how ISPs operate, meaning that it is harder to collect and accurately identify the data. However, others say that ISPs would still be receiving the query data with some security (sending queries via port 443) in place. If ISPs were to set up a fingerprinting technique, the whole premise of DoH would be faulty. If there is still a way for ISPs to look at query data, then the argument that DoH helps prevent ISPs from collecting query data would be false. This research assumes that few DoH resolvers and ISPs have the resources to create fingerprinting techniques to figure out the specific destination of the DoH requests without having to decode the DoH request extensively. Using fingerprinting methods  to check what additional queries are made as well, as specific plug-ins and applications are run in the background of each website.

# Approach

There would be several different methods used in a waterfall methodology where we use one method, such as first picking out HTTP(Hypertext Transfer Protocol) requests. Since they are not secure, ISPs can easily access information about the webpage. In doing so, this Lessens the pool of HTTPS requests being identified. The first method would give us a rough estimate of how many web pages are not secure under DoH. We would then run a reverse DNS lookup to take an IP address to find the host address. In a regular DNS lookup, A client takes the hostname like google.com and finds the associated IP address for the hostname. Using a DNS lookup can help figure out the percentage of web pages we can name by first looking at their IP address. Using a reverse DNS lookup will give us another percentage of webpages and less protected requests under DoH. Under DoH, we should only see generic names such as Facebook or Google, but we should not see what specific content is viewed. We can continue this process of finding web pages that are insecure by checking the certificates of the pages we try to reach.

Using this information, we can now see another percentage of websites easily traceable by ISPs by merely using online tools. When we reach our final method of looking into DoH queries, there is stateless fingerprinting of websites. Stateless fingerprinting is a way to trace information about particular web pages or browsers. Then using that information, we can conclude what site is being accessed and by whom. Similar research by Gómez-Boix, Laperdrix, and Baudry, shows how easily one could identify a specific user or web page and the device they are using.[3] Their research shows that the difficulty of uniquely identifying users is complicated; it is still possible. They looked at specific websites and users, but we will be more general and have a smaller pool of possible matches for our research. We can limit our assumptions of what web pages certain users are trying to reach. Using all of these techniques, we should determine the vast majority of web pages people try to reach. Using fingerprinting methods, the percentage of web pages and servers to be identified, we can have a more solid numerical value of the protection that DoH provides for the internet and its users.

From the view of ISPs, they know who the user is by requests sent. However, DoH would make looking for the receiving server/web page more difficult—using the methods above and under the assumption that ISPs do not support DoH. In contrast, web browsers and DNS resolvers mainly support DoH. ISPs can use these techniques to figure out what web pages and servers their customers are reaching. Rather than just one institution, the ISPs collect information; there would be three institutions collecting information, ISPs, DoH query resolvers (recursive resolvers), and web browsers, which affects the confidentiality of the query data sent from the client to the server/host.

# Literature Review

This research will cover DNS Over HTTPS (DoH) (Domain Name System over Hypertext Transfer Protocol Secure). The concept of DoH, was first introduced as an RFC (Request for Comments) in October of 2018 (RFC 8484).[5] DoH is a mix of newer standards such as TLS 1.x (Transport Layer Security 1.x), and older standards such as HTTPS. The usage of DoH stems from the desire to have DNS Queries / Requests that are currently in plaintext be more secure and not seeable in plaintext format. Plaintext is a format for human-readable text. The reason this is not a good idea is that fingerprinting is easy to do, and adversaries are able to read your request details making yourself susceptible to being surveilled. There are several other standards that work to secure our DNS Requests including Do53, DoT (DNS over TLS), and many others.[6] The most influential new standards being DoH, and DoT.

There seem to be researchers on both sides of the debate on whether DoH and its close cousin, DoT, are able to do their job and protect our data from being spied on while also being efficient enough to be useful.[6] In one camp there is an argument that using anything other than the current standard of Do53, would be less efficient in speed and performance.

