

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: September 11, 2020

J. Arkko  
Ericsson  
M. Thomson  
Mozilla  
T. Hardie  
Google  
March 10, 2020

Selecting Resolvers from a Set of Distributed DNS Resolvers  
draft-arkko-abcd-distributed-resolver-selection-01

Abstract

This memo discusses the use of a set of different DNS resolvers to reduce privacy problems related to resolvers learning the Internet usage patterns of their clients.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 11, 2020.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Operational Context . . . . .	4
3. Goals and Constraints . . . . .	4
4. Query distribution strategies . . . . .	6
4.1. Client-based . . . . .	6
4.1.1. Analysis of client-based selection . . . . .	6
4.1.2. Enhancements to client-based selection . . . . .	7
4.2. Name-based . . . . .	7
4.2.1. Name reduction . . . . .	8
5. Early conclusions . . . . .	9
5.1. Analysis conclusions . . . . .	9
5.2. Recommendations . . . . .	9
5.3. Poor distribution strategies . . . . .	9
6. Effects of query distribution . . . . .	10
6.1. Caching considerations . . . . .	10
6.2. Consistency considerations . . . . .	10
6.3. Resolver load distribution and failover . . . . .	11
6.4. Query performance . . . . .	11
6.5. Debugging . . . . .	11
7. Further work . . . . .	11
8. References . . . . .	12
8.1. Normative References . . . . .	12
8.2. Informative References . . . . .	12
8.3. URIs . . . . .	13
Appendix A. Acknowledgements . . . . .	13
Authors' Addresses . . . . .	13

## 1. Introduction

The DNS [DNS] is a complex system with many different security issues, challenges, deployment models and usage patterns. This document focuses on one narrow aspect within DNS and its security.

Traditionally, systems are configured with a single DNS recursive resolver, or a set of primary and alternate recursive resolvers. Recursive resolver services are offered by organisations such as enterprises, ISPs, and global providers. Even when clients use alternate recursive resolvers, they are typically all provided by the same organisation.

The resolvers will learn the Internet usage patterns of their clients. A client might decide to trust a particular recursive resolver with information about DNS queries. However, it is

difficult or impossible to provide any guarantees about data handling practices in the general case. And even if a service can be trusted to respect privacy with respect to handling of query data, legal and commercial pressures or surveillance activity could result in misuse of data. Similarly, outside attacks may occur towards any DNS services. For a service with many clients, these risks are particularly undesirable.

This memo discusses whether DNS clients can improve their privacy through the potential use of a set of multiple recursive resolver services. The goal is indeed an improvement only. There is no expectation that it would be possible to have no part of the DNS infrastructure aware of what queries are being made, but perhaps there are mitigations that would make possible information collection from the DNS infrastructure harder.

It should be understood that this is a narrow aspect within a bigger set of topics even within privacy issues around DNS, let alone other security issues, deployment models, or the many protocol questions within DNS. Some of these other topics include detecting the tampering DNS query responses [DNSSEC], encrypting DNS queries [DOT] [DOH], application-specific DNS resolution mechanisms, or centralised deployment models. Those other topics are not covered in this memo and need to be dealt with elsewhere.

Specifically, the scope of this memo is not limited to DNS-over-TLS (DOT) or DNS-over-HTTPS (DOH) deployments nor does it take a stand on operating system vs. application or local vs. centralized DNS deployment models. This memo is intended to provide useful information for those that wish to consider trustworthiness of their recursive resolvers as a part of their privacy analysis.

Naturally, there are some interactions between different topics. For instance, privacy is affected both by what happens to data in transit and at the endpoints, so where privacy is a concern, one would expect to consider both aspects, and lack of consideration on one probably leads to issues that dwarf the problems that this memo can address. Both issues are also important aspects when considering defense against pervasive monitoring efforts [PMON].

The rest of this memo is organized as follows. Section 2 discusses the operational context that we imagine the multiple recursive resolver arrangement might be applied in. Section 3 specifies security goals for a system that employs multiple recursive resolvers.

One key aspect of a system using multiple resolvers would be how to select which particular recursive resolver to use in a particular

situation. This is discussed in Section 4. This section covers a number of possible strategies and further considerations for this selection, along with an analysis of the implications of choosing a particular strategy. There are technical issues in the use of multiple recursive resolvers, and there are both technical and non-technical questions in deciding on what recursive resolvers should even be in the set.

Some early recommendations are provided in Section 5. Section 6 discusses operational and other implications of the distributed approach. Finally, Section 7 discusses potential further work in this area.

## 2. Operational Context

Our perspective is that of a client, choosing to either distribute or not distribute its queries to a set of different resolvers. And if the client decides to use distribution, it can choose exactly how it does that.

There are obviously additional operational aspect of this - such as central configuration mechanisms, resolver selection application choices, and so on. But these are not covered in this memo.

It should also be observed that the practices suggested in this memo are currently not widely used. Operational and other issues may be discovered, such as those outlined in Section 6.

Many of these issues need further work, but this memo aims to discuss the concept and analyse its impacts before dwelling into the technical arrangements for configuring and using this particular approach.

## 3. Goals and Constraints

This document aims to reduce the concentration of information about client activity by distributing DNS queries across different resolver services, for all DNS queries in the aggregate and for DNS queries made by individual clients. By distributing queries in this way, the goal is to reduce the amount of information that any given DNS resolver service can acquire about client activity. As such, creates a benefit for the client, but also makes these resolvers less valuable targets for attacks relating to that client's activity.

Any method for distributing queries from a single client needs to consider these benefits with regards to the following constraints:

- o A careful selection of the set of trusted resolvers must be the first priority. It does not make sense to add less trustworthy resolvers merely for the sake of distribution. For instance, there is no reason to mix resolvers with good reliability or high degree of privacy regulation with other resolvers: just include the best resolvers in the set.
- o As the goal is to reduce the amount of information given to any given resolver, a strategy that tells the same information to all resolvers is a poor one. A design that results in replicating the same query toward multiple services would thus be a net privacy loss. This happens quite easily over a long period of time, unless the distribution method is carefully designed.

More subtle leaks arise as a result of distributing queries for sub-domains and even domains that are superficially unrelated, because these could share a commonality that might be exploited to link them. For instance, some web sites use names that appear unrelated to their primary name for hosting some kinds of content, like static images or videos. If queries for these unrelated names were sent to different services, that effectively allows multiple resolvers to learn that the client accessed the web site.

A distribution scheme also needs to consider stability of query routing over time. A resolver can observe the absence of queries and infer things about the state of a client cache, which can reveal that queries were made to other resolvers. In general, different queries for the same resolution context, such as sub-resources for a web page load, which have some probability of being related or linked, should be sent to the same resolver. Failure to do so reveals information to more than one resolver.

In effect, there are two goals in tension:

- o split queries between as many different resolvers as possible; and
- o reduce the spread of linkable queries across multiple resolvers.

The need to limit replication of private information about queries eliminates naive distribution schemes, such as those discussed in Section 5.3. The designs described in Section 4 all attempt to balance these different goals using different properties from the context of a query (Section 4.1) or the query name itself (Section 4.2).

#### 4. Query distribution strategies

This section introduces and analyzes several potential strategies for distributing queries to different resolvers. Each strategy is formulated as an algorithm for choosing a resolver  $R_i$  from a set of  $n$  resolvers  $\{R_1, R_2, \dots, R_n\}$ .

The designs presented in Section 4 assume that the stub resolver performing distribution of queries has varying degrees of contextual information. In general, more contextual information allows for finer-grained distribution of information between resolvers.

##### 4.1. Client-based

The simplest strategy is to distribute each different client to a different resolver. This reduces the number of users any particular service will know about. However, this does little to protect an individual user from the aggregation of information about queries at the selected resolver.

In this design clients select and consistently use the same resolver. This might be achieved by randomly selecting and remembering a resolver. Alternatively, a resolver might be selected using consistent hashing that takes some conception of client identity as input:

$$i = h(\text{client identity}) \% n$$

For the purposes of this determination, a client might be an entire device, with the selection being made at the operating system level, or it could be a selection made by individual applications. In the extreme, an individual application might be able to partition its activities in a way that allows it to direct queries to multiple resolvers.

##### 4.1.1. Analysis of client-based selection

This is a simple and effective strategy, but while it provides distribution of DNS queries in the aggregate, it does little to divide information about a particular client between resolvers. It effectively only reduces the number of clients that each resolver can acquire information about. This provides systemic benefit, but does not provide individual clients with any significant advantage as there is still some resolver service that has a complete view of the user's DNS usage patterns.

In addition, there are specific issues where this selection method is used in particular deployment modes. Where different applications

make independent resolver selections, activities that involve multiple applications can result in information about those activities being exposed to multiple resolvers. For instance, an application could open another application for the purposes of handling a specific file type or to load a URL. This could expose queries related to the activity as a whole to multiple resolvers.

Making different selections at the level of a device resolves this issue. But of course it is still possible that an individual who uses multiple devices might perform similar activities on those devices, but have DNS queries distributed to different resolvers, resulting in replicating that individual's information to multiple resolvers. The individual may or may not be identifiable through fingerprinting of the specific set of queries being made from the devices.

#### 4.1.2. Enhancements to client-based selection

Clients can break continuity of records by occasionally resetting state so that a different resolver is selected. A client might choose to do this when it moves to a new network location, and may otherwise appear as a new client its current resolver. But it is unclear if there's a sufficient advantage to breaking continuity, as the potential benefits are offset by the client's information being disclosed to several resolvers as part of performing a series of resets. And it is possible that a particular individual's usage patterns can be identified across network locations and periods of using other resolvers.

