A Study on DNS Architecture Realizing Institutional Privacy Preservation

Satoru Sunahara

A Thesis Submitted to Graduate School of Information Science and Technology in fulfillment of the requirements for the degree of Doctor of Information Science and Technology at the

HOKKAIDO UNIVERSITY

February 2026

Preface

This thesis aims to achieve comprehensive privacy protection for communications in the Domain Name System (DNS), an essential service in the modern Internet environment.

While existing encryption protocols (DoT, DoH, and DoQ) have made great progress in encrypting communications between end devices and DNS full-service resolvers, current research has not adequately addressed the protection of institutional privacy. The communication path between DNS full-service resolvers and authoritative DNS servers remains unencrypted, leaving significant vulnerabilities. This situation indicates that institutional privacy is effectively unprotected in the current Internet infrastructure.

Institutional privacy may include query data about sensitive activities that are kept private or political activities related to ideological or value-based activities. If such sensitive activities were exposed through traffic analysis, it could lead to a significant decline in public trust and serious reputational damage. Furthermore, the leakage of such information could adversely affect an institution's strategic confidentiality and public image.

This study looks at a privacy problem: current standards (RFC 9461 and RFC 9539) always have some unprotected data transfer when finding protocols

that can encrypt data.

To fix these related things, this study puts forward two ways to allow DNS encryption across the entire path. First, it makes use of the DoH request path. This creates a way for resolvers to show the privacy choice of devices, whether they want privacy or compatibility more. Second, it introduces a way to put encryption protocol details, which supported by authoritative DNS servers, into NS records. This allows safe and quick protocol finding without using unprotected data transfer.

Chapter 1 explains why this study was done and the problems it addresses. It focuses on the continuing privacy risks on unprotected DNS routes. Chapter 2 provides an overview of studies on privacy for both end-users and organizations. It points out the limits of current standard specifications (RFC 9461 and RFC 9539), specifies that unencrypted data transfer happens when finding protocols, and user choice is not reflected in protocol choice. Chapter 3 describes the design and test version of a Full-DoH setup. It shows that good performance is possible and finds that checking certificates uses the most performance resources. Chapter 4 proposes an extended framework that shares protocol information through NS records, and shows that superior name resolution performance can be achieved while maintaining privacy, in comparison with RFC 9461. Chapter 5 concludes that the proposed setup gets past the limits of present methods, allowing steady privacy protection during the DNS resolution process. It also talks about problems for the future, like protecting communication header details such as Server Name Indication (SNI).

The results of this study represent an important step toward realizing comprehensive privacy protection in DNS communication, and are expected to contribute to improving the security and reliability of the Internet.

Acknowledgments

I am sincerely grateful to Professor Katsuyoshi Iida (Hokkaido University) for his continuous support during this research. His comments during our discussions, especially regarding the refinement of my research focus, were extremely helpful. Through his guidance, I have consistently received pertinent suggestions regarding the direction of the research, and I have learned greatly with respect to academic attitudes and perspectives.

I would also like to extend my sincere appreciation to Professor Yoshiaki Takai and Professor Hiroyuki Minami (Hokkaido University). Their advice during the preparation of this thesis helped improve the clarity and coherence of the research. In addition, I am deeply grateful to Associate Professor Yong Jin (Institute of Science Tokyo) and Professor Nariyoshi Yamai (Tokyo University of Agriculture and Technology) for providing constructive comments and opportunities for academic discussions.

Furthermore, I am grateful to Lecturer Hikaru Ichise of Oita University and Visiting Researcher Rikima Mitsuhashi of Hokkaido University for their active participation in research group discussions. Their comments, particularly regarding the organization and development of this research, helped me refine the structure of the thesis and improve its coherence. Their support also

made my daily research activities much more manageable.

I am also deeply indebted to Professor Shigeki Hagihara of Chitose Institute of Science and Technology. His advice and encouragement played an important role in my decision to pursue a doctoral program, and his continued support has been a constant source of motivation throughout my academic journey.

Finally, I extend my deepest appreciation to my family, who have supported me throughout this long research journey, both emotionally and in everyday life. I am especially grateful to my wife, Noriko, for her unwavering support and patience.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Objective in this study

The Internet was not originally designed with the expectation that it would evolve into a critical social infrastructure as it is today. Rather, it was conceived as an experimental network that prioritized interoperability and availability among a limited community of users. In its early design, the foremost objective was to enable flexible interconnection among multiple networks and to support large-scale communication services. Consequently, security requirements such as the confidentiality of transmitted data and user authentication were not incorporated into the fundamental design principles.

The suite of Internet protocols developed under this design philosophy implicitly assumed that the nodes along the communication path were trustworthy and adopted an architecture that emphasized reachability and fault tolerance. As a result, data were generally transmitted in plaintext, creating conditions under which third parties positioned along the communication path could technically obtain the contents of communications with relative

ease. While these characteristics did not pose significant concerns in restricted usage environments such as research institutions and government agencies, they became serious security challenges with the commercialization of the Internet and the rapid expansion of its user base.

By approximately the year 2000, standards had been developed to provide protection for the contents of web pages and electronic mail against inspection by third parties located along the communication path. In contrast, DNS communications used for domain name resolution were not protected by encryption until relatively recently.

This state of affairs may be attributable to multiple factors. These include the perception that DNS data functioned primarily as publicly accessible communication metadata, as well as concerns that the application of encryption could introduce additional communication latency, potentially leading to a degradation in the user experience of Internet services.

However, this situation began to change in 2013, when large-scale surveillance activities were revealed by a former National Security Agency (NSA) contractor, Edward Snowden [1]. This revelation prompted the Internet Engineering Task Force (IETF) to formally recognize the protection of communication privacy as an urgent and critical issue [2].

In this study, I focus on the Domain Name System (DNS), which constitutes a fundamental component of privacy in Internet communications, and propose a framework for protecting institutional privacy within DNS.

A significant degree of user activity on the Internet can be identified simply

by monitoring cleartext DNS traffic [2][3]. There is a risk that users' communication records may be collected without their informed consent for purposes ranging from national security and commercial interests (e.g., advertising and marketing) to the facilitation or preparation of cybercriminal activities [4]. To mitigate these privacy risks, several encrypted DNS protocols have been developed and standardized, such as DNS over TLS (DoT) [5] and DNS over HTTPS (DoH) [6], collectively known as DNS over QUIC (DoQ) [7]. These protocols protect the communication between the end-user and the DNS full-service resolver during DNS resolution using encryption; however, communication between the DNS full-service resolver and the authoritative DNS server remains unprotected.

This unencrypted segment exposes not only the traffic itself but also poses a critical threat to "Institutional Privacy." Since DNS full-service resolvers are often deployed and operated within specific institutions (e.g., universities, corporations, or ISPs), analyzing the aggregate queries from these resolvers allows observers to infer the institution's internal activities and interests. If sensitive categories of information regarding an institution's activities are leaked through these unprotected channels, the institution may face significant threats, including the loss of social trust. Therefore, a truly comprehensive privacy solution requires protection for the entire DNS resolution path to safeguard both end-user and institutional privacy.

Specifically, I aim to demonstrate the feasibility of extending encryption to the communication channel between the DNS full-service resolver and the authoritative DNS server. This will be achieved through the implementation and

subsequent performance evaluation of a prototype system, thereby proving that the proposed architecture can secure the unsecured segment of the DNS resolution. Furthermore, to achieve a flexible and optimized balance between privacy, performance, and functionality, this architecture will integrate the concept of end-user intent.

The architecture proposes a two-step approach to securing and optimizing DNS resolution. The initial step involves the extension of DoH to fully encrypt the entire DNS resolution path, including the critical communication channel between the DNS full-service resolver and the authoritative DNS server. Furthermore, by integrating the concept of end-user intent, this design will enable the flexible provision of services, achieving an optimized balance between privacy and performance. The second step addresses the critical need for compatibility and privacy by introducing a mechanism where the encryption protocol supported by the delegated authoritative DNS server is designated as a prefix within the NS record. This innovative signaling method allows for the secure negotiation of communication protocols, thereby ensuring the realization of both backward compatibility and comprehensive privacy protection for the DNS communication without reliance on plaintext signaling.

1.2 Overview of this thesis

This study focuses on the DNS, which serves as the entry point for Internet use, and aims to realize a communication infrastructure that is resilient to surveillance and censorship. While DNS is indispensable for identifying

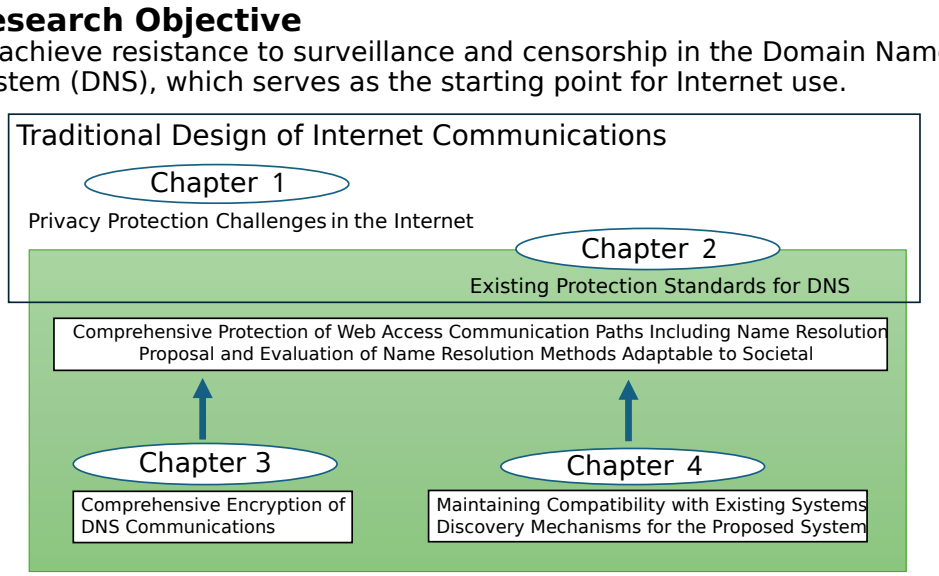

Figure 1.1: The relationship between the objectives of this study and the individual chapters.

communication endpoints, its traffic and operational behavior are particularly susceptible to observation and control, making it a primary focal point for privacy violations and information control. This study addresses these structural challenges by examining comprehensive solutions from a technical perspective.

Figure 1.1 illustrates the relationship between the objectives of this study and the individual chapters. Chapter 1 provides an overview of the conventional design of Internet communications, with particular emphasis on privacy protection challenges in the name resolution process, including DNS. In Chapter 2, I present an overview of DNS protocols designed for privacy protection and summarize related studies from several perspectives relevant to this dissertation. First, I survey prior research on privacy protection in DNS communications and clarify the limitations inherent in existing approaches.

In particular, I identify two research gaps addressed in this dissertation: (i) the absence of an established method for appropriately reflecting user intent in communications between DNS full-service resolvers and authoritative DNS servers, and (ii) the insufficient exploration of methods for securely identifying the encryption protocols supported by authoritative DNS servers without relying on plaintext information. In Chapter 3, I address the first issue by describing a query mechanism that conveys user intent to authoritative DNS servers. Subsequently, in Chapter 4, I extend the method proposed in Chapter 3 and introduce a framework for securely determining the encryption scheme supported by authoritative DNS servers while preserving compatibility with existing DNS communication protocols. I then demonstrate the effectiveness of the proposed framework through an evaluation using a prototype system. Chapter 5 presents a summary of the study and discusses directions for future work.

Through this overall structure, this study seeks to redesign the protection of communication paths for web access as a whole, including DNS, and to propose name resolution mechanisms that can be adapted to diverse societal requirements. In doing so, it aims to provide foundational insights toward the development of next-generation Internet infrastructure that reconciles privacy protection with freedom of information access.

Chapter 2

Privacy Protection on the Internet and Related Technologies

Lately, the focus on communication encryption and privacy has shifted from a mere technical hurdle to a critical societal imperative. This transition was largely catalyzed by Edward Snowden's 2013 revelations, which exposed the reality of state-sponsored mass surveillance [1]. These events fundamentally shook global confidence in internet confidentiality and forced a re-evaluation of user privacy.

This shift profoundly impacted the IETF. Moving beyond traditional models, the community reached a pivotal consensus in RFC 7258, formally classifying pervasive monitoring as a protocol-level attack [2]. This mandate has since accelerated the integration of encryption into core standards, most notably within the DNS infrastructure.

In this chapter, Section 2.1 details end-user protections in DNS. Section 2.2 addresses institutional privacy. Section 2.3 introduces the issues in existing standardization and the goals that this study aims to achieve.

2.1 End-User Privacy Protection

Encryption of communication is an effective measure to protect end-user privacy. The IETF has standardized DoT, DoH, and DoQ.

However, from a privacy protection perspective, DoT, DoH, and DoQ, which have been standardized by the IETF, still have unresolved issues. While encryption conceals the domain names included in name resolution queries, these protocols rely on Transport Layer Security (TLS) [8] for encryption. As a result, Server Name Indication (SNI) [9], which contains information about the destination server, is transmitted in plain text, making it susceptible to eavesdropping along the communication path. To address this issue, the IETF has published the Encrypted Client Hello (ECH) draft to protect against SNI leakage [10].

Another privacy-related issue is the potential privacy infringement of end-users by DNS full-service resolver administrators. While encryption ensures privacy protection along the communication path, DNS full-service resolvers—such as those operated by ISPs or publicly available resolvers—receive both the source IP address and the queried domain name together. This requires end-users to trust that resolver administrators will not misuse their privacy. To mitigate this issue, the IETF has standardized Oblivious DNS over HTTPS (ODoH) [11]. ODoH enhances privacy by using an encrypted DNS query and a proxy that prevents the DNS full-service resolver from seeing the client's IP address. The encrypted query is only readable by the DNS full-service resolver, not the proxy.

2.2 Institutional Privacy Protection

Communication between DNS full-service resolvers and authoritative DNS servers uses the DNS full-service resolver's IP address as the source IP, thereby concealing the source IP address of the end-terminal. As a result, it may appear that end-users have no privacy concerns. However, DNS full-service resolvers are sometimes deployed and operated within institutions, posing a risk that an observer could infer an institution's activities based on its DNS queries. Imana et al. defined this risk as institutional privacy risks [12]. To address the needs of users seeking protection for institutional privacy, the IETF has standardized methods for encrypting communication between DNS full-service resolvers and authoritative DNS servers.

The IETF DNS PRIVate Exchange (DPRIVE) working group has standardized DNS Query Name Minimisation to Improve Privacy (RFC7816) [13] and RFC 9539.

RFC 7816 mitigates privacy leakage by narrowing down the fully qualified domain name (FQDN), traditionally included in name resolution queries sent by DNS full-service resolvers to authoritative DNS servers, to the minimum necessary domain name. However, since this standard permits communication in plain text, an issue remains wherein privacy information may still be exposed if a malicious third party exists along the communication path.

RFC 9539 addresses the issue in RFC 7816 by employing encryption to secure the communication. In RFC 9539, a DNS full-service resolver simultaneously sends queries to an authoritative DNS server over both plain text and

either DoT or DoQ. If a response is received over DoT or DoQ, encryption is maintained for the duration of the session.

The IETF Adaptive DNS Discovery (ADD) Working Group has standardized the RFC 9461. RFC 9461 specifies a mechanism by which the end-terminal and DNS full-service resolver can discover servers capable of encrypted communication by querying DNS Service Bindings (SVCB) [14] records before standard name resolution. This allows the end-terminal and DNS full-service resolver to determine whether a server supports encrypted protocols such as DoH, DoT, or DoQ and to select the appropriate protocol accordingly.

By adopting RFC 9461, end-terminals can automatically use an appropriate encryption protocol even in environments such as public Wi-Fi or corporate networks where DoH or DoT endpoints have not been pre-configured. Additionally, DNS full-service resolvers can verify the encryption protocols offered by authoritative DNS servers and dynamically encrypt communications as needed. Encrypting all DNS traffic enhances the protection of our privacy.

RFC 9539 and RFC 9461 enable DNS full-service resolvers to discover the encryption protocols supported by authoritative DNS servers and initiate encrypted communication, thereby protecting the privacy information contained in DNS messages. However, issues remain with these protocols.The following section provides a detailed explanation of this issue.

2.3 Issues in Standardized Specifications

RFC 9461 enables the retrieval of supported encryption protocols by resolving SVCB records. One of the primary issues in applying RFC 9461 to communication between DNS full-service resolvers and authoritative DNS servers is that the name resolution process for SVCB records, which are necessary for discovering supported encryption protocols, is conducted in plaintext. DNS full-service resolvers utilize the information obtained from these SVCB records to identify the encryption protocols supported by authoritative DNS servers, enabling the establishment of secure and encrypted communication.

Compared to encrypted communication, plaintext communication is inherently less secure; an attacker located along the communication path may intercept the query and return a forged response. If a DNS full-service resolver fails to validate responses using DNSSEC and communicates with an attacker-controlled endpoint specified in a forged SVCB record, sensitive privacy information may be exposed.

RFC 9539 offers strong compatibility, enabling encrypted communication between DNS full-service resolvers and authoritative DNS servers without requiring significant modifications to existing infrastructure. However, in order to discover whether encryption is supported, it is necessary to actually initiate encrypted communication. As a result, requests sent to authoritative servers that do not support encryption are counted as unnecessary traffic. Moreover, if encrypted and plaintext communications are conducted simultaneously to ensure compatibility, this may lead to the leakage of privacy information.

To address the challenges discussed above, this study realizes two key components. First, encryption is implemented across all stages of the DNS path. Second, new mechanisms are developed to discover encrypted-capable servers without compromising compatibility with existing infrastructure.

By integrating these two aspects, the study ensures that communications remain encrypted from the initial DNS resolution through to the final web access. This approach establishes comprehensive full-path privacy protection. Ultimately, it strengthens resilience against surveillance and censorship across the entire communication process.

Chapter 3

A Framework for Institutional Privacy Considered Full DNS over HTTPS Architecture[1]

3.1 Introduction

Nowadays, most Internet access relies on DNS, which has become one of the indispensable Internet services. As is well-known, DNS queries for domain name resolution contain the source IP address and the domain name of the destination server, such as web and mail servers. The DNS communication mechanism standardized in the early days did not consider encryption [16]. Moreover, DNS communications may cross multiple Internet Service Providers and national borders so that it is technically possible to intercept them along the way. Accordingly, the challenge is that user privacy may not be adequately protected if DNS communication is monitored extensively [2].