It states that having DoH and DoT as a standard may prevent a small amount of DNS query to be breached, but the new standards may not be accountable for the loss of speed and efficiency as we will see. When looking at the research done by Hounsel and other research groups that discussed the cons/benefits of the different standards, there seemed to be a bias on the type of data as well as the conclusion they reached. The reason why this becomes more clear and why we should look at how the data is collected is mainly due to the providers of DNS. The DNS query standard does not cause an issue; rather it is the provider of the DNS resolver method that has more to do rather than the standard itself. We can see in this blog article, they show that each different DNS resolver has different load times. This plays a major effect in the legitimacy of the Hounsel, et. al.'s research due to the reasoning that if they only used one or a few recursive resolvers. There would be no way to tell if the data they received was due to the queries being slow or the DNS provider slowing down the requests overall. Hounsel et. al. 's research shows us that we must be able to figure out all the issues that may occur. In the case of this specific research, there is the issue of not knowing where the performance issues lie and we can see it affected their data. We can learn from such issues and we can create more safety measures to ensure our data is accurate.

Looking at the security of DoH, we can see that currently both nations and corporations are unable to distinguish the difference between DoH queries and regular HTTPS queries, as DoH Queries also go through the HTTPS port of 443 and is difficult to track as to which packet being received is of which standard.[8] If such data concerns are of great risks, the current system of blocking DNS traffic from certain sites deemed not trustworthy or malicious are the easiest ways to block people from accessing sites that would otherwise be blocked and unable to route to. Using DoH, it surpasses such needs and security. With how DoH is set up, people would be able to access sites and links deemed "bad" or "harmful" to a nation or company (Hoang, et. al., 2019). Using DoH can be an advantage, because it would help prevent censorship of information that people would be trying to access. The issue would then come from when the block of not just DNS queries, but from blocking complete IP addresses from being reachable, which would cause larger issues. What would need to be done is to combine even more standards and protocols to make sure that all information is secure. When we look at Chai, et. al.'s research, we can use other standards such as ESNI (encrypted Server Name Identification).[2] This in tandem with DoH makes it possible to more tightly block and protect the site in which people may try to access if the site is censored on regular DNS querying methods. Using Chai et. al. 's research, we can see how DoH may not be a one-off protocol but may actually be merged with other security and privacy standards to have more protections for people.

Another possibility  that may arise is the issue of advertisements and creating monopolies. When we look at Weaver, Kreibich, and Paxson's research on the issue of using DNS redirecting via NXDOMAIN. NXDOMAIN happens when you incorrectly route to a wrong webpage. An example being if you type 'googel.com'  instead of 'google.com'. This becomes a way for companies to create a profit, we can see some eerily similar methodologies where specific companies are up to speed and have a stronghold on information about their customers.[13] This would cause an issue where only some companies would have a stronghold on the market of DoH, and in doing so would have most of the power to control what ads each individual sees as well as where and to whom such information would be shared with. This can be seen in the idea that only a few CDN (Content Delivery Networks), web browsers, and DNS resolvers would have the whole market to divide between a few major players making it harder for other providers to join the market.[1] Having such strong players can be much like when we look at the analysis of Weaver, Kreibich, Paxson's research. There are a few specific companies that joined the DNS redirecting market making them the whole market between 6 players. We do not want to recreate the past by doing the same thing with DoH, as there are more implications such as giving  all of our DNS query data to another third-party when ISPs (internet Service Providers) already collect such information. DoH would not stop ISPs from fingerprinting data and using it as a means to create a higher percentage of de-anonymizing our traffic. This can lead to information being found and recorded even with DoH. Rather than giving our traffic to more companies who should not have our information, we should work to secure our data and allow the least amount of people and corporations to see it. The confidentiality of our data should be the top priority. Such research has already been done with DoT, and we are able to listen in on the specific port for DoT, and gain information about where the information is going to and from.[7]

In this paper, the scope is to see how we can change the narrative and give more insight on what the chances are for large ISPs to create some quick fingerprinting technique to quickly look at someone's DNS query and to show what the effectiveness of such methods are. And how simple (or difficult) it may be for a non-tech savvy consumer to set-up DoH. Using Houser, et. al.'s research on DoT would be a good place to begin and look more deeply into how effective a small scale effort to find the query data of DoH. Additionally, look at what may be a good alternative or change in the standard. To ensure DoH is safe and secure for all internet users. And finally to tackles that may arise in regards to security and privacy. To give another angle to the recent topic of DoH would generate more ideas as well as more reasons as to why we need to look at all aspects of a new standard before it is given to the public for personal protection for privacy.