Breaking continuity is less effective if any state, in particular cached results, is retained across the change. If activities that depend on DNS querying are continued across the change then it might be possible for the old resolver to make inferences about the activity on the new resolver, or the new resolver to make similar guesses about past activity. As many modern applications provide session continuity features across shutdowns and crashes, this can mean that finding an appropriate point in time to perform a switch can be difficult.

#### 4.2. Name-based

Clients might also additionally attempt to distribute queries based on the name being queried. This results in different names going to different resolvers.

A naive algorithm for name distribution uses the target name as input to a fixed hash:

$i = h(\text{queried name}) \% n$

However, this simplistic approach fails to prevent related queries from being distributed to different resolvers in several ways. For instance, queries that are executed after receiving a CNAME record in a response will leak the same information as the original query that resulted in the CNAME record. Services that use related domain names - such as where "example.com" uses "static.example.com" or "cdn.example.net" - might reveal the use of the combined service to a resolver that receives a query for any associated name. In both cases, sensitive information is effectively replicated across multiple resolvers.

#### 4.2.1. Name reduction

In order to reduce the effect of distributing similar names to different servers, a grouping mechanism might be used. Leading labels in names might be erased before being input to the hashing algorithm. This requires that the part of the suffix that is shared between different services can be identified. For the purposes of ensuring that queries are consistently routed to the same resolver, a weak signal is likely sufficient.

Several options for grouping domain names into equivalence sets might be used:

- o The public suffix list [1] provides a manually curated list of shared domain suffixes. Names can be reduced to include one label more than the list allows, referred to as effective top-level domain plus one (eTLD+1). This reduces the number of cases where queries for domains under the same administrative control are sent to different resolvers.
- o Services often relies on multiple domain names across different eTLD+1 domains. Developing equivalence sets might be needed to avoid broadcasting queries to servers. Mozilla maintains a manually curated equivalence list [2] for web sites that aims to maps the complete set of unrelated names used by services to a single service name.
- o Other technologies, such as the proposed first party sets [3] or the abandoned DBOUND [DBOUND] provide domain owners a means to declare some form of equivalence for different names.

Each of these techniques are imperfect in different ways. They may also skew the distribution of queries in ways that might concentrate information on particular resolvers. Moreover, resolver choice based solely on the name to be resolved rather than per-client information



reduces the anonymity set of queries sent to each resolver. In contrast to a client-based strategy, attackers can predict the target resolver for a given name using a name-based strategy. This may have implications for on-path attacker attempts to identify otherwise encrypted queries. Of course, even a name-based mechanism might use some non-public information as well for its choice, which might reduce these issues.

## 5. Early conclusions

### 5.1. Analysis conclusions

Both the client-based and more advanced name-based strategies may provide benefits. The former may provide primarily a systemic benefit, while the latter may provide also some privacy benefits to each individual client. However, neither strategy is perfect, and can leak the same information to multiple resolvers in some cases.

### 5.2. Recommendations

Both strategies are, however, likely generally beneficial in the common cases, and can improve the overall privacy situation. And they are certainly a considerable privacy improvement over a situation where a large number of clients use a single resolver.

Their use may also reduce any pressures against specific resolvers, as information available in these specific resolvers does not constitute all information about all clients. As such, the use of one of these distribution strategies is tentatively recommended, subject to further testing, discussion, and resolving any remaining operational issues.

The naive name-based strategy is, however, not recommended, and neither are other, even simpler strategies listed in Section 5.3. It should be noted that no technique presented in this memo can defend against a situation where an actor such as a surveillance agency has access to information from all resolvers.

### 5.3. Poor distribution strategies

Random allocation to a resolver might be implemented:

```
i = rand() % n
```

Similar drawbacks can be seen where clients iterate over available resolvers:

```
i = counter++ % n
```

Whether this choice is made on a per-query basis, these two methods eventually provide information about all queries to all resolvers over time. Domain names are often queried many times over long periods, so queries for the same domain name will eventually be distributed to all resolvers. Only one-off queries will avoid being distributed.

Implementing either method at a much slower cadence might be effective, subject to the constraints in Section 4.1.2. This only slows the distribution of information about repeated queries to all resolvers.

## 6. Effects of query distribution

Choosing to use more than one DNS resolver has broader implications than just the effect on privacy. Using multiple resolvers is a significant change from the assumed model where stub resolvers send all queries to a single resolver.

### 6.1. Caching considerations

Using a common cache for multiple resolvers introduces the possibility that a resolver could learn about queries that were originally directed to another resolvers by observing the absence of queries. Though this can reduce caching performance, clients can address this by having a per-resolver cache and only using the cache for the selected resolver.

### 6.2. Consistency considerations

Making the same query to multiple resolvers can result in different answers. For instance, DNS-based load balancing can lead to different answers being produced over time or for different query origins. Or, different resolvers might have different policies with respect to blocking or filtering of queries that lead to clients receiving inconsistent answers.

In the extreme, an application might encounter errors as a result of receiving incompatible answers, particularly if a server operator (incorrectly) assumes that different DNS queries for the same client always originate from the same source address. This is most likely to occur if name-based selection is used, as queries could be related based on information that the client does not consider.

### 6.3. Resolver load distribution and failover

Any selection of resolvers that is based on random inputs will need to account for available capacity on resolvers. Otherwise, resolvers with less available query-processing capacity will receive too high a proportion of all queries. Clients only need to be informed of relative available capacity in order to make an appropriate selection. How relative capacities of resolvers are determined is not in scope for this document.

The choice of different resolvers would also need to work well with whatever mechanisms exist for failover to alternate resolvers when one is not responsive. The same is true of IPv4/IPv6 connectivity, the availability of communications to specific ports, etc. And the dynamic situation should obviously not lead to extensive leakage to different resolvers, either.

### 6.4. Query performance

Distribution of queries between resolvers also means that clients are exposed to greater variations in performance [HBSHF19]. In contrast, using a single resolver, as would result from the client-based method in Section 4.1, promotes use of a persistent connection.

### 6.5. Debugging

The use of multiple resolvers may complicate debugging.

## 7. Further work

Should there be interest in the deployment of ideas laid out in this memo, further work is needed. There would have to be ways to configure systems to use multiple resolvers, including for instance:

- o Central configuration mechanisms to enable the use of multiple resolvers, perhaps through usual network configuration mechanisms or choices made by applications using resolver services directly. It may also be necessary to employ discovery mechanisms, such as, e.g., [I-D.schinazi-httpbis-doh-preference-hints] or [I-D.pauly-dprive-adaptive-dns-privacy] (but see Section 3)
- o Mechanisms to allow both failover to working resolvers when a resolver is unreachable,
- o Additional testing for potential operational issues discussed in Section 2 would be beneficial.

Finally, more work is needed to determine factors other than privacy that could motivate having queries routed to the same resolver. The choice between different approaches is often a combination of several factors, and privacy is only one of those factors.

## 8. References

### 8.1. Normative References

- [DNS] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [DNSSEC] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <<https://www.rfc-editor.org/info/rfc4033>>.
- [DOH] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [DOT] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [PMON] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.

### 8.2. Informative References

- [DBOUND] Levine, J., "Publishing Organization Boundaries in the DNS", draft-levine-dbound-dns-03 (work in progress), April 2019.
- [HBSHF19] Austin Hounsell, ., Kevin Borgolte, ., Paul Schmidt, ., Jordan Holland, ., and . Nick Feamster, "Analyzing the Costs (and Benefits) of DNS, DoT, and DoH for the Modern Web", ANRW '19, July 22, 2019, Montreal, QC, Canada, n.d., <<https://dl.acm.org/doi/10.1145/3340301.3341129>>.
- [I-D.pauly-dprive-adaptive-dns-privacy] Kinnear, E., Pauly, T., Wood, C., and P. McManus, "Adaptive DNS: Improving Privacy of Name Resolution", draft-pauly-dprive-adaptive-dns-privacy-01 (work in progress), November 2019.

- [I-D.schinazi-httpbis-doh-preference-hints]  
Schinazi, D., Sullivan, N., and J. Kipp, "DoH Preference Hints for HTTP", draft-schinazi-httpbis-doh-preference-hints-01 (work in progress), January 2020.
- [MSCVUS] Wikipedia, ., "Microsoft Corp. v. United States", [https://en.wikipedia.org/wiki/Microsoft\\_Corp.\\_v.\\_United\\_States](https://en.wikipedia.org/wiki/Microsoft_Corp._v._United_States) , n.d..

### 8.3. URIs

- [1] <https://publicsuffix.org/>
- [2] <https://github.com/mozilla-services/shavar-prod-lists/blob/master/disconnect-entitylist.json>
- [3] <https://github.com/krgovind/first-party-sets>

### Appendix A. Acknowledgements

The authors would like to thank Christian Huitema, Ari Keraenen, Mark Nottingham, Stephen Farrell, Gonzalo Camarillo, Mirja Kuehlewind, David Allan, Daniel Migault, Goran AP Eriksson, Christopher Wood, and many others for interesting discussions in this problem space.