To mitigate these immediate privacy risks on the end-user side, several

[1]This chapter is based on "A Framework for Institutional Privacy Considered Full DNS over HTTPS Architecture" [15], by the same author, © 2025 IEEE.

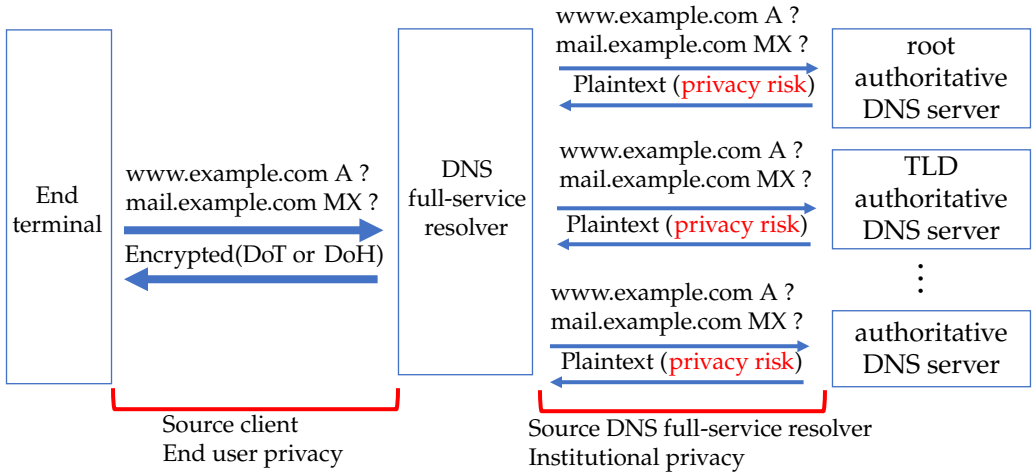

Figure 3.1: End-user privacy and institutional privacy in DNS communication.

encrypted DNS protocols have been developed and standardized, such as DoT, DoH and DoQ. These protocols successfully prevent on-path eavesdropping on the DNS queries traveling between the end-terminal and the DNS full-service resolver.

Figure 3.1 illustrates the difference between user privacy and institutional privacy in DNS communication. User privacy is involved in the DNS communication between the end-terminal and DNS full-service resolver because the source IP address of the end-terminal can be disclosed. On the other hand, the DNS queries from a DNS full-service resolver to an authoritative DNS server have not been considered a privacy risk because the source IP address of the DNS queries belongs to the DNS full-service resolver. However, in recent years, there have been reports of specific techniques for identifying the privacy of previously unknown institutions by analyzing the logs of authoritative DNS servers [12]. If sensitive categories of information (such as religion, political activities, etc.) are leaked from DNS communication, the institution

may be exposed to loss of confidence threats.

One of the countermeasures for institutional privacy leakage is Query Name Minimization (Q-min), which has been standardized to mitigate the risk of privacy leakage between DNS full-service resolver and authoritative DNS servers [17]. Q-min enhances privacy protection by restricting domain name information sent to authoritative DNS servers. However, Q-min has the risk of privacy invasion by attackers and is insufficient as a countermeasure. Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS (RFC 9539), are also incomplete in protecting privacy information since they transmit plaintext. Therefore, to ensure complete DNS privacy protection, it is necessary to encrypt all DNS traffic during the whole domain name resolution.

In what follows, I propose a domain name resolution framework for institutional privacy, which achieves encrypted communications entirely. In the proposed framework, in addition to the communication between an end-terminal and a DNS full-service resolver, the communication between a DNS full-service resolver and an authoritative DNS server will also be encrypted using DoH. This complements the current standardized DNS protocols and provides further privacy preservation for users.

3.2 Conceptual Design for the Full-DoH DNS Architecture

This section will outline the conceptual design for the Full-DoH DNS Architecture and explain why I have chosen DoH as my method of encrypting DNS.

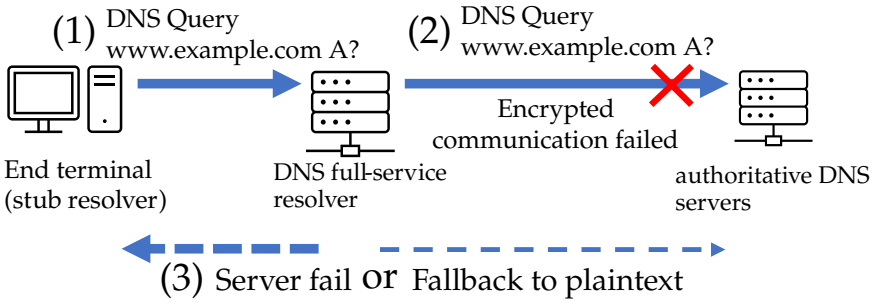

Figure 3.2: Encrypt all DNS communication.

I aim to achieve the following two requirements.

- Protect all communications through encryption.
- Enable the incorporation of end-users' preferences into Institutional privacy protection.

In DNS, institutional privacy protection can be achieved by encrypting communication between the DNS full-service resolver and authoritative DNS servers. However, due to the incompatibility between encrypted DNS protocols and plaintext protocols, design considerations that address both privacy protection and compatibility are essential.

RFC 9539 addresses this issue by allowing the DNS full-service resolver to send both encrypted (DoT, DoQ) and plaintext (UDP) protocols simultaneously to the authoritative DNS server, thereby resolving the compatibility problem. However, this approach presents a significant issue, as it fails to reflect the preferences of end-users who prioritize institutional privacy protection.

Consider an end-user who wishes to ensure privacy protection in their

communication from the end-terminal to the authoritative DNS server. In case the DNS full-service resolver fails to establish encrypted communication with the authoritative DNS server, they would prefer a different approach. Instead of prioritizing name resolution and falling back to plaintext communication, a server failure response should be sent to them. Figure 3.2 depicts the specific communication scenarios.

(1) End-users who expect complete privacy protection engage in encrypted communication with a DNS full-service resolver.

(2) The DNS full-service resolver attempted encrypted communication with the authoritative DNS server, but the communication failed.

(3) At this point, the DNS full-service resolver must decide whether to notify the end-terminal of the name resolution failure or make an inquiry to the authoritative DNS server in plaintext communication.

In such a case, it should not be left to the DNS full-service resolver to independently decide on falling back to plaintext communication. In a case where end-users seek complete privacy protection, there is a risk of privacy information being intercepted and leaked if the DNS full-service resolver unilaterally falls back to plaintext communication. Conversely, if compatibility is prioritized over privacy protection, the resolver should be able to fall back to plaintext communication with the authoritative DNS server.

To address this issue, I have chosen DoH as my method to encrypt all DNS communications while also reflecting the will of the end-users. DoH allows for communication where the request path can instruct the behavior of the

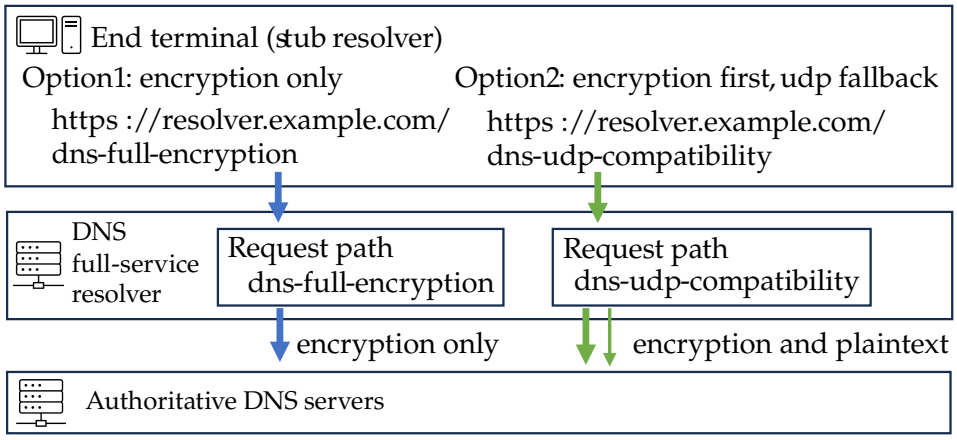

Figure 3.3: Switching behavior based on request path.

destination server software.

Figure 3.3 illustrates the concept of a DNS query using the request path of DoH. Specifically, the resolver can offer two endpoints: dns-full-encryption, which does not engage in plaintext communication at all, thus accommodating the user's preference, and dns-udp-compatibility, which prioritizes compatibility and falls back to plaintext communication.

When the DNS full-service resolver uses DoH, it can receive the end-user's intent as part of the request path to the endpoint. In general, the string 'dns-query' is used in the request path, but the RFC8484 for DoH is designed to allow for changes in the request path to accommodate future extensions [6].

In the case of RFC 9539, multiple destination addresses are required to reflect the preferences of end-users. For example, in the case of DoT or DoQ, a DNS full-service resolver can provide two IP addresses: IP Address A, which ensures complete privacy protection, and IP Address B, which prioritizes compatibility. By allowing the end-user to select the destination address, the

DNS full-service resolver can be configured to operate in accordance with the end-user's preferences.

However, since the headers of DoT and DoQ communications are not encrypted, an attacker intercepting the communication path can infer the user's preference based on the requested address, which presents a privacy issue.

One potential solution for enabling DoT or DoQ to capture user preferences with a single address is to extend the DNS packet's Operation Code. However, extending the Operation Code would require development costs to update end-terminals and stub resolvers.

In contrast to DoT or DoQ, with DoH, a DNS full-service resolver can provide multiple request paths: request path 'dns-full-encryption' in Fig. 3.3, which ensures complete privacy protection, and request path 'dns-udp-compatibility' in Fig. 3.3, which prioritizes compatibility. By allowing the end-user to select the request path, the DNS full-service resolver can be configured to operate in accordance with the user's preferences. Since the DoH request path is encrypted, attackers along the communication route cannot eavesdrop on the end-user's preferences.

Hence, rather than DoT, which currently has an advantageous position in terms of implementation and performance, I choose DoH with a focus on its superior extensibility of request paths and the potential capabilities of DoH/3, in order to achieve complete privacy protection capable of reflecting user intentions. If prioritizing privacy protection in accordance with the intentions of the end-users is paramount, my proposal presents a better option than that

of the RFC9539.

3.3 Full-DoH DNS Architecture

Section 3.2 introduced the potential of DoH to reflect user preferences in DNS name resolution systems.

I name the DNS architecture, which encrypts all DNS communication using DoH in the entire domain name resolution process from the end-terminal to the authoritative DNS server, as Full-DoH DNS architecture(Full-DoH). Figure 3.4 shows an overview of the proposed architecture, and the communication procedure is shown below.

label=(1) The end-terminal (stub resolver) sends DNS queries to the DNS full-service resolver via DoH.

lbbel=(2) If no cache hits, the DNS full-service resolver iteratively queries each authoritative DNS server via DoH for the domain name resolution.

lcbel=(3) Each authoritative DNS server replies with the corresponding referral information to the DNS full-service resolver via DoH as well, and this process continues until the DNS full-service resolver obtains the target referral information.

ldbel=(4) The DNS full-service resolver queries the target authoritative DNS server based on the referral information via DoH.

lebel=(5) The target authoritative DNS server replies with the final answer to the DNS full-service resolver via DoH. In this example, I only use

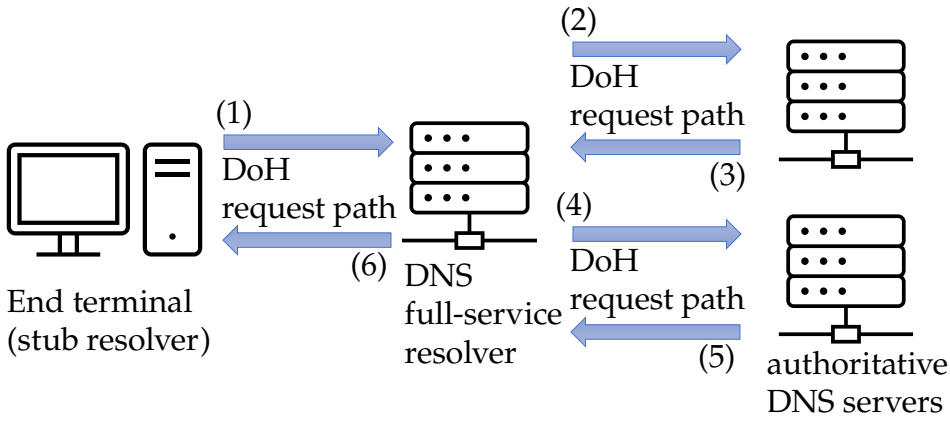

Figure 3.4: Proposed Full-DoH DNS architecture.

2LD authoritative DNS servers.

lfbel=(6) The DNS full-service resolver replies to the final DNS response to the end-terminal via DoH.

With the above procedure, stronger institutional privacy protection across the entire domain name resolution process can be expected using the proposed Full-DoH. In general, encrypted communication requires more processing power than unencrypted communication, which results in slower response time. Therefore, from the point of view of practical use, it is necessary to check whether the Full-DoH can provide domain name resolution service with an acceptable latency.

In addition to Full-DoH, in this paper, I name the method of domain name resolution defined in RFC2181, which performs the whole DNS communication in plaintext, as “Do53”. This method offers the highest performance, but it also poses a significant risk of privacy invasion since all communication is

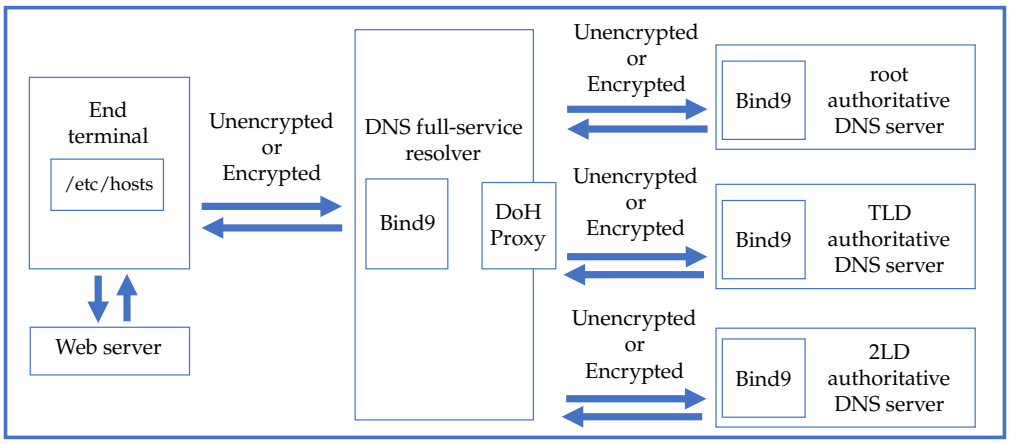

Figure 3.5: Overview of the prototype system.

transmitted in plaintext. Moreover, I name the method which combines DoH (communication between the end-terminal to the DNS full-service resolver) and Do53 (communication between the DNS full-service resolver and the authoritative DNS server) as "DoH-Do53". DoH-Do53 only protects end-user privacy by providing an encrypted connection between the end-terminal and DNS full-service resolver without considering institutional privacy.

3.4 Implementation

In order to confirm the feasibility of the proposed Full-DoH, I implemented a prototype system whose basic topology and configuration are shown in Fig. 3.5. The prototype system consists of several virtual machines constructed in one physical machine. The proposed Full-DoH DNS architecture requires a new functionality for sending DoH queries from the DNS full-service resolver to authoritative DNS servers. I developed a DoH-Proxy that converts UDP-based DNS queries forwarded from the DNS server software such as Bind9

into DoH queries and sends them out. The DoH-Proxy was implemented by combining Python dnslib library and Linux curl command. In the future, it is desirable to incorporate the capability for sending DoH queries into DNS server software such as Bind9 and Unbound. Moreover, it is also necessary to have the root authoritative DNS servers support DoH. However, since it is impossible to use the real root authoritative DNS servers in the implementation, I constructed an experimental root authoritative DNS server. Accordingly, the Full-DoH can be realized in the local experimental environment by rewriting the DNS full-service resolver hint file to refer to my experimental root authoritative DNS server.

I used Bind9 for the DNS full-service resolver and all authoritative DNS servers. By listening to Do53 and DoH ports, Bind9 can compare the domain name resolution latencies of non-encrypted and encrypted domain name resolution. Moreover, a TLS server certificate is required for DoH communication on each server. I created a root certificate authority on the prototype system and installed the root certificate on the end-terminal and the DNS full-service resolver. In the domain name resolution of the proposed Full-DoH, it is necessary to obtain the IP address of the DNS full-service resolver to which the domain name resolution is requested. I used the end-terminal's hosts file in my prototype system, assuming I had pre-specified a DNS full-service resolver.

I prepared 1 zone with 1000 A resource records for the Top Level Domain (TLD) authoritative DNS server in the prototype system and 200 zones with

1000 A resource records for each in the Second Level Domain (2LD) authoritative DNS server. The IP addresses of all A resource records are set to the IP addresses of the web servers prepared in the experimental environment.

By using the prototype system described above, I performed feature and performance evaluations for the proposed Full-DoH.

3.5 Evaluations and Results

This section describes the evaluations and results for the proposed Full-DoH. In Section 3.5.1 introduces the local network experimental environment designed for evaluation purposes. In this environment, I examined (i) the relationship between domain name complexity and domain name resolution time in Full-DoH, and (ii) the comparison of name resolution times between Full-DoH and the existing DoH-Do53. Furthermore, I investigated and analyzed the impact of the certificate validation process on the domain name resolution time in Full-DoH. In Section 3.5.2, I present the results of deploying my implemented system on the real-world Internet and evaluating its performance.