# Methodology

To collect the information, we ran a tcpdump to log traffic that comes into our machine and logs information sent from the client's machine, which runs Ubuntu version 20.04 (Focal Fossa). Tcpdump is a program that analyzes packets. Tcpdump can log both TCP/IP and other network information for the machine running the command. Using tcpdump, we can collect all of our internet logs which then runs the ja3 and ja3s fingerprinting scripts. Using the ja3 and ja3s scripts allows us to collect information about the TLS connection between the client machine and the server we are reaching out to (in our case, the main Google Search Engine page and a couple of Google suites; namely Google Drive and Google Photos). First, using TCPdump, we collected the logs in the form of a pcap on our ubuntu box from all interfaces on our machine. We then ran both ja3 and ja3s on each pcap file we recorded and saved. We then save the ja3 and ja3s outputs as JSON files. Saving the files as JSON ensures the data is more human-readable and clarifies what each request looks like bouncing between the google suite and our machine. When we run both ja3 and ja3s, we receive the Message-Digest Algorithm 5 (MD5) hash for each request and as seen in figures 2 - 7. The hashes for all of the ja3s are the same, meaning that we cannot differentiate all of the webpages we reached, all of the google suites: google's Search Engine, Google Photos, and Google Drive. An MD5 hash is a way to transform a string length into a 128-bit value and compare and verify the integrity of a file. If the MD5 hash is incorrect, then the file's integrity is void, and we can no longer ensure the information being presented.

# Discussion

When we look at figures 2-7 for the hashes, we can see that for all of the ja3s hashes; the MD5 hashes are the same, meaning whichever google suite we ping and connect. Even though the IP addresses for the Google suites are different, the hashes are identical.  For how identifiable that specific instance is, it is not possible by looking at just the ja3s provided information. The hashes are the same since we cannot accurately differentiate between the Google suites and the Google Search Engine page. The only thing that we can tell as a main identifiable aspect is the number of requests sent between the server and the client. When it came to accessing Google's Search Engine, the queries were limited, and fewer bytes were transferred for the request. For the Google suites, Google Drive, and Google Photos, the request had more bytes sent to and from our machine. Besides this identifying factor, there was no way to identify whether we were pinging a specific Google suite product or just the Google Search Engine.

One limitation we had was the scope of this paper. Doing this on a personal network and not on an academic network, there may be some issues with logging the pcaps because we are not on a dedicated network for our research to be on solely. Additionally, there may have been some noise/traffic on our network that was not relevant to our research, and the machine is actively used. Another limitation is the small sample size. We only ran this on 2 Google suite products as well as the main Google Search Engine. If we were also to try logging pcaps from the other Google suite products, we might get different results than what we have at the moment. There may be more factors we have not accounted for, leading to other observations we were unable to make.

There are simple ways to identify DoH traffic and packets easily from the current work done, but ja3/ja3s do not help identify the specific web page. With the current state of DoH and the technology currently used to power the protocol, there need to be additional security inputs to ensure that DoH packets are more secure and less likely to be identified by third-party systems and scripts. From the works of Frank Nijeboer, there seems to be an abundance of ways to identify DoH packets by looking at the packet sizes and more advanced techniques.[15] Since DoH is becoming the standard for DNS requests; there should be more thought put into how to securely and safely send and receive DoH traffic.