### Authors' Addresses

Jari Arkko  
Ericsson

Email: [jari.arkko@piuha.net](mailto:jari.arkko@piuha.net)

Martin Thomson  
Mozilla

Email: [martin.thomson@gmail.com](mailto:martin.thomson@gmail.com)

Ted Hardie  
Google

Email: [ted.ietf@gmail.com](mailto:ted.ietf@gmail.com)

ADD  
Internet-Draft  
Intended status: Standards Track  
Expires: July 26, 2021

M. Boucadair  
Orange  
T. Reddy  
McAfee  
D. Wing  
Citrix  
N. Cook  
Open-Xchange  
T. Jensen  
Microsoft  
January 22, 2021

DHCP and Router Advertisement Options for Encrypted DNS Discovery  
draft-btw-add-home-12

Abstract

The document specifies new DHCP and IPv6 Router Advertisement options to discover encrypted DNS servers (e.g., DNS-over-HTTPS, DNS-over-TLS, DNS-over-QUIC). Particularly, it allows to learn an authentication domain name together with a list of IP addresses and a port number to reach such encrypted DNS servers. The discovery of DNS-over-HTTPS URI Templates is also discussed.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 26, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Overview and Rationale . . . . .	4
4. DHCPv6 Encrypted DNS Options . . . . .	6
4.1. Encrypted DNS ADN Option . . . . .	6
4.2. Encrypted DNS Address Option . . . . .	7
4.3. DHCPv6 Client Behavior . . . . .	9
5. DHCPv4 Encrypted DNS Option . . . . .	9
5.1. Encrypted DNS Option . . . . .	9
5.2. DHCPv4 Client Behavior . . . . .	11
6. IPv6 RA Encrypted DNS Options . . . . .	11
6.1. Encrypted DNS ADN Option . . . . .	12
6.2. Encrypted DNS Address Option . . . . .	13
7. DoH URI Templates . . . . .	14
8. Hosting Encrypted DNS Forwarder in Local Networks . . . . .	16
8.1. Managed CPEs . . . . .	16
8.1.1. DNS Forwarders . . . . .	16
8.1.2. ACME . . . . .	16
8.1.3. Auto-Upgrade Based on Domains and their Subdomains . . . . .	16
8.2. Unmanaged CPEs . . . . .	17
9. Legacy CPEs . . . . .	18
10. Security Considerations . . . . .	18
10.1. Spoofing Attacks . . . . .	18
10.2. Deletion Attacks . . . . .	19
10.3. Passive Attacks . . . . .	19
10.4. Wireless Security – Authentication Attacks . . . . .	20
11. IANA Considerations . . . . .	20
11.1. Encrypted DNS Flag Bits . . . . .	20
11.2. DHCPv6 Options . . . . .	21
11.3. DHCPv4 Option . . . . .	21
11.4. Neighbor Discovery Options . . . . .	21
12. Acknowledgements . . . . .	22
13. Contributing Authors . . . . .	22
14. References . . . . .	22
14.1. Normative References . . . . .	22
14.2. Informative References . . . . .	23

Appendix A. Sample Target Deployment Scenarios . . . . .	26
A.1. Managed CPEs . . . . .	27
A.1.1. Direct DNS . . . . .	28
A.1.2. Proxied DNS . . . . .	29
A.2. Unmanaged CPEs . . . . .	30
A.2.1. ISP-facing Unmanaged CPEs . . . . .	30
A.2.2. Internal Unmanaged CPEs . . . . .	30
Appendix B. Make Use of Discovered Encrypted DNS Servers . . . .	31
Authors' Addresses . . . . .	32

## 1. Introduction

This document focuses on the support of encrypted DNS such as DNS-over-HTTPS (DoH) [RFC8484], DNS-over-TLS (DoT) [RFC7858], or DNS-over-QUIC (DoQ) [I-D.ietf-dprive-dnsquic] in local networks.

In particular, the document specifies how a local encrypted DNS server can be discovered and used by connected hosts by means of DHCP [RFC2132], DHCPv6 [RFC8415], and IPv6 Router Advertisement (RA) [RFC4861] options. These options are designed to convey the following information: the DNS Authentication Domain Name (ADN), a list of IP addresses, and optionally a port number. The discovery of DoH URI Templates is discussed in Section 7.

Sample target deployment scenarios are discussed in Appendix A; both managed and unmanaged Customer Premises Equipment (CPEs) are covered. It is out of the scope of this document to provide an exhaustive inventory of deployments where Encrypted DNS Options (Sections 4, 5, and 6) can be used.

Considerations related to hosting a DNS forwarder in a local network are described in Section 8.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document makes use of the terms defined in [RFC8499]. The following additional terms are used:

Do53: refers to unencrypted DNS.

Encrypted DNS: refers to a scheme where DNS exchanges are transported over an encrypted channel. Examples of encrypted DNS



are DNS-over-TLS (DoT) [RFC7858], DNS-over-HTTPS (DoH) [RFC8484], or DNS-over-QUIC (DoQ) [I-D.ietf-dprive-dnsquic].

Managed CPE: refers to a CPE that is managed by an Internet Service Providers (ISP).

Unmanaged CPE: refers to a CPE that is not managed by an ISP.

DHCP: refers to both DHCPv4 and DHCPv6.

### 3. Overview and Rationale

This document describes how a DNS client can discover a local encrypted DNS server(s) using DHCP (Sections 4 and 5) and Neighbor Discovery protocol (Section 6).

As reported in Section 1.7.2 of [RFC6125]:

Some certification authorities issue server certificates based on IP addresses, but preliminary evidence indicates that such certificates are a very small percentage (less than 1%) of issued certificates.

In order to allow for PKIX-based authentication between a DNS client and an encrypted DNS server while accommodating the current best practices for issuing certificates, this document allows for configuring an authentication domain name to be presented as a reference identifier for DNS authentication purposes.

To avoid adding a dependency on another server to resolve the ADN, the options return a list of IP addresses to locate the encrypted DNS server. In the various scenarios sketched in Appendix A, encrypted DNS servers may terminate on the same IP address or distinct IP addresses. Terminating encrypted DNS servers on the same or distinct IP addresses is deployment specific. It is RECOMMENDED to return both the ADN and a list of IP addresses to a requesting host.

Note that in order to optimize the size of discovery messages when all servers terminate on the same IP address, a host may rely upon the discovery mechanisms specified in [RFC2132][RFC3646][RFC8106] to retrieve a list of IP addresses to reach their DNS servers. Nevertheless, this approach requires a client that supports more than one encrypted DNS to probe that list of IP addresses. To avoid such probing, the options defined in the following sections associate an IP address with an encrypted DNS type. No probing is required in such design.

A list of IP addresses to reach an encrypted DNS server can be returned in the option to accommodate current deployments relying upon primary and backup servers. Whether one IP address or more are returned in an option is deployment specific. For example, a router embedding a recursive server or forwarder has to include one single IP address pointing to one of its LAN-facing interfaces. This address can be a private IPv4 address, a link-local address, a Unique Local IPv6 unicast Address (ULA), or a Global Unicast Address (GUA).

If more than one IP address are to be returned in an Encrypted DNS server option, these addresses are ordered in the preference for use by the client.

Because DoT and DoQ may make use of customized port numbers instead of default ones, the Encrypted DNS server options are designed to return alternate port numbers.

Some ISPs rely upon external resolvers (e.g., outsourced service or public resolvers); these ISPs provide their customers with the IP addresses of these resolvers. These addresses are typically configured on CPEs using dedicated management tools. Likewise, users can modify the default DNS configuration of their CPEs (e.g., supplied by their ISP) to configure their favorite DNS servers. This document permits such deployments.

If the encrypted DNS is discovered by a host using both RA and DHCP, the rules discussed in Section 5.3.1 of [RFC8106] MUST be followed.

The DNS client establishes an encrypted DNS session with the discovered DNS IP address(es) and port number, and uses the mechanism discussed in Section 8 of [RFC8310] to authenticate the DNS server certificate using the authentication domain name conveyed in the encrypted DNS options.

Devices may be connected to multiple networks; each providing their own DNS configuration using the discovery mechanisms specified in this document. Nevertheless, it is out of the scope of this specification to discuss DNS selection of multi-interface devices. The reader may refer to [RFC6731] for a discussion of issues and an example of DNS server selection for multi-interfaced devices.

DHCP/RA options to discover encrypted DNS servers (including, DoH URI Templates should the WG pursue that approach pending feedback) takes precedence over DEER [I-D.pauly-add-deer] since DEER uses unencrypted DNS to an external DNS resolver, which is susceptible to both internal and external attacks whereas DHCP/RA is only vulnerable to internal attacks.

#### 4. DHCPv6 Encrypted DNS Options

This section defines two DHCPv6 options: DHCPv6 Encrypted DNS ADN option (Section 4.1) and DHCPv6 Encrypted DNS Address option (Section 4.2).

##### 4.1. Encrypted DNS ADN Option

The DHCPv6 Encrypted DNS ADN option is used to configure an authentication domain name of the encrypted DNS server. The format of this option is shown in Figure 1.

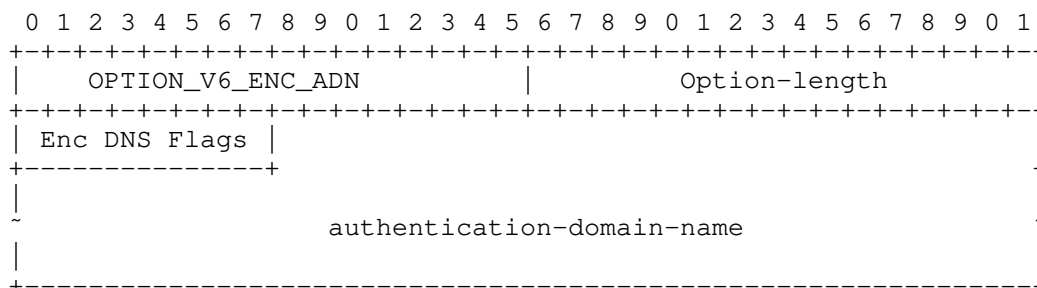


Figure 1: DHCPv6 Encrypted DNS ADN Option

The fields of the option shown in Figure 1 are as follows:

Option-code: OPTION\_V6\_ENC\_ADN (TBA1, see Section 11.2)

Option-length: Length of the enclosed data in octets.