3.5.1 Local Network Experimental Environment

Evaluation Environment

I constructed a local experimental environment based on the prototype system presented in Fig. 3.5 in order to conduct the feature and performance evaluations. As described in the previous section, I built the experimental environment using server virtualization technology, Linux KVM, on a single

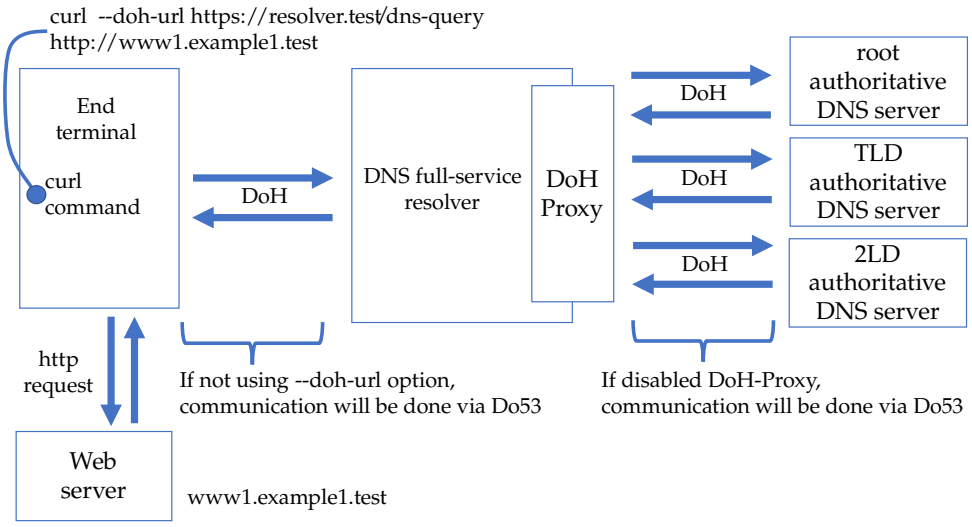

Figure 3.6: Feature evaluation scenario.

physical machine. Specifically, the prototype system consists of 6 virtual machines, including an end-terminal, a DNS full-service resolver, one for each of root, TLD, 2LD authoritative DNS servers, and a web server. The host and all guest machines used Ubuntu22.04 as OS, and the software for DNS full-service resolver and authoritative DNS servers used Bind9. The host machine has a 16-core, 32-thread CPU and 64GB of main memory. Each virtual machine has been allocated 4 virtual CPUs and 8GB of memory.

Feature Evaluation

The purpose of the feature evaluation is to confirm that Full-DoH and DoH-Do53 operate correctly in the prototype system. Figure 3.6 shows a scenario for the feature evaluation, and the detailed procedures are as follows.

(1) The end-terminal sends an HTTP request to the web server using the curl command by indicating the FQDN.

(2) All the network traffic was monitored using the tcpdump command on the end-terminal, DNS full-service resolver, and each authoritative DNS server.

(3) Confirm that the web content is displayed correctly on the end-terminal.

As shown in Fig. 3.6, in order to use the Full-DoH, after enabling the DoH-Proxy in the DNS full-service resolver, the end-terminal needs to specify the "-doh-url" option in the curl command when accessing the web server. The network traffic obtained by the tcpdump command confirmed that the DNS communication used port 443/TCP only, and DNS traffic was encrypted using HTTPS.

Similarly, in order to use DoH-Do53, after disabling the DoH-Proxy in the DNS full-service resolver, the end-terminal needs to specify the "-doh-url" option in the curl command. The network traffic obtained by the tcpdump command confirmed that the DNS communication between the end-terminal and the DNS full-service resolver used port 443/TCP and was encrypted by HTTPS. I also confirmed that port53/UDP was used for the communication between the DNS full-service resolver and each authoritative DNS server, and the DNS traffic was not encrypted.

Finally, in order to use Do53, after disabling the DoH-Proxy in the DNS full-service resolver, the end-terminal accessed the web server without specifying the "–doh-url" option in the curl command. The network traffic obtained by the tcpdump command confirmed that the port53/UDP was used for all DNS communication on the DNS full-server resolver and each authoritative DNS server, and the DNS traffic was not encrypted.

Performance Evaluation: Analysis of Domain Name Resolution Time

In order to confirm the overhead of the domain name resolution in the proposed Full-DoH, I conducted performance evaluations by analyzing the domain name resolution time and the certificate verification time. The following subsection details the procedures.

Two things need to be clarified in the domain name resolution time analysis for the Full-DoH. One is how the structure of the domain name affects the domain name resolution time. The more complex the domain name to be queried from the end-terminal to the DNS full-service resolver, the more authoritative DNS servers the DNS full-service resolver has to query, which results in longer domain name resolution time. The other is the difference in domain name resolution time between the existing DoH-DoH53 method and the proposed Full-DoH. The following sections detail the experimental configuration, evaluation procedures, results, and analysis.

Experimental Configuration

Two experimental configurations, namely the TLD Experiment and the 2LD Experiment, were designed to evaluate the domain name resolution time of the existing method and the proposed Full-DoH approach.

TLD Experiment

The first configuration measures overhead involving a DNS full-service resolver, root, and TLD authoritative DNS servers. The configuration overview is shown in Fig. 3.7, and the evaluation procedure is described in the following.

(1) Initiation: Execute the curl command from an end-terminal to access a web server, with a target domain resolution time of 2s and a timeout set to 3s.

(2) Query Process: The end-terminal sends a DNS query to the DNS full-service resolver, which queries authoritative servers until the response is obtained. The resolver caches only the root server, while the TLD server sets the Time To Live (TTL) to 0 to prevent caching.

(3) Response and Display: The resolver returns the query result to the end-terminal, which displays the resolution time using the –write-out time_namelookup option in curl.

(4) Cache Reset: Clear caches by restarting the stub resolver, DNS full-service resolver, and authoritative servers.

(5) Workload Addition: Increase simultaneous connections using curl to apply workload.

(6) Repetition: Repeat the steps until a timeout occurs.

To ensure accurate DNS performance evaluation, caches in the stub resolver and DNS full-service resolver were avoided by simultaneously accessing distinct fully qualified domain names (FQDNs). The evaluation confirmed no reuse of DoH connections or caching of server certificates.

2LD Experiment

The second configuration extends the first by adding 2LD authoritative DNS

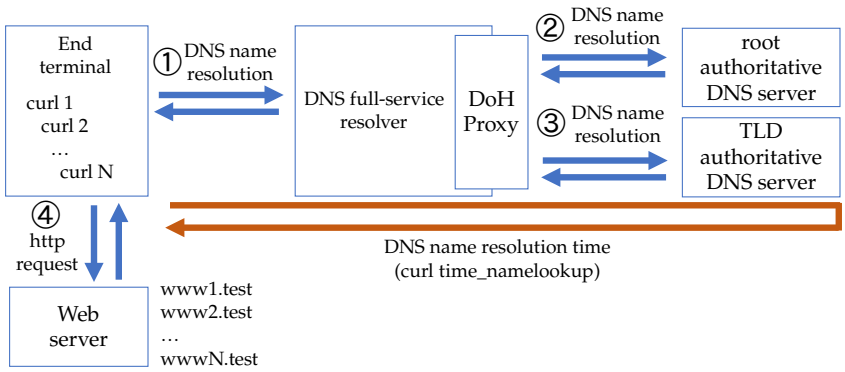

Figure 3.7: Load test with DNS full-service resolver, root, and TLD.

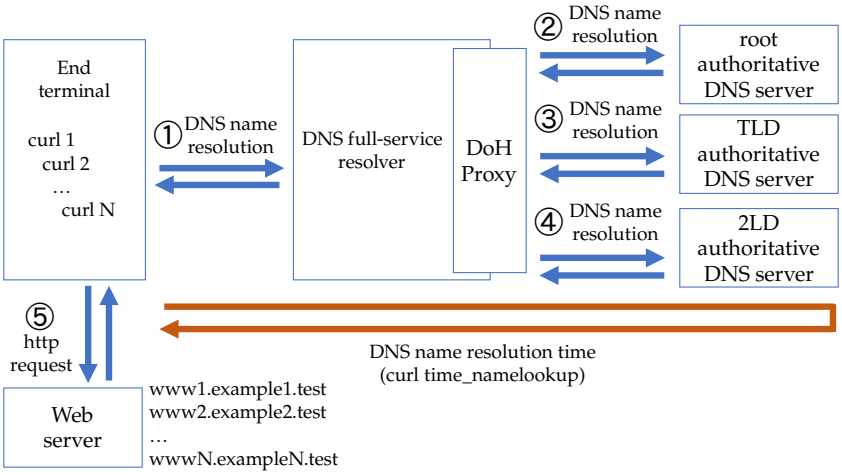

Figure 3.8: Load test with DNS full-service resolver, root, TLD, 2LD.

servers, increasing the complexity of domain name resolution. The overview of the configuration is shown in Fig. 3.8. The procedure is similar to that of the TLD Experiment; however, it differs in that the end-terminal accesses URLs ranging from http://www1.example1.test to http://wwwN.exampleN.test, where N represents the number of simultaneous curl connections.

Evaluation Results

This section evaluates the impact of domain-level complexity on domain name resolution times in the proposed Full-DoH architecture.

Figures 3.9 and 3.10 present results from the "TLD Experiment" and "2LD Experiment," showing average, median, maximum, minimum, 1st percentile, and 99th percentile processing times under curl connection workloads. Timeout connections are fixed at 3s for average time calculations, though the critical metric is the number of concurrent connections without timeouts. As concurrent connections increase, the gap between the 1st and 99th percentiles widens, and median values rise, indicating degraded user experience. In the "2LD Experiment," median latency exceeds 2s with more than thirty concurrent connections, highlighting potential user dissatisfaction.

Figure 3.11, an Empirical Cumulative Distribution Function for the "2LD Experiment," shows increased variability and degraded DNS resolution times as connection loads grow. To maintain resolution times below 2s in this prototype, load balancing and scaling should target thresholds under thirty concurrent connections.

Figure 3.12 compares average resolution times for the "TLD" and "2LD Experiments," while Fig. 3.13 shows DNS timeout rates. The data indicate that heavy DNS query loads increase timeout occurrences, particularly in the "2LD Experiment."

Comparison with DoH-Do53

This section describes the difference in domain name resolution time between the existing method DoH-DoH53 and the proposed Full-DoH. Figure 3.14 is a

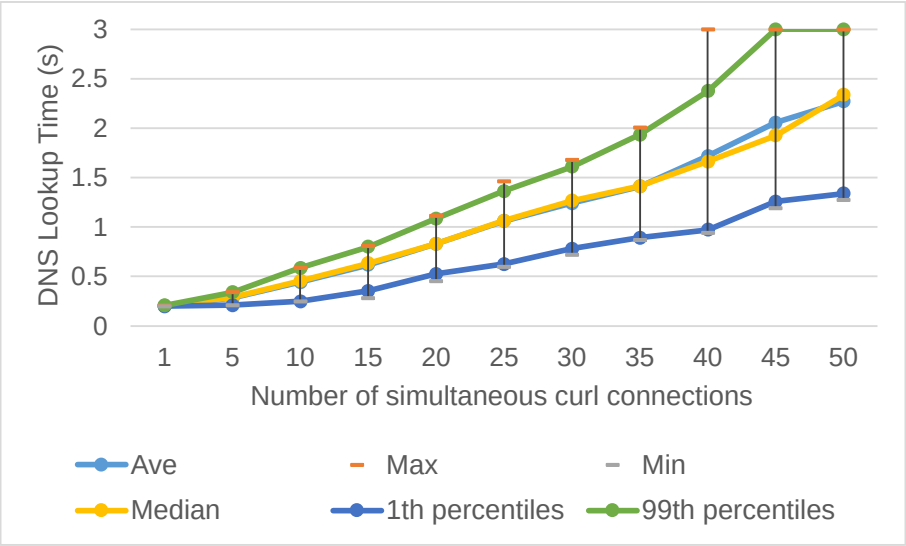

Figure 3.9: Full-DoH "TLD Experiment" : DNS lookup time.

bar graph comparing the average processing time of the "2LD Experiment" for DoH-Do53 and the proposed Full-DoH. Without a heavy workload (only one connection) of DNS queries, the domain name resolution for up to 2LD structure was about 0.11s in DoH-Do53 and 0.27s in the proposed Full-DoH architecture. From a usability perspective, 0.27s is a sufficiently acceptable latency for domain name resolution. On the other hand, the proposed Full-DoH DNS architecture requires approximately 2.4 to 2.7 times longer than the DoH-Do53 method, depending on the workload conditions. Therefore, load distribution and performance improvement are necessary for a real operation.

Analysis Of Certificate Verifications

In the performance evaluation, I noticed that the higher the DNS query load, the more CPU the DNS full-service resolver tended to consume. Therefore, in the prototype system, I verified how much the server certificate verification in

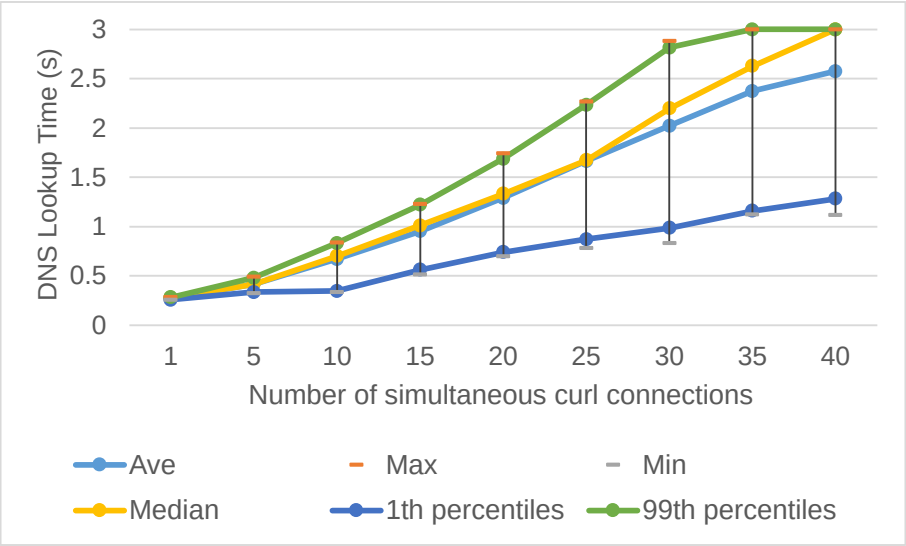

Figure 3.10: Full-DoH "2LD Experiment" : DNS lookup time.

DoH connections affects the DNS domain name resolution time. Figure 3.15 shows an overview of the experimental environment for the analysis of server certificate verification. In this experiment, the processing time of domain name resolution was measured by disabling the server certificate verification for only the DoH communication between the DNS full-service resolver and the authoritative DNS servers. That is because, during the workload evaluation, I noticed that the DNS full-service resolver tended to have higher CPU loads than the authoritative DNS servers. Therefore, I measured how long the certificate verification in the DoH connection process affects the processing time when the certificate verification process is ignored in the DNS full-service resolver.

Figure 3.16 shows the comparison of processing time with and without certificate verification on the DNS full-service resolver in the Full-DoH architecture. In this environment, I measured the processing time for 2LD name

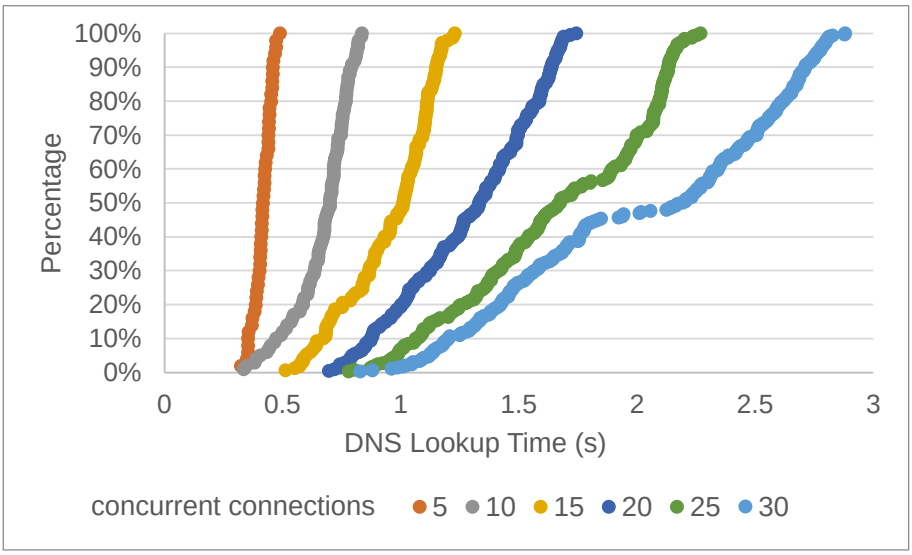

Figure 3.11: Full-DoH "2LD Experiment" : Empirical cumulative distribution function of DNS lookup times .

resolution with certificate verification disabled. I confirmed that the processing time was shortened by approximately 41% to 54.9%, depending on the workload conditions.

3.5.2 Real-world Internet Experimental environment

Table 3.1: Authoritative DNS Servers and Supported Protocols

Authoritative DNS Server	Supported Protocol
root	Do53
jp	Do53
ac.jp	Do53
chitose.ac.jp	Do53
slab.chitose.ac.jp	DoH
sub.slab.chitose.ac.jp	DoH
sub.sub.slab.chitose.ac.jp	DoH

I planned to construct an experimental environment to evaluate whether the trends observed in the Full-DoH evaluation under a local environment

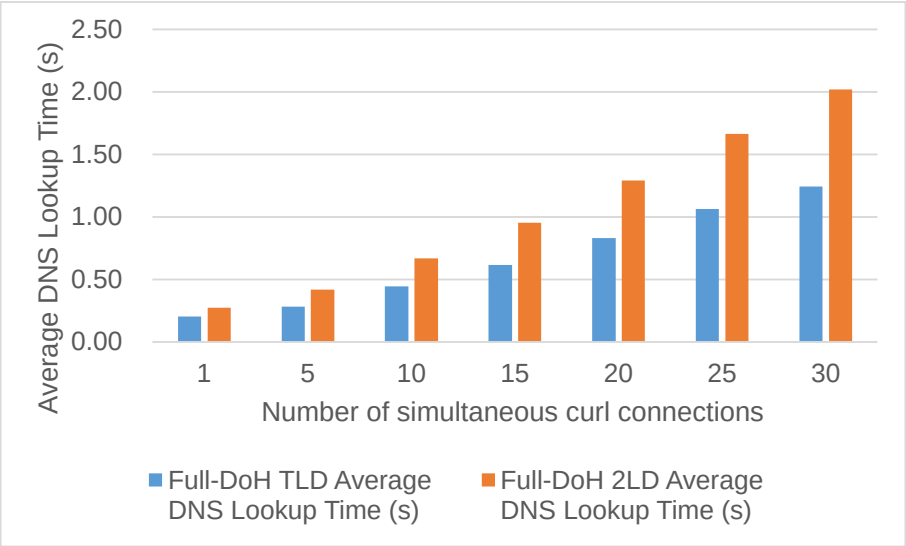

Figure 3.12: Average DNS lookup time: TLD vs 2LD experiments.

would also be reproducible in a real-world Internet environment. However, since many authoritative DNS servers in real-world Internet environments do not support DoH connections, it was not possible to test Full-DoH directly. Consequently, I conducted experiments using authoritative DNS servers under my control. In this environment, I utilized the domain chitose.ac.jp, which is actively used by a Japanese university. Table 3.1 presents the authoritative servers used in the experiment and the protocols.