# Conclusion

In this research, we answer the question, “how feasible is it to use a range of identification/fingerprinting methods. Using methods like ja3 and ja3s to view and identify contents of DoH packets and identify the specific web page to which the traffic is traveling.

From all the data collected and analyzed, we can see that even if not all the data is accurate, there is a good chance of knowing where the packets are sent in a general sense. Meaning we can identify the packets with good accuracy and give us a general idea of the flow of traffic, and where it reaches, but never shows us the specific web page we are visiting. Seeing the fingerprinting method leaves DoH with more to be desired. However, using the fingerprinting method of ja3 and ja3s, correctly labeling and differentiating between the Google Search Engine’s web page from other Google suite products such as Google Drive and Google Photos was impossible. The reason for not identifying is primarily due to how DoH works, where we connect the web browser straight to the server via a RR instead of connecting through the OS DNS table. The process makes it so that when we try to see where the client connects to the server, it will always show the web browser connecting to the requested server, but never the specific web page.

With the research done, there is still hope for DoH to be more secure and give the users peace of mind that their information is truly secure, ensuring that no one sees traffic sent out to the internet. Once more secure protocols are created, much like TLS 1.3, the possibility of DoH being more secure such as obfuscating all other handshake messages after the initial SeverHello.[9] Additionally, creating ways to mask the DOH requests within the HTTPS port may be another way to successfully heighten the integrity of using DoH for the general public.

# Future Direction

Future research in terms of fingerprinting on DoH should be done to look at a broader number of browsers and RRs (that support DoH) and a broader number of sites to see if other markers can easily signify a specific site. This research gives us more insight into the ease of use for non-tech savvy users, simplifies the process, and secures DNS into a reality for more people.

Additionally, we can do additional research in more advanced fingerprinting techniques to correctly identify the DoH packets and their destination. Using machine learning and the models created, we can make the task automated. There is also the possibility of the work being wholly automated and making the best possible suggestions to identify destinations successfully.

# Acknowledgments

Thank you to Professor Milton Mueller for providing the spark to begin research on this topic as well as giving the initial seeds of thought on how to look at DoH from a policy aspect, and Professor Paul Pearce for the more technical guidance as well as relevant standards. I also would like to thank Brenden Kuerbis for being the one to guide me on my more daily tasks as well as being the individual to contact for additional resources.

# References

1. Borgolte,K., Chattopadhyay,T., Feamster,N., Kshirsagar,M., Holland,J., Hounsel,A. & Schmitt,P. (2019), "How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem." *SSRN*. 27 July. https://ssrn.com/abstract=3427563.
2. Chai,Z., Ghafari,A. & Houmansadr,A. (2019)." On the importance of encrypted-sni (ESNI) to censorship circumvention". In 9th USENIX Workshop on Free and Open Communications on the Internet (FOCI 19), Santa Clara, CA. USENIX Association
3. Gómez-Boix,A., Laperdrix,P., & Baudry,B. (2018) "Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale." WWW2018 - TheWebConf 2018 : 27th International World Wide Web Conference, Apr 2018, Lyon, France. pp.1-10, ff10.1145/3178876.3186097ff. Ffhal-01718234v2f
4. Hoang,N. P., Doreen,S. and Polychronakis,M. (2019) "Measuring I2P Censorship at a Global Scale," in USENIX FOCI, Santa Clara, CA.
5. Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DOH)", RFC 8484, October 2018, <https://tools.ietf.org/html/rfc8484>
6. Hounsel,A., Borgolte,K., Schmitt,P., Holland,J. and Feamster,N.(2019). Analyzing the costs (and benefits) of DNS, DoT, and DoH for the modern web. arXiv preprint arXiv:1907.08089.
7. Houser,R., Li,Z., Cotton,C. & Wang,H. (2019). "An Investigation on Information Leakage of DNS over TLS," in ACM CoNEXT '19. New York, NY, USA: ACM, pp. 123–137.
8. Lu,C., Liu,B., Li,Z., Hao,S., Duan,H., Zhang,M., Leng,C., Liu,Y., Zhang,Z. &Wu,J. (2019). An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?. In Proceedings of the 2019 Internet Measurement Conference (IMC '19). ACM.
9. Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, August 2018, <https://tools.ietf.org/html/rfc8446>
10. Siby,S., Juarez,M., Diaz,C., Vallina-Rodriguez,N. and Troncoso,C. (2019), "Encrypted DNS --> Privacy? A Traffic Analysis Perspective".
11. Siby,S., Juarez,M., Vallina-rodriguez,N. and Troncoso,C. (2018), "DNS privacy not so private: the traffic analysis perspective," The 11th Workshop on Hot Topics in Privacy Enhancing Technologies, pp. 3– 4.
12. van Heugten,J.,(2018), "Privacy analysis of dns resolver solutions," Master's thesis, University of Amsterdam.
13. Weaver,N., Kreibich,C. AND Paxson,V. (2011)"Redirecting DNS for ads and profit". In USENIX Workshop on Free and Open Communications on the Internet (FOCI), San Francisco, CA, USA.
14. Wijenbergh,J. (2019) "Performance comparison of DNS over HTTPS to unencrypted DNS," Bachelor's thesis, Radboud University.
15. Nijeboer, FJ. (2020) "Dectection of HTTPS encrypted DNS traffic", University of Twente
16. "TLS Fingerprinting with JA3 and JA3S","John Althouse", available Online: [https://engineering.salesforce.com/tls-fingerprinting-with-ja3-and-ja3s-247362855967]