Enc DNS Flags (Encrypted DNS Flags): Indicates the type(s) of the encrypted DNS server conveyed in this attribute. The format of this 8-bit field is shown in Figure 2.

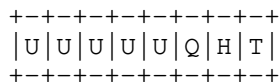


Figure 2: Encrypted DNS Flags Field

T: If set, this bit indicates that the server supports DoT [RFC7858].

H: If set, this bit indicates that the server supports DoH [RFC8484].

Q: If set, this bit indicates that the server supports DoQ [I-D.ietf-dprive-dnssoquic].

U: Unassigned bits. These bits MUST be unset by the sender. Associating a meaning with an unassigned bit can be done as per Section 11.1.

In a request, these bits are assigned to indicate the requested encrypted DNS server type(s) by the client. In a response, these bits are set as a function of the encrypted DNS supported by the server and the requested encrypted DNS server type(s).

To keep the packet small, if more than one encrypted DNS type (e.g., both DoH and DoT) are to be returned to a requesting client and the same ADN is used for these types, the corresponding bits must be set in the 'Encrypted DNS Types' field of the same option instance in a response. For example, if the client requested DoH and DoT and the server supports both with the same ADN, then both T and H bits must be set.

authentication-domain-name: A fully qualified domain name of the encrypted DNS server. This field is formatted as specified in Section 10 of [RFC8415].

An example of the authentication-domain-name encoding is shown in Figure 3. This example conveys the FQDN "doh1.example.com.", and the resulting Option-length field is 18.

0x04	d	o	h	1	0x07	e	x	a
m	p	l	e	0x03	c	o	m	0x00

Figure 3: An Example of the DNS authentication-domain-name Encoding

#### 4.2. Encrypted DNS Address Option

The DHCPv6 Encrypted DNS Address option is used to configure a list of IP addresses and a port number of the encrypted DNS server. The format of this option is shown in Figure 4.

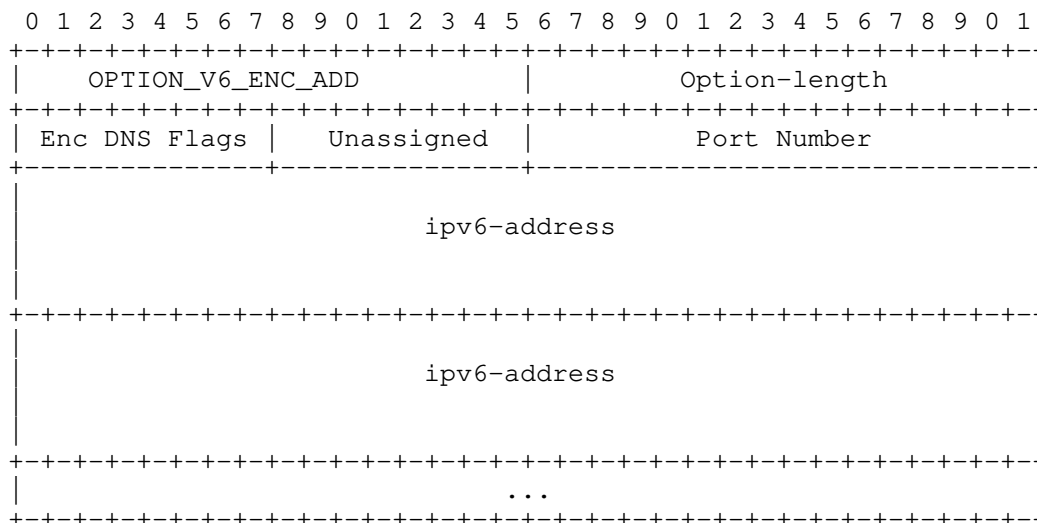


Figure 4: DHCPv6 Encrypted DNS Address Option

The fields of the option shown in Figure 4 are as follows:

Option-code: OPTION\_V6\_ENC\_ADD (TBA2, see Section 11.2)

Option-length: Length of the enclosed data in octets.

Enc DNS Flags (Encrypted DNS Flags): Indicates the type(s) of the encrypted DNS server conveyed in this attribute. The format of this 8-bit field is shown in Figure 2. In a request, these bits are set to indicate the requested encrypted DNS server type(s) by the client. In a response, these bits are set as a function of the encrypted DNS supported by the server and the requested encrypted DNS server type(s).

Unassigned: These bits MUST be unset by the sender. Associating a meaning with an unassigned bit can be done via Standards Action [RFC8126].

Port Number: If not null, it indicates the port number to be used for the encrypted DNS. If this field is set to zero, this indicates that default port numbers should be used. As a reminder, the default port number is 853 for DoT and 443 for DoH.

ipv6-address(es): Indicates one or more IPv6 addresses to reach the encrypted DNS server. An address can be link-local, ULA, or GUA.

Multiple instances of `OPTION_V6_ENC_ADN` (or `OPTION_V6_ENC_ADD`) may be returned to a DHCPv6 client; each pointing to a distinct encrypted DNS server type.

If more than one encrypted DNS server types is supported on the same IP address and default port numbers are used, one instance of `OPTION_V6_ENC_ADD` option with the appropriate bits set in "Encr DNS Types" field should be returned by the DHCP server.

#### 4.3. DHCPv6 Client Behavior

To discover an encrypted DNS server, the DHCPv6 client MUST include `OPTION_V6_ENC_ADN` and `OPTION_V6_ENC_ADD` in an Option Request Option (ORO), as in Sections 18.2.1, 18.2.2, 18.2.4, 18.2.5, 18.2.6, and 21.7 of [RFC8415]. The DHCPv6 client sets the Encrypted DNS Types field to the requested encrypted DNS server type(s).

If the DHCPv6 client requested more than one encrypted DNS server type, the DHCP client MUST be prepared to receive multiple `OPTION_V6_ENC_ADN` (or `OPTION_V6_ENC_ADD`) options; each option is to be treated as a separate encrypted DNS server.

The DHCPv6 client MUST silently discard multicast and host loopback addresses conveyed in `OPTION_V6_ENC_ADD`.

### 5. DHCPv4 Encrypted DNS Option

#### 5.1. Encrypted DNS Option

The DHCPv4 Encrypted DNS option is used to configure an authentication domain name, a list of IP addresses, and a port number of the encrypted DNS server. The format of this option is illustrated in Figure 5.

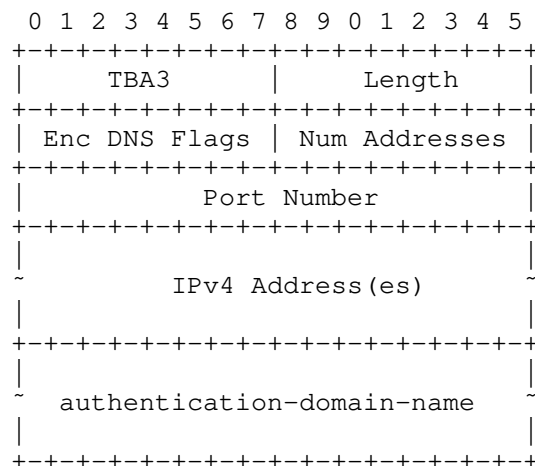


Figure 5: DHCPv4 Encrypted DNS Option

The fields of the option shown in Figure 5 are as follows:

Code: `OPTION_V4_ENC_DNS` (TBA3, see Section 11.3).

Length: Length of the enclosed data in octets.

Enc DNS Flags (Encrypted DNS Flags): Indicates the type(s) of the encrypted DNS server conveyed in this attribute. The format of this field is shown in Figure 2.

Num Addresses: Indicates the number of included IPv4 addresses.

Port Number: If not null, it indicates the port number to be used for the encrypted DNS. A null value indicates that default port numbers are used. As a reminder, the default port number is 853 for DoT and 443 for DoH.

IPv4 Address(es): Indicates one or more IPv4 addresses to reach the encrypted DNS server. Both private and public IPv4 addresses can be included in this field. The format of this field is shown in Figure 6. This format assumes that an IPv4 address is encoded as a1.a2.a3.a4.

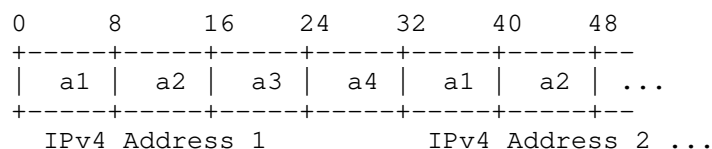


Figure 6: Format of the IPv4 Addresses Field

authentication-domain-name: The domain name of the encrypted DNS server. This field is formatted as specified in Section 10 of [RFC8415]. The format of this field is shown in Figure 7. The values s1, s2, s3, etc. represent the domain name labels in the domain name encoding.

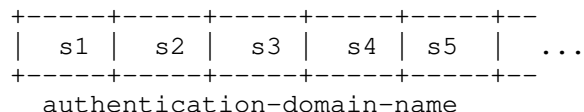


Figure 7: Format of the Authentication Domain Name Field

OPTION\_V4\_ENC\_DNS is a concatenation-requiring option. As such, the mechanism specified in [RFC3396] MUST be used if OPTION\_V4\_ENC\_DNS exceeds the maximum DHCPv4 option size of 255 octets.