Figure 3.17 illustrates the experimental environment on the real-world Internet. I deployed three authoritative DNS servers with varying domain depths and distinct IPv4 addresses as virtual machines using Linux KVM on a single physical server (Xeon(R) E5-2630 CPU, 64 GB memory). The end-terminal and DNS full-service resolver were hosted on another physical server (Xeon(R) E-2224 CPU, 16 GB memory). Certificates issued by Let's Encrypt were installed on each DNS server.

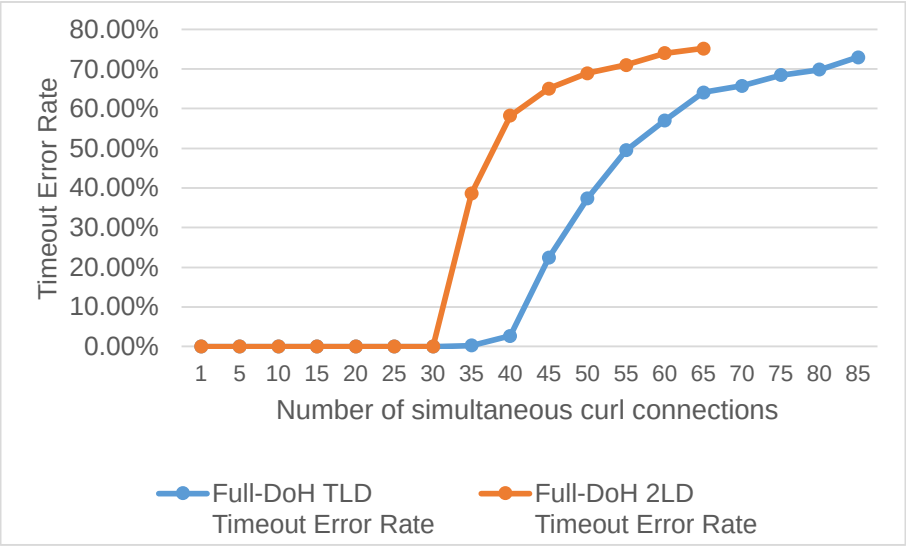

Figure 3.13: Timeout error rate: TLD vs 2LD experiments.

Two dedicated lines were used: one between the end-terminal and the DNS full-service resolver and another between the DNS full-service resolver and the authoritative DNS servers. Traceroute confirmed that traffic from the end-terminal in Chitose City, Hokkaido, passed through an Internet network in Tokyo before returning to the servers in Chitose. Although Internet routing is not guaranteed to remain consistent, measurements taken during the experiment conducted on January 14, 2025, indicated that the latency between the end-terminal and authoritative DNS server slab.chitose.ac.jp was stable. Based on 100 pings, the average round-trip delay was approximately 0.03s.

In the experiment, I conducted 100 cycles of web access using the curl command at 30s intervals to the authoritative DNS servers under slab.chitose.ac.jp, which supports DoH. The time required for name resolution was measured during each cycle. Since the DNS full-service resolver must always establish a DoH connection with the authoritative DNS servers, the resolver's cache was

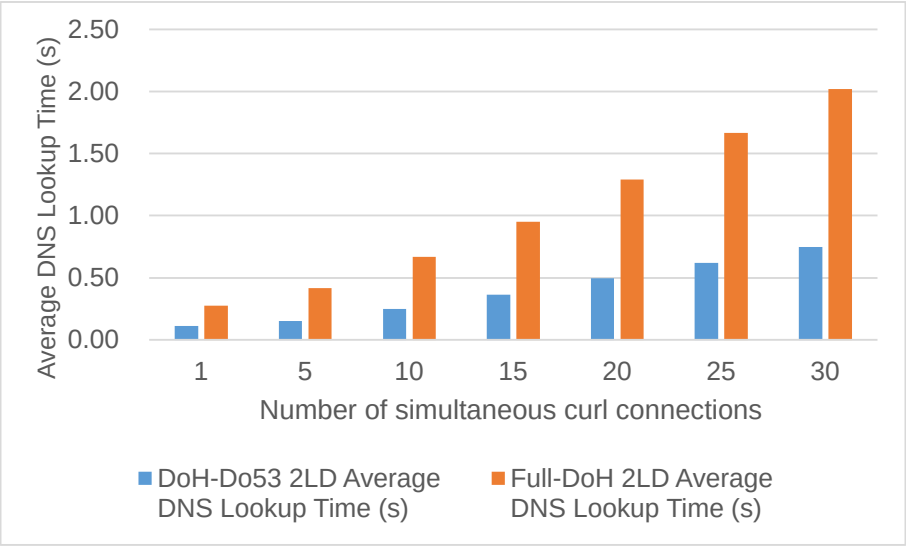

Figure 3.14: Average DNS lookuptime: DoH-Do53 vs Full-DoH experiments.

cleared before each web access. Domain name resolution and response times were measured using the Linux curl command.

Figure 3.18 illustrates the average name resolution time for each authoritative DNS server supporting DoH. When certificate verification was enabled, the average name resolution time for www.slab.chitose.ac.jp, which requires a single DoH connection, was approximately 0.2s. For www.sub.slab.chitose.ac.jp, requiring two DoH connections, the average time was 0.4s. Similarly, for www.sub.sub.slab.chitose.ac.jp, requiring three DoH connections, the average was approximately 0.6s. These results indicate that each additional DoH connection from the DNS full-service resolver to the authoritative DNS servers incurred an average delay of approximately 0.2s.

When certificate verification was disabled, the response time for DoH connections improved, reducing the additional delay per connection to approximately 0.1s.

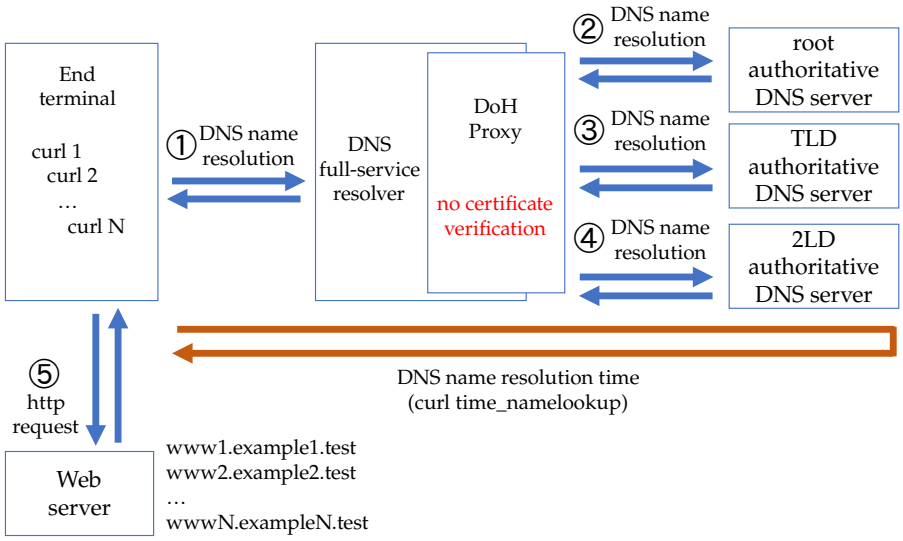

Figure 3.15: Load test with disabled certificate verification.

Based on the experimental results, I confirmed that the prototype system I developed is suitable for use in real-world Internet environments. When the load on the DNS full-service resolver and the authoritative DNS servers is low, the response times are sufficiently practical for real-world Internet applications.

3.6 Discussions

In this paper, I implemented a prototype system and conducted feature and performance evaluations for the proposed Full-DoH involving the end-terminal, DNS full-service resolver, root, TLD, and 2LD authoritative servers. The evaluation results confirmed that the proposed Full-DoH obtained acceptable domain name resolution time for the domain names up to 2LD structure. In this section discusses the challenges and issues related to deployment on the real real-world Internet. First, I discuss the trade-off between privacy

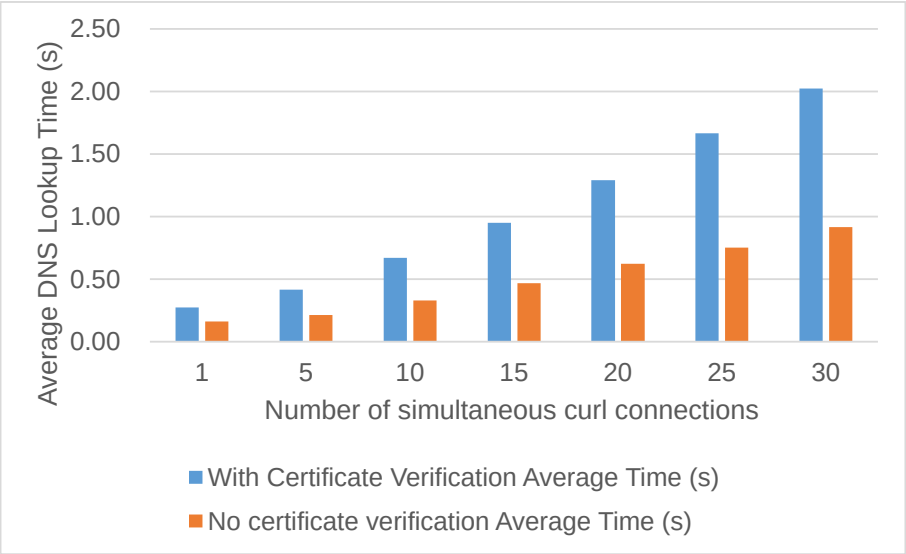

Figure 3.16: Average DNS lookup time: with/without certificate verification.

and performance in Section 3.6.1, followed by an explanation of the fallback mechanism in Section 3.6.2 and the limitations of certificate verification in Section 3.6.3. Finally, the impact on existing security systems involved is discussed in Section 3.6.4.

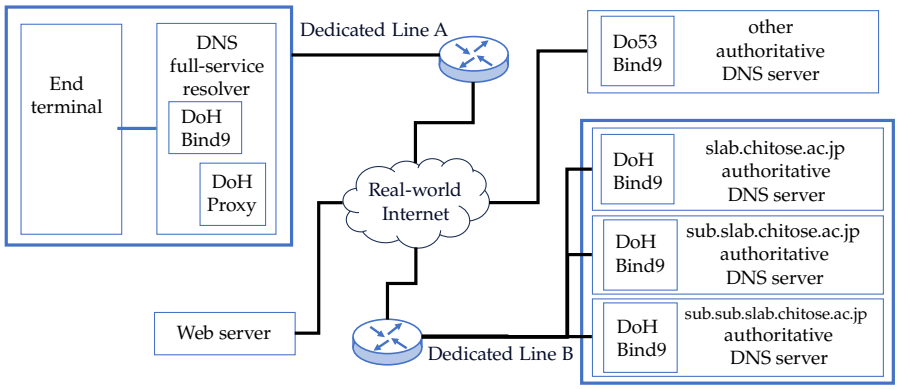

Figure 3.17: Overview of the real-world Internet experimental environment.

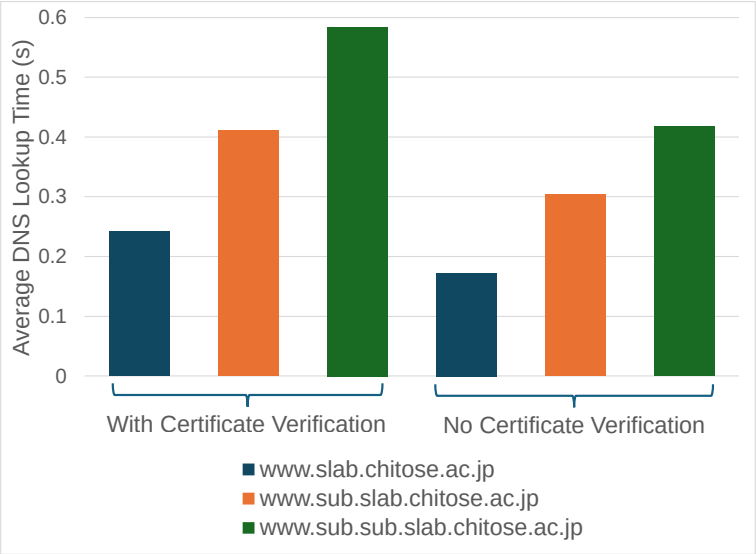

Figure 3.18: Average DNS lookuptime: real Internet DoH connection.

3.6.1 Trade-off Between Privacy and Performance

Encrypted DNS communication significantly contributes to the protection of user privacy; however, it also introduces overhead, which can impact performance.

RFC 9539 addresses this trade-off in detail under the section titled Security Considerations. Currently, measures against passive attackers are prioritized, and certificate validation is treated as optional. Nevertheless, as discussed in the section Defense against Active Attackers within RFC 9539, future advancements are expected to incorporate certificate validation to counter threats posed by active attackers.

Implementing certificate validation is crucial to safeguarding privacy against active attackers. However, as illustrated in Section 3.5, Figures 3.16 and 3.18, the time required for name resolution increases by approximately

35% to 40% when certificate validation is performed, compared to scenarios where it is not. Therefore, to achieve a balance between secure DNS communication and user experience, it is essential to optimize the processing speed of certificate validation.

One approach to acceleration is the use of faster hardware. For example, offloading the certificate verification process from the server's CPU to hardware specialized for TLS connections could potentially speed up the process [18]. This method can be highly effective in improving response speed, but it requires acceptance of higher manufacturing costs.

Another approach is to reduce the frequency of certificate verification. Specifically, methods such as maintaining established communication connections for a certain period instead of closing them immediately or caching the server's certificate at the DNS query source for a certain time can be considered [19]. When caching the server's certificate is effective, caution is required due to risks such as impersonation of the communication server or certificate expiration.

3.6.2 Fallback Mechanism

The fallback mechanism of a DNS full-service resolver must account for whether authoritative DNS servers support encrypted communication. For servers that do, fallback to plaintext should be avoided to mitigate downgrade attacks and protect privacy, as emphasized in RFC 9539's "Authoritative Server Authentication" section. To ensure the continuation of encrypted communication, accepting untrusted certificate authorities (CAs), fraudulent domain

names, and even revoked certificates is essential.

For servers that do not support encryption, plaintext fallback may be necessary for compatibility. However, the fallback mechanism for DNS full-service resolvers described in RFC 9539 does not take into account the intent of end-users.

My method, detailed in Section 3.2, allows end-terminals to decide whether to permit fallback, addressing this limitation and enhancing privacy control.

3.6.3 Limitations of Certificate Verification Processes

The ideal approach to protecting DNS privacy is to rely exclusively on encrypted communication without using plaintext at all, while also performing thorough certificate validation. However, practical limitations in certificate validation must be considered.

The first concern relates to the operational management of certificates. Maintaining valid certificates requires obtaining them from trusted CAs and periodically renewing them to ensure continuous validity. However, the effort involved in certificate renewal may lead to cases where certificates are not updated in a timely manner. Indeed, studies on DNS full-service resolvers have observed that, in the worst cases, 60.78% of DoT and 44.05% of DoH implementations operate with invalid certificates [20].

One potential solution to this issue is the automation of certificate renewal using the Automatic Certificate Management Environment (ACME) protocol [21]. While ACME is limited to Domain Validation and does not support Extended Validation, it can improve the reliability of certificates compared to

operating with invalid certificates.

The second concern is certificate revocation checking. When Online Certificate Status Protocol (OCSP) is used for revocation checking, communication with the OCSP responder typically occurs over HTTP(plaintext), leaking privacy [22]. Alternatively, using Certificate Revocation Lists (CRLs) for revocation checking can lead to degraded name resolution performance if the CRL becomes excessively large [23] [24].

To address these concerns, the current best practice is to adopt OCSP stapling, which balances privacy protection and performance in certificate revocation checking [25].

While these challenges represent a long road ahead, it is imperative that I continue my efforts to achieve secure DNS communication.

3.6.4 Impact on Existing Security Systems

Two major differences exist between the traditional DNS and DoH communication. The first is the shift from using UDP to a TCP-based HTTPS protocol. The second is the encryption provided by TLS. Operators of DNS full-service resolvers and authoritative DNS servers need to be fully aware of how these differences could impact potential attack methods and existing security systems.

For example, one of the causes of DNS reflection, a type of DDoS attack, was the relative vulnerability of UDP to IP spoofing [26]. However, TCP-based communication requires a three-way handshake for connection establishment,

making it difficult for attackers with spoofed IP addresses to continue communication after sending the initial SYN packet. Therefore, the resistance to DNS reflection via IP spoofing is stronger in traditional UDP connections. Nevertheless, it is necessary to consider countermeasures against DDoS attacks originating from TCP, such as SYN flood attacks, which involve sending a large number of SYN packets to consume server resources. For TCP-based attacks, it is effective to block malicious communication through an Intrusion Prevention System, which refers to unencrypted HTTPS headers and similar elements as is common in general web systems.

Encryption via TLS safeguards privacy. However, it may complicate data analysis for network administrators in ensuring security, potentially hindering the detection of malware utilizing DoH. In particular, there is a concern that identifying malware that connects to Command and Control servers using malicious DNS tunneling tools may become increasingly challenging. It has been reported that some approaches using machine learning were effective for this issue [27] [28].

3.7 Conclusion

Conventional DNS communication is done in plaintext, which is insufficient for privacy protection. I have proposed a Full-DoH DNS architecture. The proposed Full-DoH DNS architecture encrypts all DNS communication during the domain name resolution process, which involves end-terminals, DNS full-service resolver, and authoritative DNS servers so that it not only can prevent eavesdropping in the communication links but also can protect the

end-user privacy as well as the institutional privacy. Presently, a significant number of authoritative DNS servers, including root authorities, do not support DoH. Consequently, it is not yet clear whether a practically viable level of performance can be achieved if name resolution from end-terminals to these authoritative DNS servers is entirely facilitated through DoH connections.

Accordingly, I implemented a prototype system in a local experimental environment and real-world Internet to evaluate its features and performance. The prototype includes a DoH-Proxy positioned between a DNS full-service resolver and authoritative DNS servers, converting standard DNS traffic to DoH traffic. Through experiments, I confirmed that the proposed method functions correctly in both a local experimental environment and the real-world Internet. Performance evaluation showed acceptable domain resolution times under the same conditions. Further analysis revealed that the primary overhead in the Full-DoH system stems from server certificate verification on the DNS full-service resolver, and I proposed potential solutions for its optimization.