# Appendix

The appendix contains figures and screenshots of all the ja3 and ja3s json outputs, They are described in the discussion portion of the paper found in section []

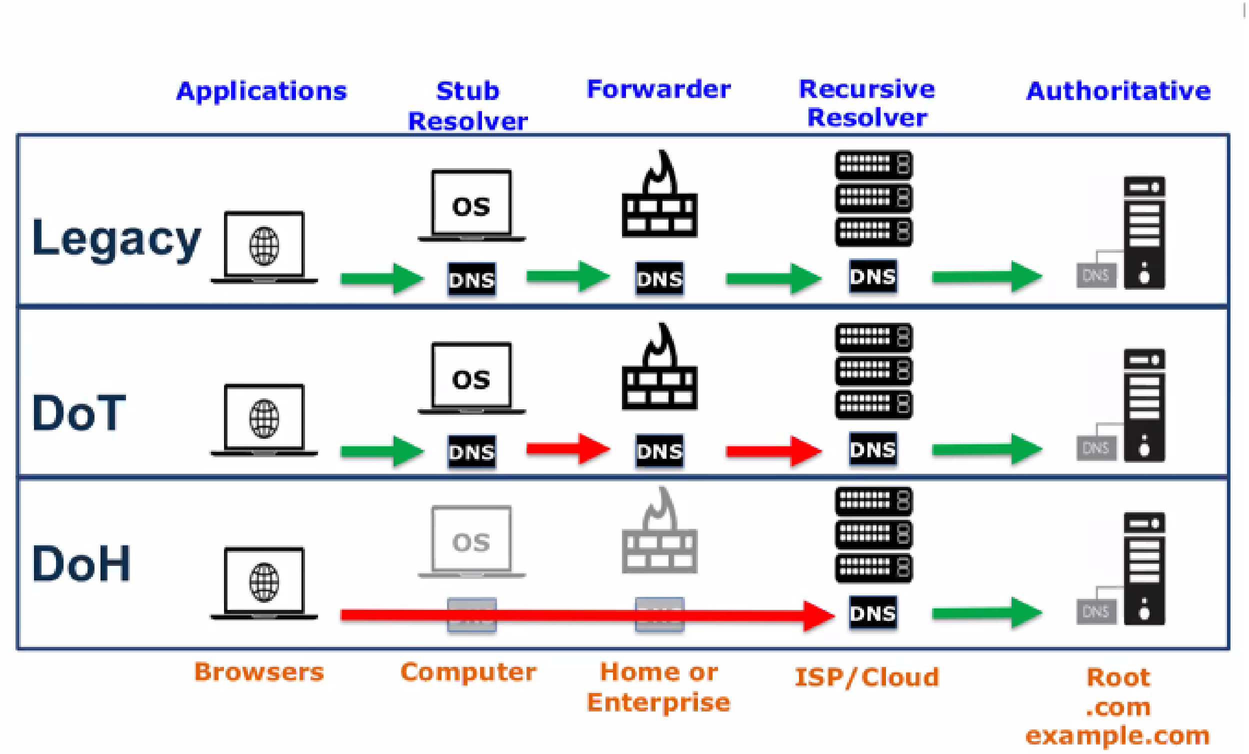
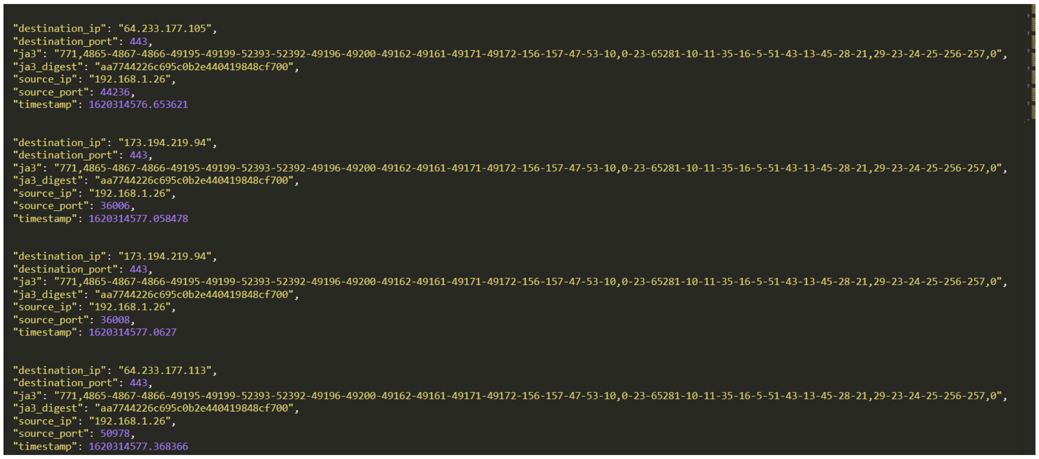
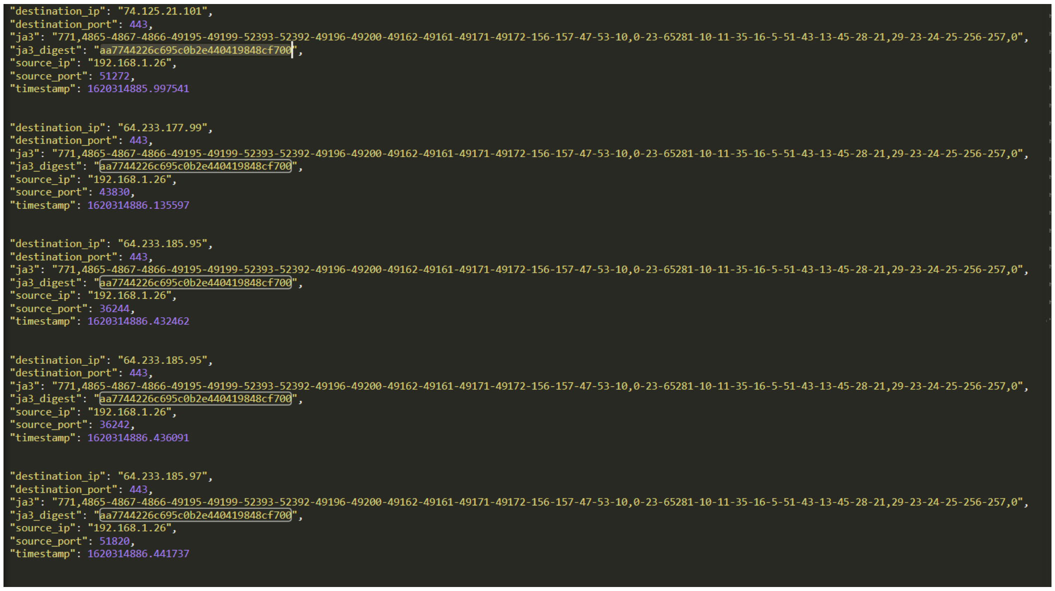