## 5.2. DHCPv4 Client Behavior

To discover an encrypted DNS server, the DHCPv4 client requests the Encrypted DNS server by including OPTION\_V4\_ENC\_DNS in a Parameter Request List option [RFC2132]. The DHCPv4 client sets the Encrypted DNS Types field to the requested encrypted DNS server.

If the DHCPv4 client requested more than one encrypted DNS server type, the DHCPv4 client MUST be prepared to receive multiple DHCP OPTION\_V4\_ENC\_DNS options; each option is to be treated as a separate encrypted DNS server.

The DHCPv4 client MUST silently discard multicast and host loopback addresses conveyed in OPTION\_V4\_ENC\_DNS.

## 6. IPv6 RA Encrypted DNS Options

This section defines two Neighbor Discovery [RFC4861]: IPv6 Router Advertisement (RA) Encrypted DNS ADN option (Section 6.1) and IPv6 RA Encrypted DNS Address option (Section 6.2). These options are useful in contexts similar to those discussed in Section 1.1 of [RFC8106].



### 6.1. Encrypted DNS ADN Option

The IPv6 RA Encrypted DNS ADN option is used to configure an authentication domain name of the encrypted DNS server. The format of this option is illustrated in Figure 8.

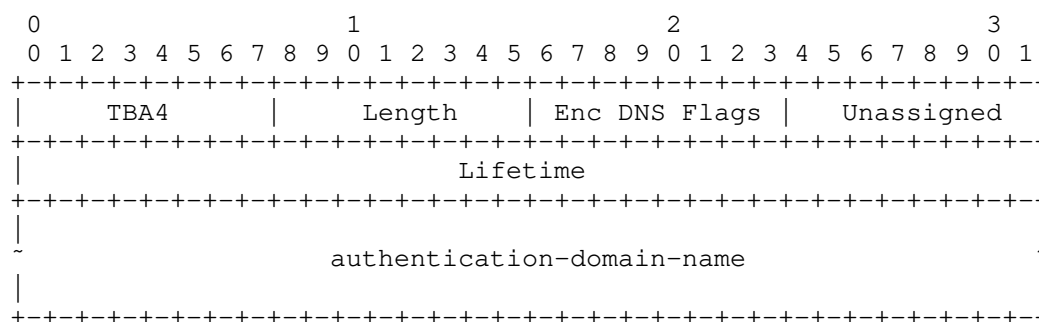


Figure 8: RA Encrypted DNS ADN Option

The fields of the option shown in Figure 8 are as follows:

**Type:** 8-bit identifier of the Encrypted DNS Option as assigned by IANA (TBA4, see Section 11.4).

**Length:** 8-bit unsigned integer. The length of the option (including the Type and Length fields) is in units of 8 octets.

**Enc DNS Flags (Encrypted DNS Flags):** Indicates the type(s) of the encrypted DNS server conveyed in this attribute. The format of this field is shown in Figure 2.

**Unassigned:** This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

**Lifetime:** 32-bit unsigned integer. The maximum time in seconds (relative to the time the packet is received) over which the discovered Authentication Domain Name is valid.

The value of Lifetime SHOULD by default be at least 3 \* MaxRtrAdvInterval, where MaxRtrAdvInterval is the maximum RA interval as defined in [RFC4861].

A value of all one bits (0xffffffff) represents infinity.

A value of zero means that this Authentication Domain Name MUST no longer be used.

authentication Domain Name: The domain name of the encrypted DNS server. This field is formatted as specified in Section 10 of [RFC8415].

This field MUST be padded with zeros so that its size is a multiple of 8 octets.

## 6.2. Encrypted DNS Address Option

The IPv6 RA Encrypted DNS Address option is used to configure a port number and a list of IPv6 addresses of the encrypted DNS server. The format of this option is illustrated in Figure 9. All of the addresses share the same Lifetime value. Similar to [RFC8106], if it is desirable to have different Lifetime values per IP address, multiple Encrypted DNS Address options may be used.

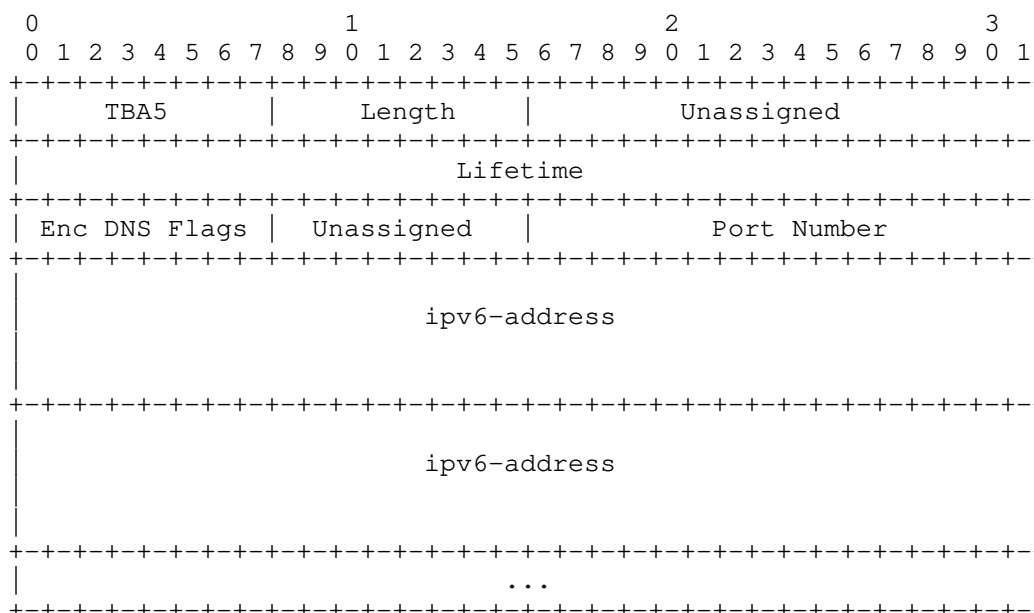


Figure 9: RA Encrypted DNS Address Option

The fields of the RA Encrypted DNS Address option shown in Figure 9 are as follows:

Type: 8-bit identifier of the Encrypted DNS Address Option as assigned by IANA (TBA5, see Section 11.4).

Length: 8-bit unsigned integer. The length of the option (including the Type and Length fields) is in units of 8 octets.

Unassigned: This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Lifetime: 32-bit unsigned integer. The maximum time in seconds (relative to the time the packet is received) over which the discovered encrypted DNS IPv6 addresses are valid.

The value of Lifetime SHOULD by default be at least  $3 * \text{MaxRtrAdvInterval}$ , where MaxRtrAdvInterval is the maximum RA interval as defined in [RFC4861].

A value of all one bits (0xffffffff) represents infinity.

A value of zero means that these IPv6 addresses MUST no longer be used.

Enc DNS Flags (Encrypted DNS Flags): Indicates the type(s) of the encrypted DNS server conveyed in this attribute. The format of this field is shown in Figure 2.

Port Number: If not null, it indicates the port number to be used for the encrypted DNS. A null value indicates that default port numbers must be used. As a reminder, the default port number is 853 for DoT and 443 for DoH.

ipv6-address(es): One or more IPv6 addresses of the encrypted DNS server. An address can be link-local, ULA, or GUA.

## 7. DoH URI Templates

DoH servers may support more than one URI Template [RFC8484]. Also, if the resolver hosts several DoH services (e.g., no-filtering, blocking adult content, blocking malware), these services can be discovered as templates. The following discusses a mechanism for a DoH client to retrieve the list of supported templates by a DoH server.

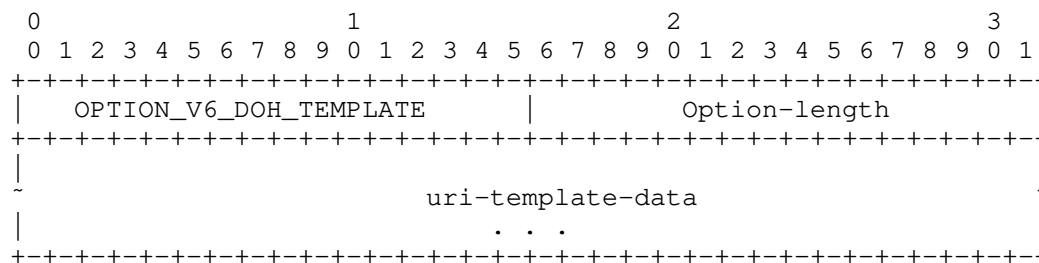
Upon discovery of a DoH resolver (Sections 4, 5, and 6), the DoH client may contact that DoH resolver to retrieve the list of supported DoH services using DEER [I-D.pauly-add-deer]. This will allow the client to discover the resolver's supported DoH templates or DoH resolvers that the discovered resolver designates using DNS SVCB queries [I-D.schwartz-svcb-dns]. The designated DoH resolvers and DoH resolver discovered using DHCP/RA may be hosted on the same or distinct IP addresses.

Let's suppose that a host has discovered an encrypted DNS server that is DoH-capable. The host has also discovered the following information:

- o ADN: doh.example.com
- o Locator: 2001:db8:1::1

The client will use DEER [I-D.pauly-add-deer] to discover the DoH templates supported by the DNS server at the Locator (2001:db8:1::1). In addition to the checks included in DEER, clients should verify the ADN (doh.example.com) is valid for the certificate provided by the DoH resolver. However, the IP address of the DEER-discovered resolver may differ from the Locator field value. This will allow the ISP to offer different DoH services to the endpoints attached to local networks.