Chapter 4

A Privacy-Preserving Full DNS over HTTPS Architecture via Compatible NS Record Based Information Sharing[2]

4.1 Introduction

In section 3, I have proposed a solution based on the Full DNS over HTTPS Architecture (Full-DoH), which leverages the DoH request path to respect end-user preferences. The Full-DoH assumes that all authoritative DNS servers support DoH connections. However, considering the deployment in the real Internet, it is expected that there will be a mix of authoritative DNS servers that support encrypted communication and those that do not. Therefore, DNS full-service resolvers have to first discover the encryption methods supported by

[2]This chapter is based on "A Privacy-Preserving Full DNS over HTTPS Architecture via Compatible NS Record Based Information Sharing" [29], by the same author, © 2025 IEICE. The material in this paper was presented in part at the Proceedings of IEICE Transactions on Communications, and all the figures of this paper are reused from [29] under the permission of the IEICE.

the authoritative DNS servers, as different DNS protocols are not compatible with one another.

The methods for resolving the compatibility issues in DNS communication are different in two types of DNS communication, one between end-terminals and DNS full-service resolvers and the other between DNS full-service resolvers and authoritative DNS servers.

When end-terminals want to protect their privacy through encrypted DNS communication with DNS full-service resolvers, the standard practice for the end-terminals is to manually configure their devices with public DNS full-service resolver services that support encryptions, such as Google Public DNS [30], Quad9 [31], or Cloudflare [32].

Manual configuration between end-terminals and DNS full-service resolvers is considered acceptable because DNS communication typically requires only a one-time setup of primary and secondary DNS servers in advance.

Communication between DNS full-service resolvers and authoritative DNS servers typically follows a 1:N relationship, making it impractical for DNS full-service resolvers to configure protocol information for each authoritative DNS server manually. Therefore, when conducting encrypted DNS communication between DNS full-service resolvers and authoritative DNS servers, it is necessary for the DNS full-service resolvers to automatically discover the encryption methods supported by the authoritative DNS servers.

The IETF has standardized Service Binding Mapping for DNS Servers as a method for DNS full-service resolvers to discover the encryption protocols

supported by authoritative DNS servers (RFC9461) [14]. Figure 4.1 illustrates an overview of the RFC 9461 method. A key issue with the RFC 9461 method is that plain text is used until encrypted communication is established. Consequently, this approach poses a risk of exposing privacy-related information from DNS queries to third parties along the communication path.

In RFC 9539, the IETF has standardized a method that allows simultaneous queries for both plain text and encrypted communication [33]. Figure 4.2 presents an overview of the RFC 9539 method. RFC 9539, when compatibility is prioritized, it becomes necessary to use both plaintext and encrypted communication concurrently, which introduces a risk of privacy leakage.

To address these issues, in this paper, I propose a new approach in which the parent authoritative DNS server indicates the encryption protocols supported by the child authoritative DNS server in the NS records during the zone delegation without violating the conventional DNS name resolution. Figure 4.3 presents an overview of the proposed method. In the proposed method, the encryption method supported by an authoritative DNS server is first discovered using the NS records in the root.hints file. Subsequently, the encryption method supported by the delegated authoritative DNS server is obtained through NS records from the delegating authoritative DNS server. This approach enables encrypted communication without relying on plain text transmission.

The main contributions of this paper are as follows:

- Design and implement a privacy-preserving full DNS over HTTPS (DoH) architecture.

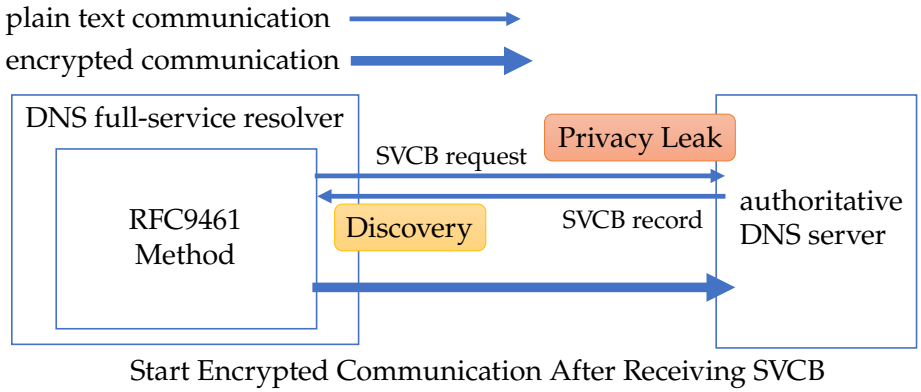

Figure 4.1: Overview of the RFC9461 method

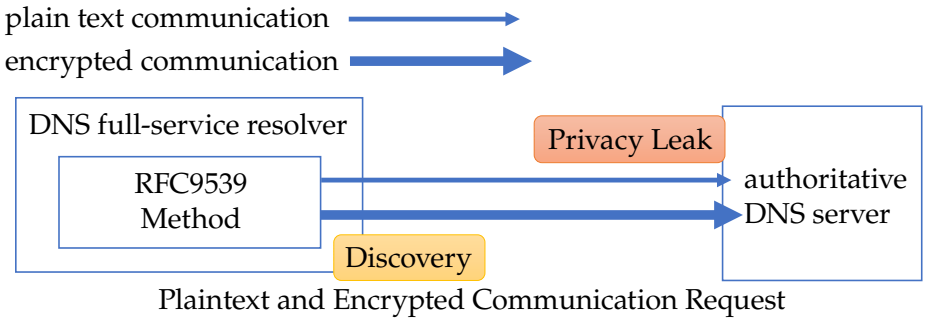

Figure 4.2: Overview of the RFC9539 method

- Propose NS record based information sharing to provide a compatibility with existing DNS architecture.
- The functionality and performance using the proposed method are evaluated using the prototype system.
- A discussion is provided on the proposed method and further enhancements for DNS privacy protection.

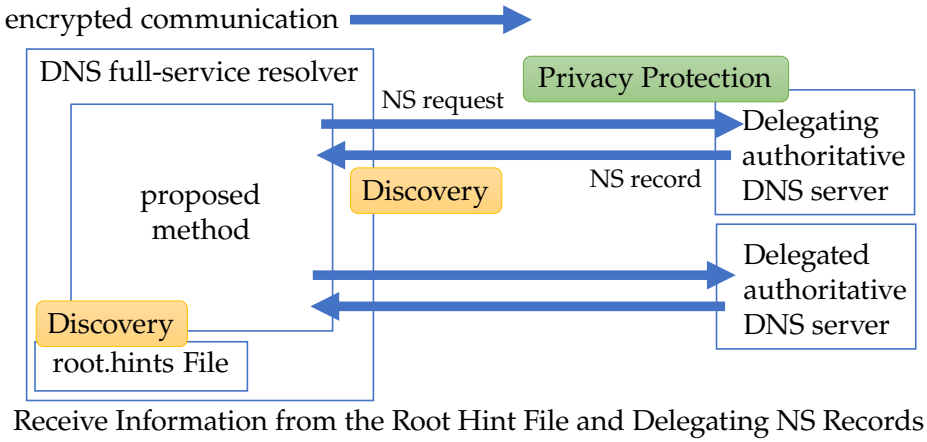

Figure 4.3: Overview of the proposed method

4.2 Proposed method

As described in Sect. 4.1, RFC 9461 and RFC 9539 still present challenges regarding privacy protection. I propose leveraging the delegation relationships within the DNS to ensure communication compatibility and discover encryption protocols without relying on plain text communication.

The specific method involves embedding encryption protocol support information as a message within the NS records of the parent-authoritative server. An example of the embedded message is shown in Table 4.1. If DoH is supported, the NS record includes "`dh-e-`", while if DoT is supported, it includes "`dt-e-`".

Figure 4.4 shows an example of NS records in the proposed method. In this example, there are two authoritative DNS servers (`ns1.example.com` and `ns2.example.com`) in `example.com` domain. Among them, `ns1.example.com` supports DoT and DoQ, while `ns2.example.com` supports DoH and Do53, both of which are specified in the red text of NS record entries in the figure.

With the proposed NS record entries, DNS full-service resolvers can recognize the encryption protocols supported by each authoritative DNS server so that they can initiate encrypted communication with the delegated authoritative DNS server without relying on plain text communication.

Table 4.1: NS record expansion

String	Meaning of the string
`dh-e-`	DoH Enable
`dt-e-`	DoT-Enable
`dq-e-`	DoQ-Enable
`dh3-e-`	DoH/3 Enable
`do53-e-`	UDP (Plain text) Enable
`dh-d-`	DoH Disable
`dt-d-`	DoT-Disable
`dq-d-`	DoQ-Disable
`dh3-d-`	DoH/3 Disable
`do53-d-`	UDP (Plain text) Disable

Figure 4.5 illustrates the flow of the proposed method. The specific communication steps are as follows:

Step (1): A fundamental assumption of the proposed method is that the `root.hints` file of the DNS full-service resolver contains the latest information of the encryption protocols supported by each root authoritative

```
example.com NS ns1.example.com
example.com NS ns2.example.com
example.com NS dt-e-dq-e-ns1.example.com
example.com NS dh-e-do53-e-ns2.example.com

ns1.example.com A 192.168.53.1
ns2.example.com A 192.168.53.2
dt-e-dq-e-ns1.example.com A 192.168.53.1
dh-e-do53-e-ns2.example.com A 192.168.53.2
```

Figure 4.4: Example of NS records in the Proposed Method

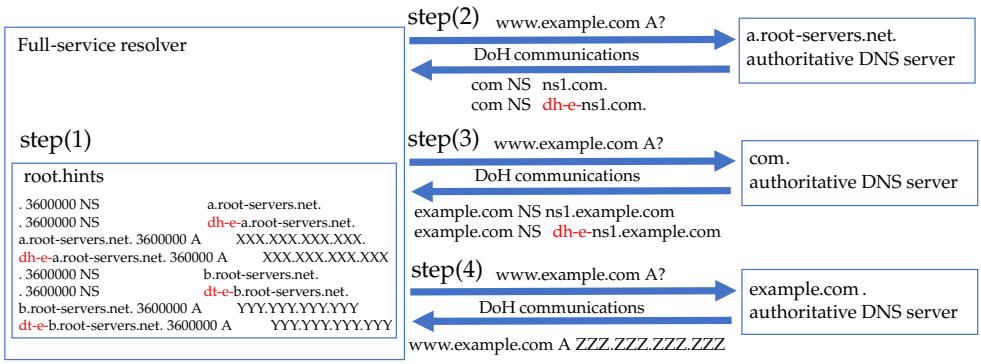

Figure 4.5: Communication flow in the Proposed Method

DNS server. Generally, the root.hints file is obtained from InterNIC or mirror sites. In the proposed method, it is necessary to manually get the file [34].

Step (2): Since the DNS full-service resolver can obtain information on the encryption protocols supported by root authoritative DNS servers from the root.hints file, it can initiate encrypted communication with a root authoritative DNS server from the beginning. When resolving names through a root authoritative DNS server, the resolver receives NS records that include information on the encryption protocols supported by the delegated servers. The DNS full-service resolver identifies the encryption protocols supported by the delegated servers based on the strings contained in the NS records.

Step (3): Since the root authoritative DNS server provides information on the encryption protocols supported by the `.com` domain, encrypted communication can be established. Similar to Step (2), the resolver identifies the encryption protocols supported by the delegated authoritative DNS

servers from the NS records.

Step (4): Since the .com authoritative DNS server provides information on the encryption protocols supported by example.com, encrypted communication can be established. The DNS full-service resolver then retrieves the A record for `www.example.com` and returns the name resolution result to the end-terminal.

Steps (1) to (4) represent the normal processing flow. My proposed method assumes that the NS records of all authoritative DNS servers contain at least one "enable" entry from Table 4.1 as a prerequisite. Moreover, unsupported protocols are explicitly marked with "disable".

Since DNS label sizes are limited to 63 bytes, it is desirable for prefixes to be concise and, where appropriate, omittable [16]. Therefore, an operational practice that authoritative DNS servers may adopt is to advertise a label such as `dh-e-ns1.example.com`, indicating support for DoH while omitting Do53, which may still be implicitly supported.

In my proposal, to accommodate such implicit support, the omission of a protocol is interpreted not as an indication of non-support but rather as an unknown state. To explicitly indicate that a protocol is not supported, the "disable" prefix should be used.

For example, if a domain label is specified as `dh-e-ns1.example.com`, a DNS full-service resolver can detect that at least the DoH protocol is supported. The support status of other protocols remains unknown; however, since Do53 is widely deployed, the DNS full-service may optimistically assume that it is supported despite not being explicitly listed. If the server prioritizes privacy

and discourages the use of plaintext, it may use `dh-e-do53-d-ns1.example.com` to signal to the resolver that Do53 fallback is not available.

An alternative design would be to treat all unspecified protocols as unsupported, thereby eliminating the need for the "disable" prefix. However, in my proposal, I intentionally allow for the implicit support of unspecified protocols by treating them as having an unknown support status.

When an authoritative DNS server supports multiple encryption protocols, it is up to the DNS full-service resolver to determine which connection to prioritize. However, as noted in RFC 9539 Section 4.6.11, `Maintaining Connections`, it is recommended to give preference to a specific encrypted DNS protocol when maintaining sessions after connection establishment, in order to minimize the cost and latency associated with encrypted DNS connections [33]. For example, one possible operational strategy is to prioritize the encrypted protocol listed first in the NS record, and if the connection attempt fails, to fall back to the subsequent protocols in the order they are listed.

When operators of authoritative DNS servers choose to support multiple encrypted DNS protocols, they should be aware that the DNS full-service resolver determines which protocol to use. For example, a resolver may initiate connections using several protocols—such as DoT, DoH, and DoQ—simultaneously, maintain all of them, and send queries over each, ultimately selecting the response that arrives first. To prevent this behavior, operators are advised either to support only a single encrypted protocol or to deploy separate servers for each protocol. For instance, _dns_dt_e_ns1.example.com could be configured to support only DoT, while _dns_dh_e_ns2.example.com

supports only DoH.

My proposed method requires adding a string to existing NS records, making it subject to character length constraints. The maximum length of an FQDN, including separators, is 255 octets, and the maximum length of a single label is 63 octets [16]. When applying my proposed method to existing NS records, adjustments must be made if the length exceeds these limits to ensure compliance with the character constraints. In addition, if the existing NS records already contain the proposed string (e.g., `dh-e-`), unintended encrypted communication may occur.

I assessed the impact of these concerns using the Top 10 million domains as of February 2025 and confirmed that the effect is minimal. Among the Top 10 million domains [35], the longest FQDN observed was 50 octets, while the longest single label was 47 octets. This ensures that at least one instance of the proposed string can be added without exceeding the length constraints. Furthermore, none of the 10 million domains contained the proposed string, indicating that the potential impact on existing NS records is relatively minimal. However, if an NS record containing the string is already in operation, it must be edited to avoid conflicts.

4.3 Experimental Environment

To show the feasibility of the proposed method, I perform a set of experiments. In this section, I describe the experimental environment for evaluating the proposed method. My goal for the experiments is to ensure that all authoritative DNS servers support encryption.

Among the authoritative DNS servers available on the real Internet, `b.root-servers.net` supports DoT [36]. However, most other authoritative DNS servers do not support encryption, making experiments in a real Internet environment challenging. Therefore, I construct a verification environment using server virtualization to evaluate the proposed method. The experimental environment must meet the following requirements:

Requirement 1: Experimental environment must support all DNS protocols, including Do53, DoT, DoH, DoQ, and DoH/3, each of which is used for communications involving end-terminal, DNS full-service resolver, and authoritative DNS servers.

Requirement 2: The environment enables performance comparisons between the proposed method and the method standardized in RFC 9461 using SVCB records.

A challenge in meeting Requirement 1 is that no single DNS server software fully supports all encryption protocols. The major DNS full-service resolvers, Unbound and PowerDNS Recursor, currently support only DoT [37][38]. BIND 9 introduced DoT forwarding in its stable version 9.20 [39]. However, this forwarding function simply transfers queries to another DNS full-service resolver that supports DoT, rather than directly resolving names by querying the appropriate authoritative DNS servers.

To overcome this limitation, I developed an experimental DNS full-service resolver capable of issuing requests to authoritative DNS servers using Do53, DoT, DoH, DoQ, and DoH/3. Figure 4.6 illustrates the architecture of my

Experimental Resolver and its integration with the DNS full-service resolver. I refer to the resolver implementing the proposed method as the "Proposed Method Resolver." The Proposed Method Resolver was implemented using q (`natesales-q`) [40], a DNS client written in Go, combined with Python 3. To handle incoming DoT, DoH, DoQ, and DoH/3 requests, I used `dnsdist` [41]. Although the communication between `dnsdist` and the Proposed Method Resolver occurs over Do53, this is a local, intra-host connection, ensuring no privacy leakage to external third parties. The program developed in Python 3 utilizes the socket library to implement both a UDP listener. Each listener processes DNS requests in parallel using the asyncio library. The Proposed Method Resolver leverages `natesales-q` to send DNS queries using Do53, DoT, DoH, DoQ, and DoH/3. Additionally, it is equipped with a function to inspect NS records received from authoritative DNS servers to determine whether they contain the string specified by the proposed method.

A challenge in meeting Requirement 2, similar to Requirement 1, is that existing DNS full-service resolver software does not support the functionality to query SVCB records to determine the encryption protocols supported by authoritative DNS servers and then establish encrypted communication accordingly.

To address this limitation, I extended the Proposed Method Resolver developed for Requirement 1 by implementing a mechanism to query SVCB records for the encryption protocols supported by authoritative DNS servers and initiate encrypted communication based on the query results. I refer to this extended resolver as the "RFC9461 Resolver". Unlike the Proposed

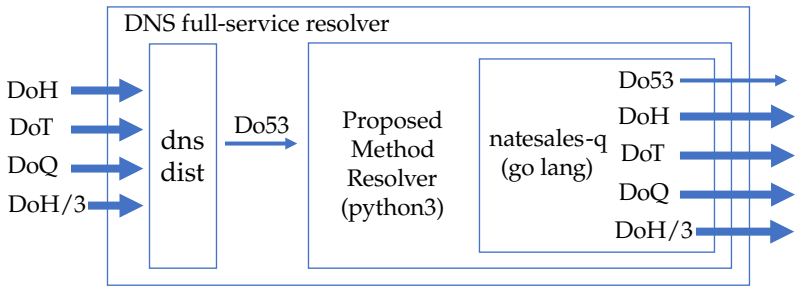

Figure 4.6: Resolver developed for the experiment

Method Resolver, the RFC 9461 Resolver does not include the functionality to inspect NS records for the proposed method.