Figure 1. Do53, DoH, and DoT. How each process is done and which steps are not skipped over when using each DNS querying method



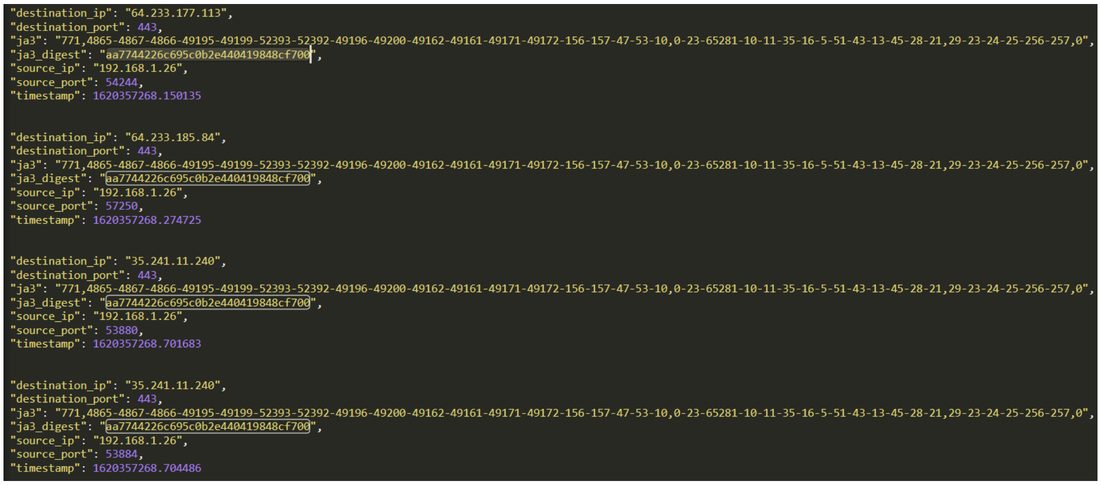
```
"destination_ip": "64.233.177.105",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 44236,
"timestamp": 1620314576.653621


"destination_ip": "173.194.219.94",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 36006,
"timestamp": 1620314577.058478


"destination_ip": "173.194.219.94",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 36008,
"timestamp": 1620314577.0627


"destination_ip": "64.233.177.113",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 50978,
"timestamp": 1620314577.368366
```

Figure 2. Google Search engine ja3 fingerprinting excerpt shows that majority of ja3 MD5 hashes are the same but have different destination ports



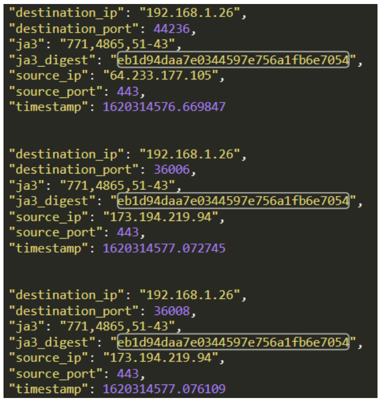
```
"destination_ip": "74.125.21.101",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 51272,
"timestamp": 1620314885.997541


"destination_ip": "64.233.177.99",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 43830,
"timestamp": 1620314886.135597


"destination_ip": "64.233.185.95",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 36244,
"timestamp": 1620314886.432462


"destination_ip": "64.233.185.95",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 36242,
"timestamp": 1620314886.436091


"destination_ip": "64.233.185.97",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 51820,
"timestamp": 1620314886.441737
```