Alternatively, dedicated DHCP/RA options may be defined to convey an URI template in order to avoid additional network traffic to bootstrap DoH configuration. An example of the format of such an option is depicted in Figure 10.



Each instance of the uri-template-data is formatted as follows:

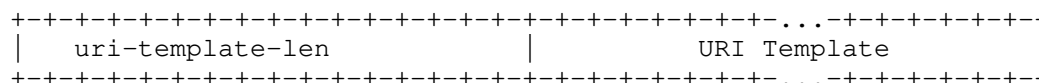


Figure 10: Example of a DHCPv6 URI Template Option

Note: More feedback from the WG is needed to decide which approach(es) to follow.

How a DoH client makes use of the configured DoH services is out of the scope of this document.

## 8. Hosting Encrypted DNS Forwarder in Local Networks

This section discusses some deployment considerations (not recommendations) to host an encrypted DNS forwarder within a local network.

### 8.1. Managed CPEs

The section discusses mechanisms that can be used to host an encrypted DNS forwarder in a managed CPE (Appendix A.1).

#### 8.1.1. DNS Forwarders

The managed CPE should support a configuration parameter to instruct the CPE whether it has to relay the encrypted DNS server received from the ISP's network or has to announce itself as a forwarder within the local network. The default behavior of the CPE is to supply the encrypted DNS server received from the ISP's network.

#### 8.1.2. ACME

The ISP can assign a unique FQDN (e.g., "cpe1.example.com") and a domain-validated public certificate to the encrypted DNS forwarder hosted on the CPE. Automatic Certificate Management Environment (ACME) [RFC8555] can be used by the ISP to automate certificate management functions such as domain validation procedure, certificate issuance and certificate revocation.

#### 8.1.3. Auto-Upgrade Based on Domains and their Subdomains

If the ADN conveyed in DHCP/RA (Sections 4, 5, and 6) is preconfigured in popular Oses or browsers as a verified resolver and the auto-upgrade (Appendix B) is allowed for both the preconfigured ADN and its sub-domains, the encrypted DNS client will learn the local encrypted DNS forwarder using DHCP/RA and auto-upgrade because the preconfigured ADN would match the subjectAltName value in the server certificate. For example, if the preconfigured ADN is "\*.example.com" and the discovered encrypted DNS forwarder is "cpe1.example.com", auto-upgrade will take place.

In this case, the CPE can communicate the ADN of the local DoH forwarder (Section 8.1.2) to internal hosts using DHCP/RA (Sections 4, 5, and 6).

Let's suppose that "\*.example.net" is preconfigured as a verified resolved in the browser or OS. If the encrypted DNS client discovers a local forwarder "cpe1-internal.example.net", the encrypted DNS client will auto-upgrade because the preconfigured ADN would match

subjectAltName value "cpe1-internal.example.net" of type dNSName. As shown in Figure 11, the auto-upgrade to a rogue server advertising "rs.example.org" will fail because it does not match "\*.example.net".

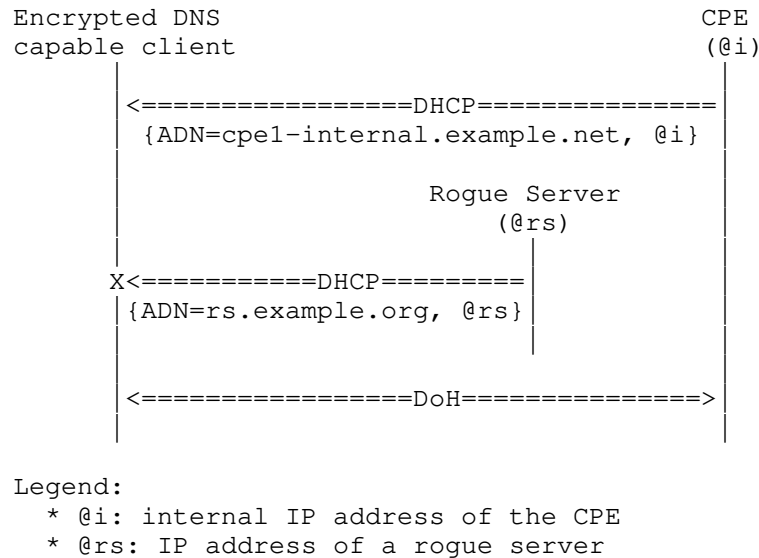


Figure 11: A Simplified Example of Auto-upgrade based on Subdomains

## 8.2. Unmanaged CPEs

The approach specified in Section 8.1 does not apply for hosting a DNS forwarder in an unmanaged CPE.

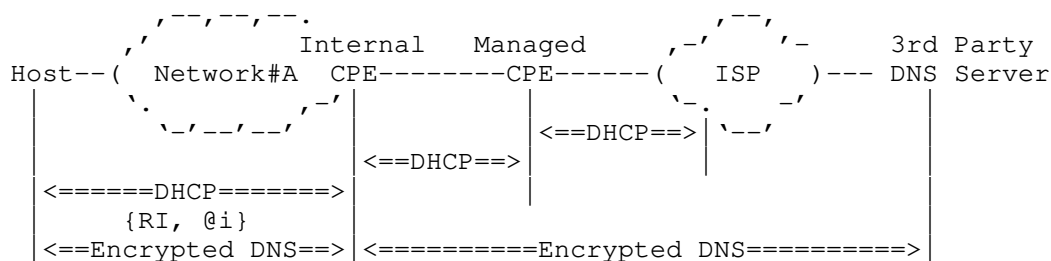
The unmanaged CPE administrator (referred to as administrator) can host an encrypted DNS forwarder on the unmanaged CPE. This assumes the following:

- o The encrypted DNS server certificate is managed by the entity in-charge of hosting the encrypted DNS forwarder.

Alternatively, a security service provider can assign a unique FQDN to the CPE. The encrypted DNS forwarder will act like a private encrypted DNS server only be accessible from within the the local network.

- o The encrypted DNS forwarder will either be configured to use the ISP's or a 3rd party encrypted DNS server.
- o The unmanaged CPE will advertise the encrypted DNS forwarder ADN using DHCP/RA to internal hosts.

Figure 12 illustrates an example of an unmanaged CPE hosting a forwarder which connects to a 3rd party encrypted DNS server. In this example, the DNS information received from the managed CPE (and therefore from the ISP) is ignored by the Internal CPE hosting the forwarder.



Legend:

\* @i: IP address of the DNS forwarder hosted in the Internal CPE.

Figure 12: Example of an Internal CPE Hosting a Forwarder

## 9. Legacy CPEs

Hosts serviced by legacy CPEs that can't be upgraded to support the options defined in Sections 4, 5, and 6 won't be able to learn the encrypted DNS server hosted by the ISP, in particular. If the ADN is not discovered using DHCP/RA, such hosts will have to fallback to use DEER as defined in [I-D.pauly-add-deer] to discover the encrypted DNS server and to retrieve the list of supported DoH services using the SVCB RRtype [I-D.schwartz-svc-b-dns] without verifying the hostname of discovered templates with the ADN. Other guidance in DEER relating to resolver verification must be followed in this case. This will prevent an unencrypted resolver on a local address from referring to an encrypted resolver at a different address without an out-of-band configuration in the client beyond the scope of this document or DEER.

## 10. Security Considerations

### 10.1. Spoofing Attacks

DHCP/RA messages are not encrypted or protected against modification within the LAN. Unless mitigated (described below), the content of DHCP and RA messages can be spoofed or modified by active attackers, such as compromised devices within the local network. An active attacker (Section 3.3 of [RFC3552]) can spoof the DHCP/RA response to provide the attacker's Encrypted DNS server. Note that such an

attacker can launch other attacks as discussed in Section 22 of [RFC8415]. The attacker can get a domain name with a domain-validated public certificate from a CA and host an Encrypted DNS server. Also, an attacker can use a public IP address and get an 'IP address'-validated public certificate from a CA to host an Encrypted DNS server.

Attacks of spoofed or modified DHCP responses and RA messages by attackers within the local network may be mitigated by making use of the following mechanisms:

- o DHCPv6-Shield described in [RFC7610], the CPEs discards DHCP response messages received from any local endpoint.
- o RA-Guard described in [RFC7113], the CPE discards RAs messages received from any local endpoint.
- o Source Address Validation Improvement (SAVI) solution for DHCP described in [RFC7513], the CPE filters packets with forged source IP addresses.

Encrypted DNS sessions with rogue servers that spoof the IP address of a DNS server will fail because the DNS client will fail to authenticate that rogue server based upon PKIX authentication [RFC6125], particularly the authentication domain name in the Encrypted DNS Option. DNS clients that ignore authentication failures and accept spoofed certificates will be subject to attacks (e.g., redirect to malicious servers, intercept sensitive data).

Encrypted DNS connections received from outside the local network MUST be discarded by the encrypted DNS forwarder in the CPE. This behavior adheres to REQ#8 in [RFC6092]; it MUST apply for both IPv4 and IPv6.

## 10.2. Deletion Attacks

If the DHCP responses or RAs are dropped by the attacker, the client can fallback to use a preconfigured encrypted DNS server. However, the use of policies to select servers is out of the scope of this document.

Note that deletion attack is not specific to DHCP/RA.