The overall configuration of the experimental environment that meets the requirements is shown in Fig. 4.7. This experimental environment was constructed within a single server. Ubuntu 24.04 was used for all operating systems, including the host OS, end-terminals, DNS full-service resolver, authoritative DNS servers, and the guest OS serving as the web server. The DNS full-service resolver consists of `dnsdist`, a Proposed Method Resolver, and an RFC 9461 Resolver. Each authoritative DNS server is configured with `dnsdist` and BIND 9. A total of 1,000 zones, along with NS records, A records, and SVCB records, were registered for each authoritative DNS server. Apache2 was used as the web server.

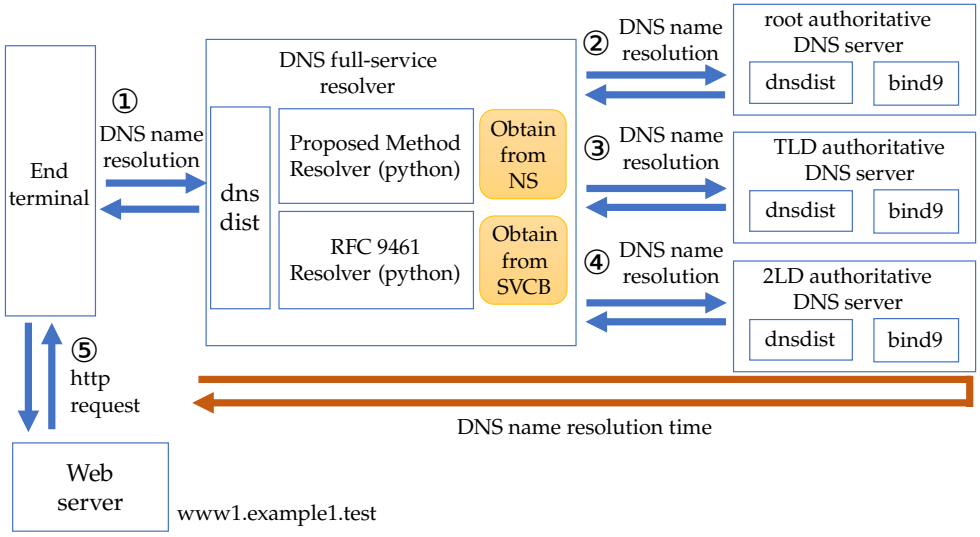

Figure 4.7: Experimental environment

4.4 Experimental Result

4.4.1 Feature Evaluation

To confirm that the constructed experimental network is working correctly, I describe the experimental results in this section. In standardized implementations, DoQ uses port 853/UDP, and HTTP/3 uses port 443/UDP. However, in the experimental environment, to facilitate recognition based on port numbers, I employ non-standard port numbers, i.e., DoQ was set to listen on port 8853/UDP, and DoH/3 was set to listen on port 8443/UDP.

Encrypted Protocol Discovery by Proposed Method: For the first functional evaluation, I confirmed that the Proposed Method Resolver establishes encrypted communication with the child-authoritative DNS server according to the NS records specified in the parent-authoritative DNS

server. Figure 4.8 shows a partial result of the Wireshark capture during this verification process. Based on the Wireshark results, I verified that the communication occurs over different encrypted protocols depending on the NS record content: DoT on port 853/TCP, DoH on port 443/TCP, DoQ on port 8853/UDP, and DoH/3 on port 8443/UDP.

Encrypted Protocol Discovery by RFC9461 Method: For the second functional evaluation, I conducted a functional evaluation of the RFC 9461 Method. Figure 4.9 presents a partial result of the Wireshark capture during this verification process. From the Wireshark results, I verified that the RFC 9461 Resolver retrieves information on the encrypted protocols supported by the authoritative DNS server via SVCB records and communicates using the corresponding encrypted protocol accordingly.

Privacy Protection Through Full DNS Encryption: For the third functional evaluation, I confirmed that all queries from the Proposed Method Resolver are encrypted. Figure 4.10 shows a partial result of the Wireshark capture during this verification process. Based on the Wireshark results, I confirmed that DNS communications remain fully encrypted throughout the name resolution process between the end-terminal, DNS full-service resolver, and authoritative DNS server.

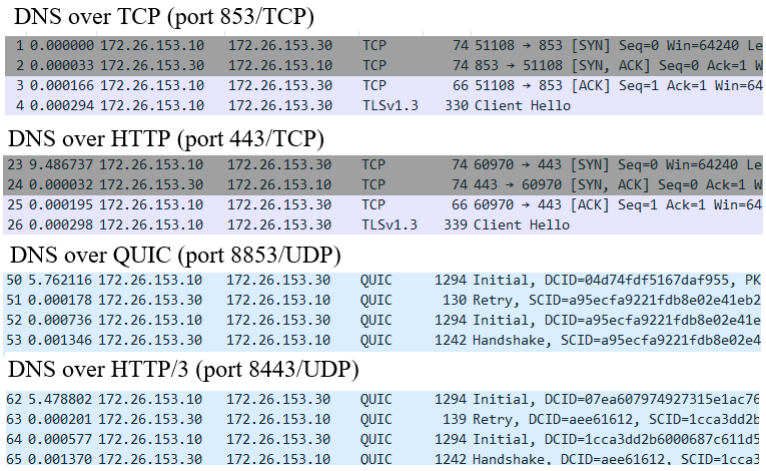

Figure 4.8: Wireshark trace of the Proposed Method Resolver

4.4.2 Performance Evaluation

In this section, I introduce a performance comparison between the proposed method and SVCB-based method standardized in RFC 9461. The measurements were conducted using the DNS testing tool `dnsperf` from the end-terminal shown in Fig. 4.7. My proposed method supports DoT, DoH, DoQ, and DoH/3. However, as my focus is on enhancing Full-DoH to respect user preferences, I used DoH for the performance comparison.

Figure 4.11 presents the name resolution time measured over 100 queries, each executed with an interval between them. While my proposed method enables encrypted communication from the initial connection, the RFC 9461 method first queries the SVCB record using plain text communication before transitioning to encrypted communication. As a result, the RFC 9461 method requires more communication steps than my proposed method, leading to a longer name resolution time.

Figure 4.12 compares the Queries Per Second (QPS) performance of the

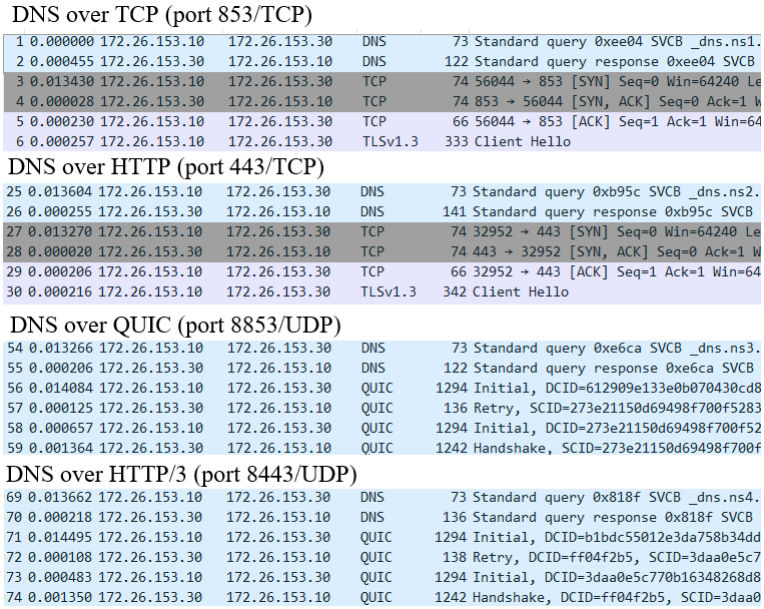

Figure 4.9: Wireshark trace of the RFC 9461 Resolver

proposed and RFC9461 methods. The x-axis shows the number of concurrent connections, and the y-axis indicates the QPS performance. In this experiment, each end-user performed a single query to resolve a second-level domain (2LD) per connection. Due to the limitations of `dnsperf`, the number of concurrent connections cannot exceed 256. Therefore, the maximum value on the y-axis is based on measurements conducted at 256 concurrent connections. In the figure, both methods demonstrate stability regardless of the number of concurrent connections. However, due to the additional SVCB record queries occurring between the DNS full-service resolver and the authoritative DNS server in the RFC 9461 method, compared to my proposed method, a difference of approximately 20 QPS was observed in the communication between the end-terminal and the DNS full-service resolver.

Figure 4.13 compares the performance of the average name resolution

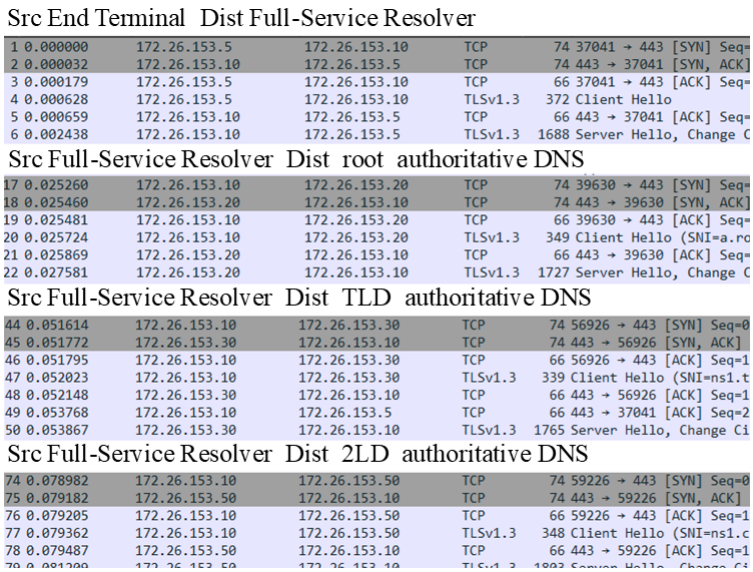

Figure 4.10: Wireshark trace of the secure NS-Based information Sharing

time between the proposed and RFC9461 methods. The x-axis represents the number of concurrent connections, while the y-axis indicates the average name resolution time. As the number of concurrent connections increases, the average resolution time for both methods also increases. However, beyond approximately 100 to 120 concurrent connections, the rate of increase becomes more gradual. The RFC9461 method reaches an average name resolution time of approximately 1 second at 200 concurrent connections, whereas the proposed method remains below 1 second even at 256 concurrent connections. This indicates that the proposed method demonstrates better performance in terms of average name resolution time.

Figure 4.14 illustrates the comparison of the total execution time of `dnsperf` between the proposed method and the RFC 9461 method as the number of concurrent connections increases. My proposed method completes all name resolutions for 200 concurrent connections (one query per connection) in

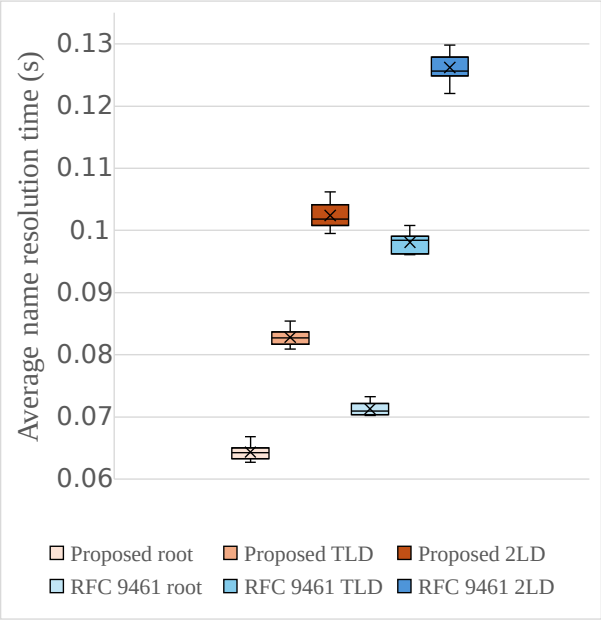

Figure 4.11: Name resolution time in Root, TLD, and 2LD (Proposed versus RFC9461)

approximately 1.6s. In contrast, the RFC 9461 method reaches 2s at 160 concurrent connections, indicating that some end-terminals are unable to complete name resolution within 2s.

Figure 4.15 compares the maximum and minimum name resolution time between the proposed and RFC9461 methods. The minimum resolution time remains nearly constant regardless of the number of concurrent connections, with the proposed method achieving approximately 0.25s and the RFC9461 method around 0.5s. These results indicate that the proposed method exhibits lower latency. Since `dnsperf` temporarily halts request transmissions once a certain query volume is exceeded until responses are received or requests time out, the maximum name resolution time per connection eventually stabilizes at a certain point. The proposed method stabilizes at approximately 0.8s, whereas the RFC9461 method reaches approximately 1.5s. In both minimum

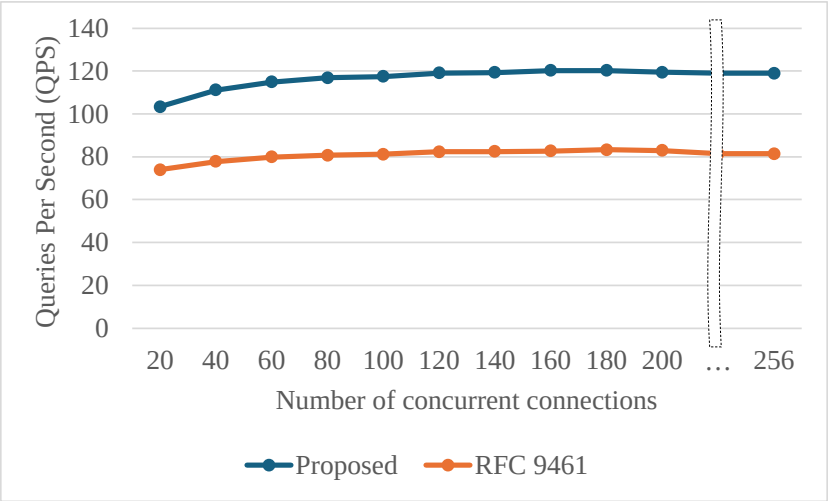

Figure 4.12: QPS performance (Proposed versus RFC9461)

and maximum name resolution times, the proposed method demonstrated superior performance compared to the RFC9461 method.

Based on these experimental results, it was confirmed that in a Full-DoH architecture, my proposed method outperforms the RFC 9461 method in terms of name resolution time, average name resolution time, and overall query processing performance.

4.5 Discussion

This chapter presents several discussions essential for the practical deployment of the proposed method. Section 4.5.1 provides a qualitative evaluation of encrypted protocol discovery and privacy protection as defined in RFC 9461, RFC 9539, and the proposed method. Section 4.5.2 discusses the advantages of combining the proposed method with SVCB records. Section 4.5.3 examines strategies to avoid prefix collisions. Section 4.5.4 describes

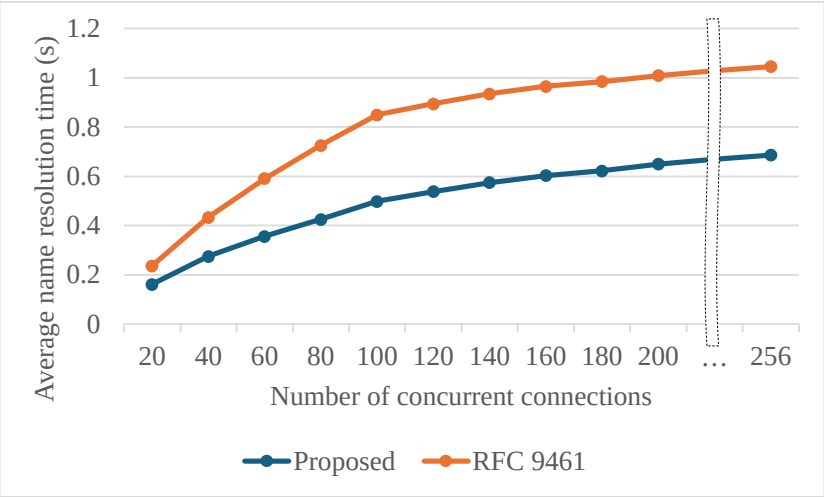

Figure 4.13: Average name resolution time performance (Proposed versus RFC9461)

a secure and reliable method for distributing root.hints, which is required for the use of the proposed method. Section 4.5.5 discusses the impact of message fragmentation caused by increased message size. Section 4.5.6 provides an in-depth discussion of potential attack vectors and challenges in the operational deployment of encrypted protocols. Finally, Section 4.5.7 outlines the limitations of the proposed method and future challenges in enhancing privacy protection in DNS communications.

4.5.1 Method Characteristics and Qualitative Comparison

The standardized methods RFC9461, RFC9539, and my proposed approach have distinct characteristics. Table 4.2 shows a qualitative evaluation of these methods. Item (1) in Table 4.2 pertain to privacy-related assessments, items (2) to (4) concern interoperability, and item (5) evaluate performance.

A detailed explanation of each item from Table 3 (1) to (5) is provided.

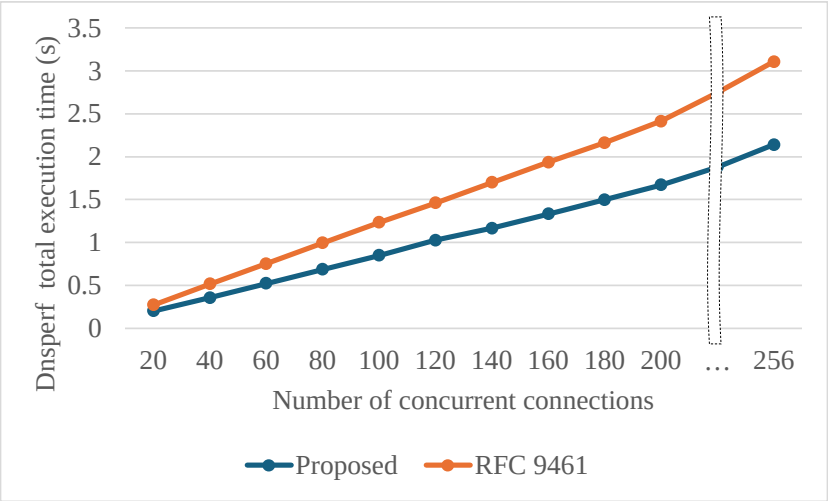

Figure 4.14: Total execution time performance (Proposed versus RFC9461)

Table4.2 (1) : There are differences among the methods regarding whether privacy information is exposed during the process leading up to the initiation of encrypted communication.