Figure 3. Google Suites, Google photos ja3 fingerprinting excerpt shows that majority of ja3 MD5 hashes are the same but have different destination ports

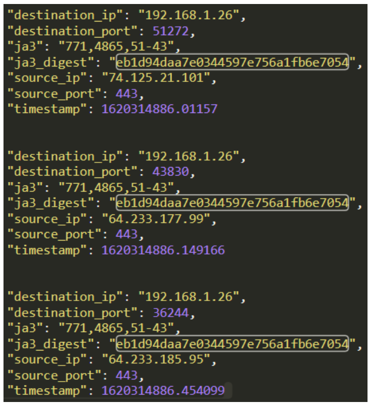


```
"destination_ip": "64.233.177.113",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 54244,
"timestamp": 1620357268.150135


"destination_ip": "64.233.185.84",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 57250,
"timestamp": 1620357268.274725


"destination_ip": "35.241.11.240",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 53880,
"timestamp": 1620357268.701683


"destination_ip": "35.241.11.240",
"destination_port": 443,
"ja3": "771,4865-4867-4866-49195-49199-52393-52392-49196-49200-49162-49161-49171-49172-156-157-47-53-10,0-23-65281-10-11-35-16-5-51-43-13-45-28-21,29-23-24-25-256-257,0",
"ja3_digest": "aa7744226c695c0b2e440419848cf700",
"source_ip": "192.168.1.26",
"source_port": 53884,
"timestamp": 1620357268.704486
```

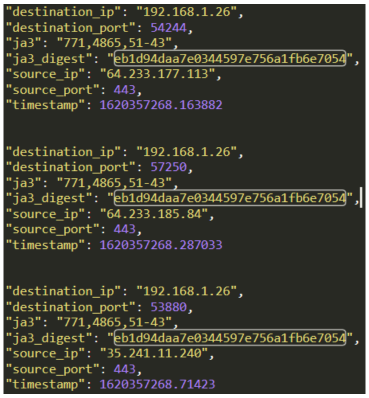
Figure 4. Google Suites, Google Drive ja3 fingerprinting excerpt shows that majority of ja3 MD5 hashes are the same but have different destination ports



```
"destination_ip": "192.168.1.26",
"destination_port": 44236,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "64.233.177.105",
"source_port": 443,
"timestamp": 1620314576.669847


"destination_ip": "192.168.1.26",
"destination_port": 36006,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "173.194.219.94",
"source_port": 443,
"timestamp": 1620314577.072745


"destination_ip": "192.168.1.26",
"destination_port": 36008,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "173.194.219.94",
"source_port": 443,
"timestamp": 1620314577.076109
```



```
"destination_ip": "192.168.1.26",
"destination_port": 51272,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "74.125.21.101",
"source_port": 443,
"timestamp": 1620314886.01157


"destination_ip": "192.168.1.26",
"destination_port": 43830,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "64.233.177.99",
"source_port": 443,
"timestamp": 1620314886.149166


"destination_ip": "192.168.1.26",
"destination_port": 36244,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "64.233.185.95",
"source_port": 443,
"timestamp": 1620314886.454099
```



```
"destination_ip": "192.168.1.26",
"destination_port": 54244,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "64.233.177.113",
"source_port": 443,
"timestamp": 1620357268.163882


"destination_ip": "192.168.1.26",
"destination_port": 57250,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "64.233.185.84",
"source_port": 443,
"timestamp": 1620357268.287033


"destination_ip": "192.168.1.26",
"destination_port": 53880,
"ja3": "771,4865,51-43",
"ja3_digest": "eb1d94daa7e0344597e756a1fb6e7054",
"source_ip": "35.241.11.240",
"source_port": 443,
"timestamp": 1620357268.71423
```

Figures 5-7. (L-R) Google Search Engine, Google Photos, Google Drive. All of these outputs show that the server's response and the hash they send out is the same "eb1d94daa7e0344597e756a1fb6e7054". This is due to the fact that the source IP is all to the same location (Google servers).