## 10.3. Passive Attacks

A passive attacker (Section 3.2 of [RFC3552]) can identify a host is using DHCP/RA to discover an encrypted DNS server and can infer that



host is capable of using DoH/DoT/DoQ to encrypt DNS messages. However, a passive attacker cannot spoof or modify DHCP/RA messages.

#### 10.4. Wireless Security - Authentication Attacks

Wireless LAN (WLAN) as frequently deployed in local networks (e.g., home networks) is vulnerable to various attacks (e.g., [Evil-Twin], [Krack], [Dragonblood]). Because of these attacks, only cryptographically authenticated communications are trusted on WLANs. This means information provided by such networks via DHCP, DHCPv6, or RA (e.g., NTP server, DNS server, default domain) are untrusted because DHCP and RA are not authenticated.

If the pre-shared-key is the same for all clients that connect to the same WLAN, the shared key will be available to all nodes, including attackers, so it is possible to mount an active on-path attack. Man-in-the-middle attacks are possible within local networks because such WLAN authentication lacks peer entity authentication.

This leads to the need for provisioning unique credentials for different clients. Endpoints can be provisioned with unique credentials (username and password, typically) provided by the local network administrator to mutually authenticate to the local WLAN Access Point (e.g., 802.1x Wireless User Authentication on OpenWRT [dot1x], EAP-pwd [RFC8146]). Not all of endpoint devices (e.g., IoT devices) support 802.1x supplicant and need an alternate mechanism to connect to the local network. To address this limitation, unique pre-shared keys can be created for each such device and WPA-PSK is used (e.g., [PSK]).

### 11. IANA Considerations

#### 11.1. Encrypted DNS Flag Bits

	1	2	3	4	5	6	7	8
	+	-	+	+	+	+	+	+
Encrypted DNS Types is a set of 8 flags:		U		U		U		U
		Q		H		T		
	+	+	+	+	+	+	+	+

where flag bits in positions 1-5 are for future assignment as additional flag bits.

This document requests IANA to create a new registry called "Encrypted DNS Types". The initial values of the registry are as follows:

Bit Position	Label	Description	Reference
1	U	Unassigned	
2	U	Unassigned	
3	U	Unassigned	
4	U	Unassigned	
5	U	Unassigned	
6	Q	DNS-over-QUIC (DoQ)	[ThisDocument]
7	H	DNS-over-HTTP (DoH)	[ThisDocument]
8	T	DNS-over-TLS (DoT)	[ThisDocument]

New flag bits are assigned via Standards Action [RFC8126].

#### 11.2. DHCPv6 Options

IANA is requested to assign the following new DHCPv6 Option Code in the registry maintained in [DHCPV6].

Value	Description	Client ORO	Singleton Option	Reference
TBA1	OPTION_V6_ENC_ADN	Yes	No	[ThisDocument]
TBA2	OPTION_V6_ENC_ADD	Yes	No	[ThisDocument]

#### 11.3. DHCPv4 Option

IANA is requested to assign the following new DHCP Option Code in the registry maintained in [BOOTP].

Tag	Name	Data Length	Meaning	Reference
TBA3	OPTION_V4_ENC_DNS	N	Encrypted DNS Server	[ThisDocument]

#### 11.4. Neighbor Discovery Options

IANA is requested to assign the following new IPv6 Neighbor Discovery Option type in the "IPv6 Neighbor Discovery Option Formats" sub-registry under the "Internet Control Message Protocol version 6 (ICMPv6) Parameters" registry maintained in [ND].

Type	Description	Reference
TBA4	DNS Encrypted DNS ADN Option	[ThisDocument]
TBA5	DNS Encrypted DNS Address Option	[ThisDocument]

## 12. Acknowledgements

Many thanks to Christian Jacquenet and Michael Richardson for the review.

Thanks to Stephen Farrell, Martin Thomson, Vittorio Bertola, Stephane Bortzmeyer, Ben Schwartz, and Iain Sharp for the comments.

Thanks to Mark Nottingham for the feedback on HTTP redirection.

The use of DHCP to retrieve an authentication domain name was discussed in Section 7.3.1 of [RFC8310] and [I-D.pusateri-dhc-dns-driu].

## 13. Contributing Authors

Nicolai Leymann  
Deutsche Telekom  
Germany

Email: n.leymann@telekom.de

## 14. References

### 14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <<https://www.rfc-editor.org/info/rfc2132>>.
- [RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, DOI 10.17487/RFC3396, November 2002, <<https://www.rfc-editor.org/info/rfc3396>>.

- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, DOI 10.17487/RFC8106, March 2017, <<https://www.rfc-editor.org/info/rfc8106>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

#### 14.2. Informative References

- [Auto-upgrade] The Unicode Consortium, "DoH providers: criteria, process for Chrome", <[docs.google.com/document/d/128i2YTV2C7T6Gr3I-81zlQ-\\_Lprnsp24qzy\\_20Z1Psw/edit](https://docs.google.com/document/d/128i2YTV2C7T6Gr3I-81zlQ-_Lprnsp24qzy_20Z1Psw/edit)>.
- [BOOTP] "BOOTP Vendor Extensions and DHCP Options", <<https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#options>>.
- [DHCPV6] "DHCPv6 Option Codes", <<https://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xhtml#dhcpv6-parameters-2>>.
- [dot1x] Cisco, "Basic 802.1x Wireless User Authentication", <<https://openwrt.org/docs/guide-user/network/wifi/wireless.security.8021x>>.
- [Dragonblood] The Unicode Consortium, "Dragonblood: Analyzing the Dragonfly Handshake of WPA3 and EAP-pwd", <<https://papers.mathyvanhoef.com/dragonblood.pdf>>.

- [Evil-Twin] The Unicode Consortium, "Evil twin (wireless networks)", <[https://en.wikipedia.org/wiki/Evil\\_twin\\_\(wireless\\_networks\)](https://en.wikipedia.org/wiki/Evil_twin_(wireless_networks))>.
- [I-D.ietf-dprive-dnsoquic] Huitema, C., Mankin, A., and S. Dickinson, "Specification of DNS over Dedicated QUIC Connections", draft-ietf-dprive-dnsoquic-01 (work in progress), October 2020.
- [I-D.ietf-v6ops-rfc7084-bis] Palet, J., "Basic Requirements for IPv6 Customer Edge Routers", draft-ietf-v6ops-rfc7084-bis-04 (work in progress), June 2017.
- [I-D.pauly-add-deer] Pauly, T., Kinnear, E., Wood, C., McManus, P., and T. Jensen, "Discovery of Equivalent Encrypted Resolvers", draft-pauly-add-deer-00 (work in progress), November 2020.
- [I-D.pusateri-dhc-dns-driu] Pusateri, T. and W. Toorop, "DHCPv6 Options for private DNS Discovery", draft-pusateri-dhc-dns-driu-00 (work in progress), July 2018.
- [I-D.schwartz-svcb-dns] Schwartz, B., "Service Binding Mapping for DNS Servers", draft-schwartz-svcb-dns-01 (work in progress), August 2020.
- [Krack] The Unicode Consortium, "Key Reinstallation Attacks", 2017, <<https://www.krackattacks.com/>>.
- [ND] "IPv6 Neighbor Discovery Option Formats", <<http://www.iana.org/assignments/icmpv6-parameters/icmpv6-parameters.xhtml#icmpv6-parameters-5>>.
- [PSK] Cisco, "Identity PSK Feature Deployment Guide", <[https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-5/b\\_Identity\\_PSK\\_Feature\\_Deployment\\_Guide.html](https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-5/b_Identity_PSK_Feature_Deployment_Guide.html)>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.

- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <<https://www.rfc-editor.org/info/rfc3646>>.
- [RFC6092] Woodyatt, J., Ed., "Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service", RFC 6092, DOI 10.17487/RFC6092, January 2011, <<https://www.rfc-editor.org/info/rfc6092>>.
- [RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, DOI 10.17487/RFC6125, March 2011, <<https://www.rfc-editor.org/info/rfc6125>>.
- [RFC6731] Savolainen, T., Kato, J., and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, DOI 10.17487/RFC6731, December 2012, <<https://www.rfc-editor.org/info/rfc6731>>.
- [RFC7113] Gont, F., "Implementation Advice for IPv6 Router Advertisement Guard (RA-Guard)", RFC 7113, DOI 10.17487/RFC7113, February 2014, <<https://www.rfc-editor.org/info/rfc7113>>.
- [RFC7513] Bi, J., Wu, J., Yao, G., and F. Baker, "Source Address Validation Improvement (SAVI) Solution for DHCP", RFC 7513, DOI 10.17487/RFC7513, May 2015, <<https://www.rfc-editor.org/info/rfc7513>>.
- [RFC7610] Gont, F., Liu, W., and G. Van de Velde, "DHCPv6-Shield: Protecting against Rogue DHCPv6 Servers", BCP 199, RFC 7610, DOI 10.17487/RFC7610, August 2015, <<https://www.rfc-editor.org/info/rfc7610>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8146] Harkins, D., "Adding Support for Salted Password Databases to EAP-pwd", RFC 8146, DOI 10.17487/RFC8146, April 2017, <<https://www.rfc-editor.org/info/rfc8146>>.

- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <<https://www.rfc-editor.org/info/rfc8499>>.
- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.
- [RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <<https://www.rfc-editor.org/info/rfc8555>>.
- [TR-069] The Broadband Forum, "CPE WAN Management Protocol", December 2018, <<https://www.broadband-forum.org/technical/download/TR-069.pdf>>.
- [TS.24008] 3GPP, "Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (Release 16)", December 2019, <<http://www.3gpp.org/DynaReport/24008.htm>>.