In the case of RFC 9461, the DNS full-service resolver queries the authoritative DNS server for SVCB records over plaintext communication. This occurs because the resolver has not yet discovered which encryption protocols are supported by the authoritative server.

Plaintext communication is inherently more vulnerable than encrypted communication. An on-path attacker can intercept the query and return a forged response. If the DNS full-service resolver does not validate the response using DNSSEC and proceeds to initiate a connection based on the forged SVCB record containing false connection information, sensitive privacy-related data may be leaked to the attacker.

In contrast, whether privacy leakage occurs under RFC 9539 or my

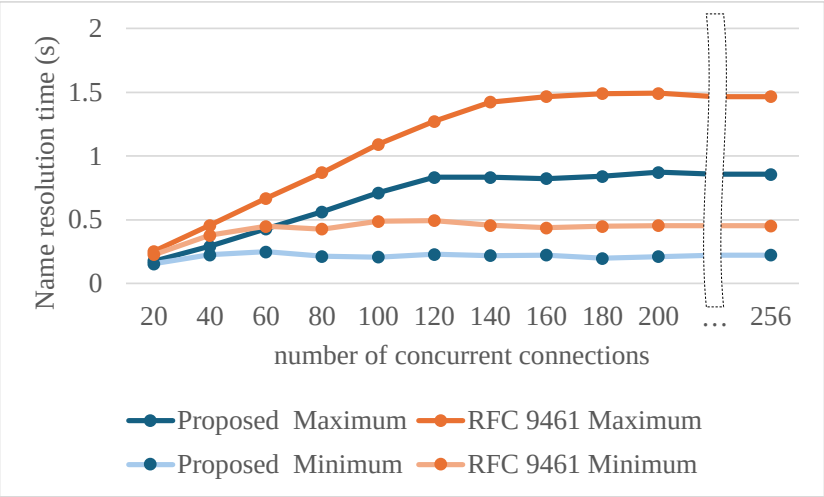

Figure 4.15: Maximum and minimum name resolution time performance (Proposed versus RFC9461)

proposed approach depends on the chosen policy. If privacy protection is prioritized and only encrypted protocols are used, no privacy leakage occurs. Unlike RFC 9461, these methods are capable of discovering encryption protocols without relying on plaintext communication.

However, when a policy prioritizes encrypted communication while retaining Do53 compatibility, RFC 9539 sends both encrypted and Do53-based DNS queries to prevent resolution failures. If encryption is supported, the client transitions to encrypted transport. Nevertheless, the concurrent use of Do53 may lead to the exposure of privacy information.

In contrast, the proposed method utilizes information from the root hints file and NS records to determine whether encrypted communication is supported by the authoritative DNS servers. When support is confirmed, it initiates encrypted communication directly without using Do53.

In this context, the proposed method demonstrates a privacy advantage

Table 4.2: Comparing encryption protocol discovery methods

	RFC 9461	RFC 9539	Proposed method
(1) Reconciling encryption protocol discovery with privacy protection	+	+	+ +
(2) Coexistence with the Do53	+ +	+ +	+ +
(3) Supports a strict mode that prohibits fallback to Do53	+ +	+ +	+ +
(4) Efficiency of encryption protocol discovery	+	+	+ +
(5) Traffic increase	-	-	-

over RFC 9539 by avoiding unnecessary exposure through Do53 when encrypted options are available.

Table4.2 (2) : Regardless of the method chosen, if compatibility and response speed are prioritized over privacy protection, Do53 can be used.

Table4.2 (3) : Regardless of the method chosen, when prioritizing privacy protection, a strict mode is supported that prohibits fallback to Do53.

Table4.2 (4) : RFC 9461 requires the resolution of the same number of SVCB records as the number of target authoritative servers, necessitating `2N` resolution queries when encrypted communication is to be established with `N` authoritative DNS servers.

RFC 9539, on the other hand, may generate unnecessary traffic, as it requires attempting communication for encrypted communication discovery even when authoritative DNS servers do not support encryption

protocols.

In contrast, the proposed method discovers the supported protocols of authoritative DNS servers by utilizing the root.hints file and NS records, completing name resolution with only `N` queries for `N` authoritative DNS servers. This approach achieves encrypted name resolution more efficiently than RFC 9461 while avoiding the superfluous traffic generated by RFC 9539's approach to non-encryption-supporting servers.

Table4.2 (5) : All approaches generate more traffic than conventional Do53. RFC 9461 requires additional SVCB record queries, the proposed method adds NS, A, and AAAA records, while RFC 9539 attempts encrypted communication even with non-supporting servers, creating unnecessary traffic.

4.5.2 Functional Considerations Beyond Encrypted Protocol Support

In this study, I focused on the discovery of encrypted protocols supported by authoritative DNS servers and the protection of user privacy. I compared my proposed method with RFC 9461 and confirmed that my approach achieves superior performance in terms of name resolution time. However, SVCB records are not limited to protocol discovery; they can also be used for other purposes. For instance, they facilitate the selection of optimal edge servers in Content Delivery Networks (CDNs), thereby reducing latency for end-terminal [42].

My proposed method is specifically designed to support encrypted protocol discovery and does not include information for optimizing CDN connections, as SVCB records do. If latency is to be considered not only in terms of DNS response time but also in terms of the time required to retrieve actual content, obtaining SVCB records is necessary.

Therefore, to enhance DNS privacy protection, I strongly recommend using my proposed method to establish encrypted communication, while resolving SVCB records only after encrypted communication has been initiated.

4.5.3 Considerations Regarding Prefix Names

According to RFC 952, valid domain name labels are limited to the use of alphabetic characters (A–Z), digits (0–9), minus sign (-), and periods (.) [43]. In my study, I followed this specification and defined a naming rule for prefixes, as shown in Table 1. However, such prefixes may potentially conflict with existing domain names. To evaluate this risk, I analyzed the Top 10 Million Domains and confirmed that no conflicts occur with my proposed prefixes. Nevertheless, this dataset does not cover all active domains, and therefore, I cannot completely rule out the possibility of name collisions.

One approach to address this issue is to employ underscore (_), which are not permitted for use under RFC 952. For example, RFC 2782 uses underscore-prefixed labels such as `_ldap._tcp.example.com` to indicate service records in the DNS [44]. Similarly, RFC 7489 specifies the use of a special label `_dmarc` in domain names to define DMARC policies [45]. As demonstrated in the example below, my proposed method can also avoid conflicts with existing

NS record strings by using underscores.

- example.com IN NS ns1.example.com
- example.com IN NS _dns_dt_e_ns1.example.com

However, at present, the use of underscores in NS records may cause validation errors during zone loading in some DNS server software. For instance, in BIND 9, such records will not be loaded unless the check-names ignore option is explicitly set.

In my experiments, I prioritized compliance with RFC 952 and therefore did not use underscores in prefix names. However, for future standardization, adopting underscore-prefixed labels can provide a practical solution for avoiding naming conflicts.

4.5.4 Managing Updates to the root.hints File

My proposal relies on the root.hints file, and therefore requires a mechanism for timely and accurate distribution across a wide range of DNS resolvers. Historically, root authoritative DNS servers have undergone updates to their root.hints file approximately once every few years, coinciding with changes to their IP addresses. Table 4.3 shows a chronological timeline of root.hints file updates corresponding to IPv4 address changes [46][47][48][49][50][51].

Updates to the root.hints file have traditionally been disseminated through various channels, beginning with announcements by Internet Assigned Numbers Authority (IANA) and individual root server operators via their official websites. These updates are then propagated through DNS server software

Table 4.3: Updates to `root.hints` file due to IPv4 address changes

Year	Root Server
2002	`j.root-servers.net`
2004	`b.root-servers.net`
2007	`l.root-servers.net`
2013	`d.root-servers.net`
2015	`h.root-servers.net`
2023	`b.root-servers.net`

vendors, operational community mailing lists, and Regional Internet Registries. DNS server operators typically obtain updated root.hints file either by accessing `https://www.internic.net/` directly or through software updates provided by DNS software maintainers.

The root.hints file updates required for my proposed method should follow the traditional procedure. For security and reliability, root authoritative DNS servers should include NS records with encryption protocol prefixes, announce updates via trusted channels, and DNS server operators should retrieve the root.hints file from `https://www.internic.net/`.

For operators who wish to update the file in a more timely manner, it is appropriate to automate the retrieval of the latest root.hints file from `https://www.internic.net` on a daily basis using scheduled execution tools such as Linux cron job.

In cases where "Running a Root Server Local to a Resolver" is already in use, updates are applied directly to the local cache of the root zone rather than to the root.hints file itself [52][53]. However, if the cached root zone data expires due to a network disruption or a similar cause, the resolver must fall back on the root.hints file to reestablish communication with the root servers.

Therefore, even in such configurations, it is advisable to schedule regular updates of the root.hints file via automated mechanisms such as cron job.

4.5.5 Concerns Regarding the Increase in Message Size

My proposed method requires the inclusion of additional NS, A, and AAAA records, which increases the message size of the ANSWER and ADDITIONAL SECTIONs in DNS responses. When the size of a single DNS message transmitted over UDP using Do53 exceeds the Ethernet Maximum Transmission Unit (MTU), fragmentation occurs. UDP fragmentation should be avoided whenever possible, as it can degrade network performance and reliability [54] [55].

DNS Flag Day 2020 recommends limiting message sizes to 1232 bytes or less as a guideline for avoiding fragmentation [56] [57]. This message size is selected as the maximum value that avoids fragmentation, even when including the IPv6 and UDP headers.

Equation (4.1) provides an approximate formula for estimating the size of a typical UDP-based DNS response message. S represents the total message size.

$$S = 12 + L_O + L_Q + \sum_{i=1}^{N} \text{Compress}(R_i) \tag{4.1}$$

First, 12 bytes are added for the DNS header. If options such as EDNS or COOKIE are enabled, their sizes are also added to the overall message as an OPT section(L_O).

The QUESTION SECTION(L_Q) of a DNS response includes the query name.

For example, when querying the name resolution of 'www.example.com,' approximately 18 bytes are added to the QUESTION SECTION. Additionally, 2 bytes each are added to represent the query type and class.

In the ANSWER and ADDITIONAL SECTIONs(R_i), name compression is applied, so in most cases, the DNS suffix portion of the FQDN is represented as a 2-byte pointer, with the length of the name server's hostname added to the overall size. Furthermore, each NS record contributes approximately 12 bytes, each A record adds around 16 bytes, and each AAAA record approximately 28 bytes of data. R_i represents the i-th record among all NS, A, and AAAA records. $N = n_\mathrm{NS} + n_\mathrm{A} + n_\mathrm{AAAA}$ denotes the total number of records. The function Compress() applies name compression to each record name.

Since the number of A and AAAA records does not necessarily match the number of NS records, Equation (1) is used as an approximate estimate of the message size.

Equation (4.2) represents the message size calculation for my proposed method.

$$\mathrm{S} = 12 + L_\mathrm{O} + L_\mathrm{Q} + 2\sum_{i=1}^{N} \mathrm{Compress}(R_i) \tag{4.2}$$

In my approach, prefixed NS records are added alongside the conventional NS records, and corresponding A and AAAA records are also added for those NS records. As a result, R_i is doubled.

Based on Equations (1) and (2), I evaluated the message size when applying the proposed method to root authoritative DNS servers in the following two cases.

Case 1 Max NS Records for example.com : In the first case, I applied the proposed method to seven NS records (`ns1.example.com` to `ns7.example.com`), along with their corresponding A and AAAA records, and measured the resulting message size.

When the proposed method is applied, the number of NS, A, and AAAA records each increases to 14. Since longer prefixes result in larger message sizes, I used the longest prefix in the experiment: `dt-e-dh-e-dq-e-dh3-e-do53-e-`. When a total of 14 NS records, including 7 modified by the proposed method, along with their associated A and AAAA records, are included, the total message size amounts to 1170 bytes. Based on this result, I confirm that for a relatively short domain such as example.com, applying the proposed method with up to seven NS records does not cause UDP fragmentation.

Case 2 When using maximum-length labels : In the second case, I investigated scenarios where FQDNs and labels are used at their maximum lengths. The maximum length of a single label is 63 bytes, and the maximum length of a fully qualified domain name (FQDN) is 255 bytes [58]. Since the proposed method involves adding a prefix to the NS record, when a label is already 63 bytes long, it becomes necessary to shorten the label to allow space for the prefix.

I conducted an experiment to determine how many NS, A, and AAAA records can be added without causing fragmentation when domain names of maximum length, such as the example below, are used:

`dt-e-dh-e-dq-e-dh3-e-do53-e-567890123456789012345.abc01234567`

```
8901234567890123456789012345678901234567890123456789.abc01234
5678901234567890123456789012345678901234567890123456789.abc01
2345678901234567890123456789012345678901234567890123456789.ex
ample.com.
```

As a result of the experiment, I confirmed that up to 8 records (8 A records and 8 AAAA records) can be accommodated without fragmentation. With 8 records, the message size was 1150 bytes; with 9 records, it increased to 1258 bytes.

Since my proposed method doubles the number of NS records, in cases where domain names are used at their maximum length, the threshold for avoiding fragmentation is up to 4 name servers prior to applying the method.

my approach increases the number of NS, A, and AAAA records; however, I confirmed that even in the case of the longest domain names, fragmentation does not occur as long as the total number of NS records remains within eight (i.e., four existing NS records and four additional ones introduced by the proposed method). When fragmentation does not occur, the impact on name resolution latency is minimal.

If fragmentation does occur, however, TCP fallback may lead to performance degradation and a decrease in system efficiency. Therefore, when applying the proposed method to authoritative DNS servers, it is important to verify the message size. If fragmentation is likely, measures such as reducing the number of original NS records or shortening label names should be considered to mitigate the issue.

4.5.6 Analysis of Potential Attack Vectors

This section discusses the security considerations related to the proposed method. When the communication between a DNS full-service resolver and an authoritative DNS server is fully encrypted using the proposed approach, an active attacker positioned along the communication path cannot read the contents of DNS queries. As a result, the system offers strong resistance to DNS cache poisoning attacks that rely on injecting forged responses.

A DNS full-service resolver that prioritizes compatibility with Do53 may allow fallback to plaintext communication in the event of encrypted communication failure. However, permitting fallback to plaintext opens the door to downgrade attacks, whereby an active adversary on the communication path can deliberately cause the encrypted connection to fail and thereby force a fallback to unencrypted transmission. Therefore, when privacy protection is a priority, fallback to plaintext communication should be avoided.

While performing a man-in-the-middle attack between a DNS full-service resolver and an authoritative DNS server is not trivial, an attacker capable of executing a BGP hijack can technically impersonate an authoritative server. To detect such impersonation, DNSSEC is effective in identifying tampering. Additionally, verifying certificates during the establishment of encrypted communication is another practical countermeasure. Nonetheless, in practice, many servers are operated with invalid certificates [20]. Therefore, improving certificate management and increasing the adoption rate of DNSSEC are both essential for enhancing overall security.

4.5.7 Limitations of Privacy Protection in the Proposed Method

The proposed approach enables the secure discovery of encryption protocols and achieves protection of DNS messages through encryption. However, the DNS communication header remains unencrypted. This implies that while the contents of queries can be concealed, it is not possible to hide which name server the DNS full-service resolver is communicating with. Specifically, information such as the destination IP address and the Server Name Indication (SNI) is included in the header, and thus, the proposed method cannot prevent leakage of such information to an active adversary monitoring the communication path.

Therefore, if one aims to protect not only DNS messages but also privacy information contained in DNS communication headers, it is necessary to introduce additional mechanisms. One possible approach is to employ a proxy, such as that used in ODoH, between the DNS full-service resolver and the authoritative DNS server. Applying a proxy mechanism such as ODoH to the communication between DNS full-service resolvers and authoritative DNS servers has the potential to significantly hinder an on-path attacker's ability to extract privacy information from communication headers or Server Name Indication (SNI).

However, this privacy assurance relies on two key assumptions: first, that multiple DNS full-service resolvers make sufficient use of the proxy, and second, that there is no collusion between the attacker and the proxy operator. If the proxy is sparsely used, the attacker may be able to correlate communication timings between the client and the proxy and between the proxy and

the authoritative server, thereby identifying the parties involved. Moreover, if the attacker operates the proxy themselves or collaborates with the proxy operator, they can observe the full communication flow, undermining the privacy guarantees. Thus, caution is required, as privacy cannot be fully ensured under such conditions.

While ODoH can protect privacy information contained in communication headers, it does not provide a mechanism for discovering whether authoritative DNS servers support encryption. Therefore, combining ODoH with my proposed method is expected to enable stronger privacy protection.

4.6 Conclusion

To protect DNS privacy, the IETF has standardized RFC 9539 and RFC 9461. However, even within these standards, challenges remained in ensuring privacy protection. To address this issue, in this paper, I proposed and evaluated a novel approach that leverages the characteristics of NS records, which provide information about child-authoritative DNS servers from parent-authoritative DNS servers.

Based on the functional evaluation results, it has been demonstrated that the proposed method does not result in privacy leakage caused by plaintext communication, which was a concern in RFC 9461 and RFC 9539. Additionally, the performance evaluation results have shown that it enables communication with lower latency compared to the method described in RFC 9461. By employing the proposed method, encrypted communication between DNS full-service resolvers and authoritative DNS servers can be initiated without

the use of plain text, thereby wholly preventing privacy leaks from DNS queries.

However, standardized encryption protocols such as DoT, DoH, and DoQ cannot protect the SNI in communication headers. Mechanisms such as ESNI and ECH have been introduced to protect SNI; however, these approaches rely on DNS and cannot be applied to communication between DNS full-service resolvers and authoritative DNS servers. At present, protecting SNI in communication between DNS full-service resolvers and authoritative DNS servers remains challenging. In future research, I plan to address privacy protection in DNS communication, including the communication headers.

Chapter 5

Conclusion

The objective of this study is to achieve complete encryption of the communication path between DNS full-service resolvers and authoritative DNS servers, which has remained unprotected in conventional encryption protocols, and thereby provide comprehensive privacy protection across the entire DNS resolution path. Although RFC 9461 and RFC 9539 introduce mechanisms to enhance DNS privacy, they still rely on plaintext communication during the discovery phase.

To address these limitations, this study proposes an integrated two-phase solution. The first phase establishes a mechanism for reflecting user intentions, such as prioritizing either privacy or compatibility, by utilizing the DoH Request Path. This mechanism enables DNS full-service resolvers to adapt their behavior based on user requirements and provides the foundation for a fully encrypted Full-DoH architecture. The second phase introduces a method by which authoritative DNS servers can provide information on supported encrypted protocols directly into the NS records. This eliminates the need for additional plaintext queries during protocol discovery and enables secure and

efficient identification of encryption capabilities.

By integrating these mechanisms, the proposed architecture eliminates the privacy risks inherent in conventional approaches while maintaining flexible controllability and practical performance. The effectiveness of the proposed method has been validated through prototype implementation and comparative performance evaluation against an existing standard, namely RFC 9461.

From the perspective of privacy protection, the proposed method enables DNS full-service resolvers to determine the appropriate encryption method directly from root hints and delegation NS records. In contrast to RFC 9461, no plaintext querying of SVCB records is necessary, and unlike RFC 9539, redundant plaintext exchanges are not required. As a result, encryption can be applied from the very beginning of communication, fundamentally preventing information leakage due to eavesdropping or active attacks. Moreover, by combining this mechanism with the DoH Request Path framework, the architecture enables users to explicitly choose their desired level of privacy protection.

From a performance standpoint, the elimination of auxiliary plaintext queries yields superior results in terms of resolution latency, average response time, and queries per second when compared to RFC 9461. Fluctuations between the minimum and maximum response times are reduced, resulting in more stable operational characteristics.

This study is of importance because it presents a comprehensive encryption approach that addresses both protocol design and system architecture

for communications between DNS full-service resolvers and authoritative DNS servers. This segment of the DNS resolution process has traditionally remained outside the scope of encryption. By introducing this approach, the proposed method enables institutional privacy protection across the entire DNS resolution path. At the same time, several important challenges remain before the results of this research can be widely deployed in operational environments and established as part of the Internet's core infrastructure.

First, international adoption of the proposed approach critically depends on standardization within the IETF. The mechanisms proposed in this study, including embedding encryption capability information into NS records and reflecting user intent through the DoH Request Path, exhibit high compatibility with existing specifications. However, these mechanisms also directly affect operational policies and interoperability. Consequently, they must be formalized not merely as implementation proposals, but as standardized specifications that clearly define their relationship with existing RFCs, ensure backward compatibility, and support phased deployment scenarios. Achieving consensus through the standardization process among registries, authoritative DNS operators, and resolver vendors is a prerequisite for practical deployment on the global Internet.

Second, although the contents of DNS communications are encrypted, metadata associated with the TLS handshake still poses a risk of information leakage. Such metadata includes the Server Name Indication (SNI) and Application-Layer Protocol Negotiation (ALPN) headers. Even in communications between DNS full-service resolvers and authoritative DNS servers,

this information may allow inference of the queried zone or the protocol in use. From the perspective of the complete resolution-path privacy targeted by this research, this issue remains unresolved. Future work should therefore consider integration with related technologies such as Encrypted Client Hello (ECH) and examine approaches to metadata protection that are specifically tailored to DNS communications.

Third, further optimization in terms of performance and operational efficiency is also an important challenge. This study identifies certificate verification as the primary source of encryption-related overhead. This finding simultaneously highlights clear opportunities for improvement. Enhancements such as more sophisticated certificate caching mechanisms, more efficient trust anchor management, or the exploration of lightweight authentication schemes tailored to DNS operational models may enable implementations that reconcile strong encryption with high performance. In addition, long-term operational evaluations in high-load environments, such as large-scale resolvers and root or TLD servers, remain an important subject for future empirical validation.

Fourth, this research suggests the potential to redefine DNS not merely as a name resolution infrastructure, but as a privacy-aware control plane capable of reflecting user and organizational policies. The explicit expression of user intent via the DoH Request Path could, in the future, evolve into a framework that supports diverse requirements, such as prioritizing performance, maximizing confidentiality, or accommodating regional constraints, directly at the DNS layer. Such extensions may also open up new application possibilities in

conjunction with zero-trust networking models and secure communication architectures for institutional environments.

This study provides concrete and experimentally validated solutions to a long-standing unresolved area in DNS encryption, while also presenting a new direction for the future evolution of Internet infrastructure. By addressing the remaining challenges related to standardization, metadata protection, performance optimization, and practical applications, the outcomes of this research are expected to extend beyond academic contributions and become widely established as effective privacy protection technologies in real-world deployments.

Bibliography

[1] Glenn Greenwald and Ewen MacAskill. *NSA collecting phone records of millions of Verizon customers daily*. The Guardian. 2013. URL: https://www.theguardian.com/world/2013/jun/06/nsa-phone-records-verizon-court-order.

[2] Stephen Farrell and Hannes Tschofenig. *Pervasive Monitoring Is an Attack*. IETF RFC7258. 2014. DOI: 10.17487/RFC7258. URL: https://www.rfc-editor.org/info/rfc7258.

[3] Tim Wicinski. *DNS Privacy Considerations*. IETF RFC9076. 2021. DOI: 10.17487/RFC9076. URL: https://www.rfc-editor.org/info/rfc9076.

[4] Nicholas Weaver, Christian Kreibich, and Vern Paxson. "Redirecting DNS for Ads and Profit". In: *USENIX Workshop on Free and Open Communications on the Internet (FOCI 11)*. 2011.

[5] Zi Hu, Liang Zhu, John Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. *Specification for DNS over Transport Layer Security (TLS)*. IETF RFC7858. 2016. DOI: 10.17487/RFC7858. URL: https://www.rfc-editor.org/info/rfc7858.

[6] Paul E. Hoffman and Patrick McManus. *DNS queries over HTTPS (DoH)*. IETF RFC8484. 2018. DOI: 10.17487/RFC8484. URL: https://www.rfc-editor.org/info/rfc8484.

[7] Christian Huitema, Sara Dickinson, and Allison Mankin. *DNS over Dedicated QUIC Connections*. IETF RFC9250. 2022. DOI: 10.17487/RFC9250. URL: https://www.rfc-editor.org/info/rfc9250.

[8] Stefan Santesson and Hannes Tschofenig. *Transport Layer Security (TLS) Cached Information Extension*. IETF RFC7924. 2016. DOI: 10.17487/RFC7924. URL: https://www.rfc-editor.org/info/rfc7924.

[9] Donald Eastlake 3rd. *Transport Layer Security (TLS) Extensions: Extension Definitions*. IETF RFC6066. 2011. DOI: 10.17487/RFC6066. URL: https://www.rfc-editor.org/info/rfc6066.

[10] Eric Rescorla, Kazuho Oku, Nick Sullivan, and Christopher A. Wood. *TLS Encrypted Client Hello*. Internet-Draft draft-ietf-tls-esni-23. Internet Engineering Task Force, 2025. 53 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/23/.

[11] Eric Kinnear, Patrick McManus, Tommy Pauly, Tanya Verma, and Christopher A. Wood. *Oblivious DNS over HTTPS*. IETF RFC9230. 2022. DOI: 10.17487/RFC9230. URL: https://www.rfc-editor.org/info/rfc9230.

[12] Basileal Imana, Aleksandra Korolova, and John Heidemann. "Institutional privacy risks in sharing DNS data". In: *Proc. ACM/IRTF Applied Networking Research Workshop*. Virtual event, 2021, pp. 69–75.

[13] Stéphane Bortzmeyer, Ralph Dolmans, and Paul E. Hoffman. *DNS Query Name Minimisation to Improve Privacy*. IETF RFC9156. 2021. DOI: 10.17487/RFC9156. URL: https://www.rfc-editor.org/info/rfc9156.

[14] Benjamin M. Schwartz. *Service Binding Mapping for DNS Servers*. IETF RFC9461. 2023. DOI: 10.17487/RFC9461. URL: https://www.rfc-editor.org/info/rfc9461.

[15] Satoru Sunahara, Yong Jin, Katsuyoshi Iida, and Yoshiaki Takai. "A Framework for Institutional Privacy Considered Full DNS Over HTTPS Architecture". In: *IEEE Access* 13 (2025), 36951–36964. ©2025 IEEE. Reprinted, with permission, from [full citation of original published article]. DOI: 10.1109/ACCESS.2025.3542633.

[16] Robert Elz, and Randy Bush. *Clarifications to the DNS Specification*. IETF RFC2181. 1997. DOI: 10.17487/RFC2181. URL: https://www.rfc-editor.org/info/rfc2181.

[17] Stéphane Bortzmeyer, Ralph Dolmans, and Paul E. Hoffman. *DNS query name minimisation to Improve Privacy*. IETF RFC9156. 2021. DOI: 10.17487/RFC9156. URL: https://www.rfc-editor.org/info/rfc9156.

[18] Takashi Isobe, Satoshi Tsutsumi, Koichiro Seto, Kenji Aoshima, and Kazutoshi Kariya. "10 Gbps implementation of TLS/SSL accelerator on FPGA". In: *Proc. IEEE Int'l Workshop on Quality of Service (IWQoS)*. Beijing, China, Jun. 2010, pp. 1–6.

[19] Panos Kampanakis and Michael Kallitsis. "Faster Post-Quantum TLS Handshakes Without Intermediate CA Certificates". In: *Proc. Cyber Security, Cryptology, and Machine Learning (CSCML 2022)*. Lecture Notes in Computer Science, vol. 13301, 2022, pp. 337–355. ISBN: 978-3-031-07689-3.

[20] Ruixuan Li, Xiaofeng Jia, Zhenyong Zhang, Jun Shao, Rongxing Lu, Jingqiang Lin, Xiaoqi Jia, and Guiyi Wei. "A Longitudinal and Comprehensive Measurement of DNS Strict Privacy". In: *IEEE/ACM Transactions on Networking* 31.6 (2023. DOI: 10.1109/TNET.2023.3262651), pp. 2793–2808. DOI: 10.1109/TNET.2023.3262651.

[21] Richard Barnes, Jacob Hoffman-Andrews, Daniel McCarney, and James Kasten. *Automatic Certificate Management Environment (ACME)*. RFC 8555. 2019. DOI: 10.17487/RFC8555. URL: https://www.rfc-editor.org/info/rfc8555.

[22] Stefan Santesson, Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Dr. Carlisle Adams. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*. RFC 6960. 2013. DOI: 10.17487/RFC6960. URL: https://www.rfc-editor.org/info/rfc6960.

[23] Sharon Boeyen, Stefan Santesson, Tim Polk, Russ Housley, Stephen Farrell, and David Cooper. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 5280. 2008. DOI: 10.17487/RFC5280. URL: https://www.rfc-editor.org/info/rfc5280.

[24] Russ Housley. *Internationalization Updates to RFC 5280*. RFC 9549. 2024. DOI: 10.17487/RFC9549. URL: https://www.rfc-editor.org/info/rfc9549.

[25] Eric Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446. 2018. DOI: 10.17487/RFC8446. URL: https://www.rfc-editor.org/info/rfc8446.

[26] Ahmed K. Soliman, Cherif Salama, and Hoda K. Mohamed. "Detecting DNS Reflection Amplification DDoS Attack Originating from the Cloud". In: *Proc. IEEE Int'l Conf. Computer Engineering and Systems (ICCES)*. Cairo, Egypt, 2018. DOI: 10.1109/ICCES.2018.8639414. Pp. 145–150. DOI: 10.1109/ICCES.2018.8639414.

[27] Rikima MITSUHASHI, Yong JIN, Katsuyoshi IIDA, and Yoshiaki TAKAI. "DGA-based Malware Communication Detection from DoH Traffic Using Hierarchical Machine Learning Analysis". In: *IEICE Transactions on Information and Systems* advpub (2024. DOI:10.1587/transinf.2024NTP0004), 2024NTP0004. DOI: 10.1587/transinf.2024NTP0004.

[28] Akarsh Aggarwal and Manoj Kumar. "An ensemble framework for detection of DNS-Over-HTTPS (DOH) traffic". In: *Multimedia Tools and Applications* 83.11 (2024. DOI: 10.1007/s11042-023-16956-9), pp. 32945–32972.

[29] Satoru Sunahara, Yong Jin, Katsuyoshi Iida, Nariyoshi Yamai, and Yoshiaki Takai. "A privacy-preserving full DNS over HTTPS architecture via compatible NS record based information sharing". In: *IEICE Transactions on Communications* E109-B.2 (2026).

[30] Google LLC. *Google Public DNS*. URL: https://developers.google.com/speed/public-dns.

[31] Quad9 Foundation. *Quad9*. URL: https://quad9.net/.

[32] Cloudflare, Inc. *cloudflare DNS*. URL: https://developers.cloudflare.com/dns/.

[33] Daniel Kahn Gillmor, Joey Salazar, and Paul E. Hoffman. *Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS*. IETF RFC9539. 2024. DOI: 10.17487/RFC9539. URL: https://www.rfc-editor.org/info/rfc9539.

[34] InterNIC. *root name servers*. URL: https://www.internic.net/domain/named.root.

[35] DomCop. *Top 10 million domains Based on Open PageRank data*. URL: https://www.domcop.com/top-10-million-domains.

[36] USC/ISI's DNS Root Server. *B-Root offers experimental support for DNS over TLS*. URL: https://b.root-servers.org/news/2023/02/28/tls.html.

[37] NLnet Labs. *Unbound*. URL: https://www.nlnetlabs.nl/projects/unbound/about/.

[38] PowerDNS. *PowerDNS Recursor*. URL: https://www.powerdns.com/powerdns-recursor.

[39] Internet Systems Consortium, Inc. *ISC Bind9*. URL: https://www.isc.org/bind/.

[40] natesales. *natesales/q*. URL: https://github.com/natesales/q.

[41] PowerDNS. *dnsdist*. URL: https://www.dnsdist.org/index.html.

[42] Benjamin M. Schwartz, Mike Bishop, and Erik Nygren. *Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)*. IETF RFC9460. 2023. DOI: 10.17487/RFC9460. URL: https://www.rfc-editor.org/info/rfc9460.

[43] K. Harrenstien, M. Stahl, and E. Feinler. *DoD Internet host table specification*. IETF RFC952. 1985. DOI: 10.17487/RFC0952. URL: https://www.rfc-editor.org/info/rfc952.

[44] Arnt Gulbrandsen, P. Vixie, and Levon Esibov. *A DNS RR for specifying the location of services (DNS SRV)*. IETF RFC2782. 2000. DOI: 10.17487/RFC2782. URL: https://www.rfc-editor.org/info/rfc2782.

[45] Murray Kucherawy and Elizabeth Zwicky. *Domain-based Message Authentication, Reporting, and Conformance (DMARC)*. IETF RFC7489. 2015. DOI: 10.17487/RFC7489. URL: https://www.rfc-editor.org/info/rfc7489.

[46] *J-root was renumbered*. URL: https://j.root-servers.org/.

[47] *New IPv4 address for b.root-servers.net*. URL: https://b.root-servers.org/.

[48] *L.ROOT-SERVERS.NET Change of IP address OCT-2007*. URL: https://l.root-servers.org/.

[49] *Advisory — D-root is changing its IPv4 address on 3 January 2013*. URL: https://d.root-servers.org/.

[50] *H-Root Will Change Its Addresses on 1 December 2015*. URL: https://d.root-servers.org/.

[51] *New addresses for b.root-servers.net*. URL: https://d.root-servers.org/.

[52] Warren Kumari and Paul E. Hoffman. *Decreasing Access Time to Root Servers by Running One on Loopback*. IETF RFC7706. 2015. DOI: 10.17487/RFC7706. URL: https://www.rfc-editor.org/info/rfc7706.

[53] Warren Kumari and Paul E. Hoffman. *Running a Root Server Local to a Resolver*. IETF RFC8806. 2020. DOI: 10.17487/RFC8806. URL: https://www.rfc-editor.org/info/rfc8806.

[54] Ron Bonica, Fred Baker, Geoff Huston, Bob Hinden, Ole Trøan, and Fernando Gont. *IP Fragmentation Considered Fragile*. IETF RFC8900. 2020. DOI: 10.17487/RFC8900. URL: https://www.rfc-editor.org/info/rfc8900.

[55] Amir Herzberg and Haya Shulman. "Fragmentation Considered Poisonous, or: One-domain-to-rule-them-all.org". In: *2013 IEEE Conference on Communications and Network Security (CNS)*. 2013, pp. 224–232. DOI: 10.1109/CNS.2013.6682711.

[56] *DNS flag day 2020*. URL: https://www.dnsflagday.net/2020/.

[57] Kazunori Fujiwara and Paul A. Vixie. *IP Fragmentation Avoidance in DNS over UDP*. IETF RFC9715. 2025. DOI: 10.17487/RFC9715. URL: https://www.rfc-editor.org/info/rfc9715.

[58] P. Mockapetris. *Domain names - implementation and specification*. IETF RFC1035. 1987. DOI: 10.17487/RFC1035. URL: https://www.rfc-editor.org/info/rfc1035.

List of Publications by the Author

Journal Papers

[1] S. Sunahara, Y. Jin, K. Iida and Y. Takai, "A Framework for Institutional Privacy Considered Full DNS Over HTTPS Architecture," IEEE Access, vol. 13, pp. 36951-36964, Feb. 2025, doi: 10.1109/ACCESS.2025.3542633.

[2] S. Sunahara, Y. Jin, K. Iida, N. Yamai and Y. Takai, "A privacy-preserving full DNS over HTTPS architecture via compatible NS record based information sharing," IEICE Transactions on Communications, vol. E109-B, no. 2, 14pages, Feb. 2026, doi: 10.23919/transcom.2025CEP0007.

Reviewed Conference Papers

[1] S. Sunahara, Y. Jin and K. Iida, "A proposal of DoH-based domain name resolution architecture including authoritative DNS servers," 32nd International Telecommunication Networks and Applications Conference (ITNAC), Wellington, New Zealand, pp. 1-3, Nov. 2022 doi: 10.1109/ITNAC55475.2022.9998349.

[2] Satoru Sunahara, Yong Jin, and Katsuyoshi Iida. “Authoritative DNS Server Discovery Method to Enhance DNS Privacy Preservation,” In Proceedings of the on CoNEXT Student Workshop 2023 (CoNEXT). Paris, France, pp. 31–32, Dec. 2023 doi: 10.1145/3630202.3630228

[3] S. Sunahara, Y. Jin, K. Iida, N. Yamai and Y. Takai, “Privacy Preserved Achievement Method for OCSP Status and Supported Protocols in Full-DoH Architecture,” 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), Osaka, Japan, pp. 1554-1555, Jul. 2024, doi: 10.1109/COMPSAC61105.2024.00235.