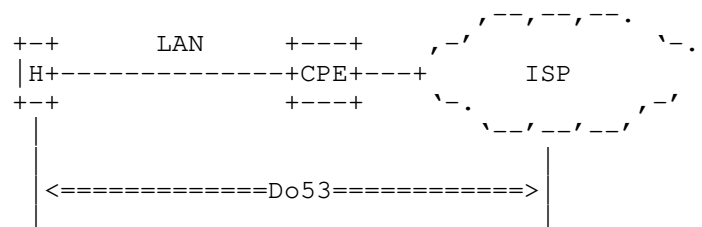
#### Appendix A. Sample Target Deployment Scenarios

Internet Service Providers (ISPs) traditionally provide DNS resolvers to their customers. To that aim, ISPs deploy the following mechanisms to advertise a list of DNS Recursive DNS server(s) to their customers:

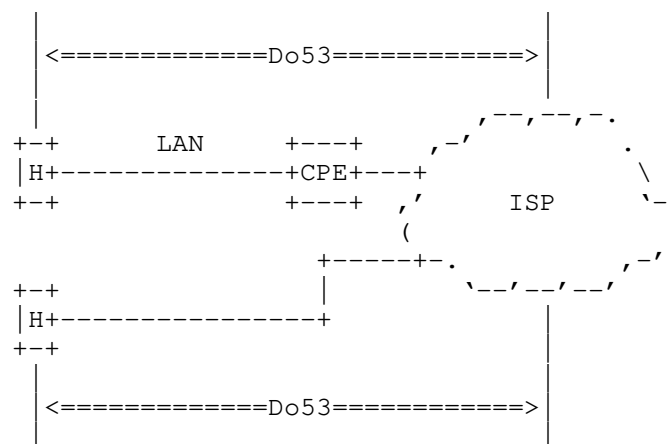
- o Protocol Configuration Options in cellular networks [TS.24008].
- o DHCPv4 [RFC2132] (Domain Name Server Option) or DHCPv6 [RFC8415][RFC3646] (OPTION\_DNS\_SERVERS).
- o IPv6 Router Advertisement [RFC4861][RFC8106] (Type 25 (Recursive DNS Server Option)).

The communication between a customer's device (possibly via Customer Premises Equipment (CPE)) and an ISP-supplied DNS resolver takes place by using cleartext DNS messages (Do53). Some examples are depicted in Figure 13. In the case of cellular networks, the cellular network will provide connectivity directly to a host (e.g., smartphone, tablet) or via a CPE. Do53 mechanisms used within the Local Area Network (LAN) are similar in both fixed and cellular CPE-based broadband service offerings.

(a) Fixed Networks



(b) Cellular Networks



Legend:

\* H: refers to a host.

Figure 13: Sample Legacy Deployments

#### A.1. Managed CPEs

This section focuses on CPEs that are managed by ISPs.



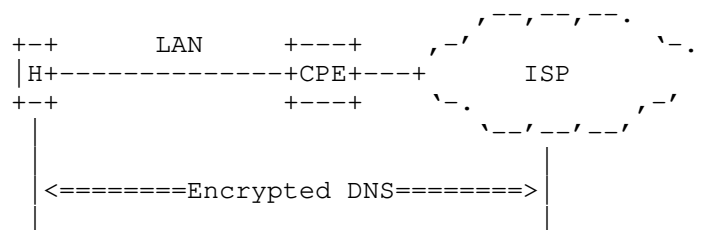
## A.1.1. Direct DNS

ISPs have developed an expertise in managing service-specific configuration information (e.g., CPE WAN Management Protocol [TR-069]). For example, these tools may be used to provision the DNS server's ADN to managed CPEs if an encrypted DNS is supported by a local network similar to what is depicted in Figure 14.

For example, DoH-capable (or DoT) clients establish the DoH (or DoT) session with the discovered DoH (or DoT) server.

The DNS client discovers whether the DNS server in the local network supports DoH/DoT/DoQ by using a dedicated field in the discovery message: Encrypted DNS Types (Sections 4, 5, and 6) .

(a) Fixed Networks



(b) Cellular Networks

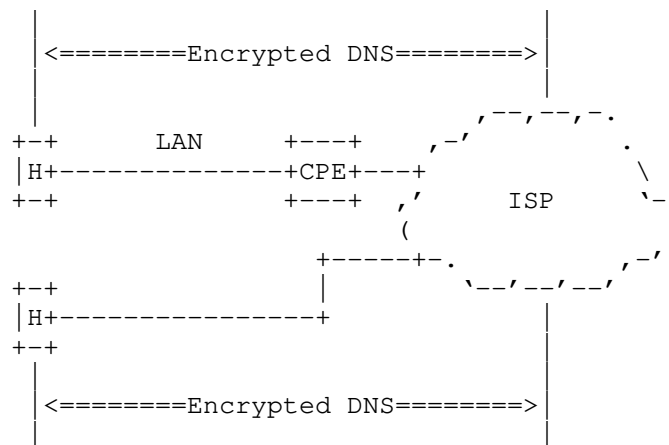


Figure 14: Encrypted DNS in the WAN

Figure 14 shows the scenario where the CPE relays the list of encrypted DNS servers it learns for the network by using mechanisms

like DHCP or a specific Router Advertisement message. In such context, direct encrypted DNS sessions will be established between a host serviced by a CPE and an ISP-supplied encrypted DNS server (see the example depicted in Figure 15 for a DoH/DoT-capable host).

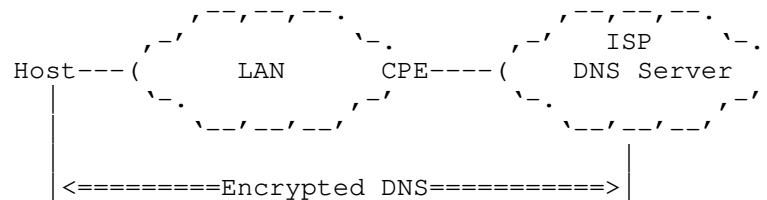


Figure 15: Direct Encrypted DNS Sessions

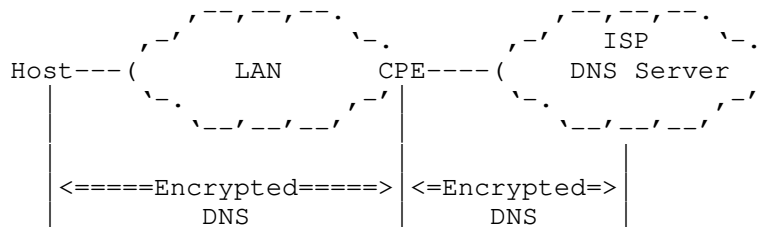
#### A.1.2. Proxied DNS

Figure 16 shows a deployment where the CPE embeds a caching DNS forwarder. The CPE advertises itself as the default DNS server to the hosts it serves. The CPE relies upon DHCP or RA to advertise itself to internal hosts as the default DoT/DoH/Do53 server. When receiving a DNS request it cannot handle locally, the CPE forwards the request to an upstream DoH/DoT/Do53 resolver. Such deployment is required for IPv4 service continuity purposes (e.g., Section 5.4.1 of [I-D.ietf-v6ops-rfc7084-bis]) or for supporting advanced services within a local network (e.g., malware filtering, parental control, Manufacturer Usage Description (MUD) [RFC8520] to only allow intended communications to and from an IoT device). When the CPE behaves as a DNS forwarder, DNS communications can be decomposed into two legs:

- o The leg between an internal host and the CPE.
- o The leg between the CPE and an upstream DNS resolver.

An ISP that offers encrypted DNS to its customers may enable encrypted DNS in one or both legs as shown in Figure 16. Additional considerations related to this deployment are discussed in Section 8.

(a)



(b)

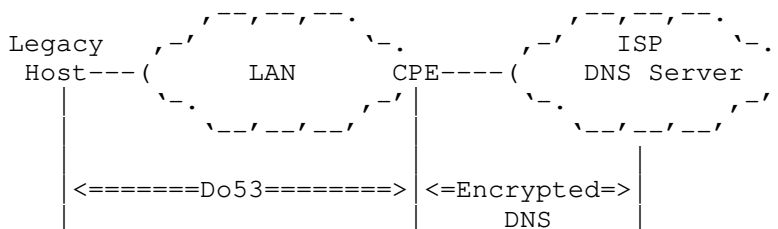


Figure 16: Proxied Encrypted DNS Sessions

## A.2. Unmanaged CPEs

### A.2.1. ISP-facing Unmanaged CPEs

Customers may decide to deploy unmanaged CPEs (assuming the CPE is compliant with the network access technical specification that is usually published by ISPs). Upon attachment to the network, an unmanaged CPE receives from the network its service configuration (including the DNS information) by means of, e.g., DHCP. That DNS information is shared within the LAN following the same mechanisms as those discussed in Appendix A.1. A host can thus establish DoH/DoT session with a DoH/DoT server similar to what is depicted in Figure 15 or Figure 16.

### A.2.2. Internal Unmanaged CPEs

Customers may also decide to deploy internal routers (called hereafter, Internal CPEs) for a variety of reasons that are not detailed here. Absent any explicit configuration on the internal CPE to override the DNS configuration it receives from the ISP-supplied CPE, an Internal CPE relays the DNS information it receives via DHCP/RA from the ISP-supplied CPE to connected hosts. Encrypted DNS sessions can be established by a host with the DNS servers of the ISP (see Figure 17).

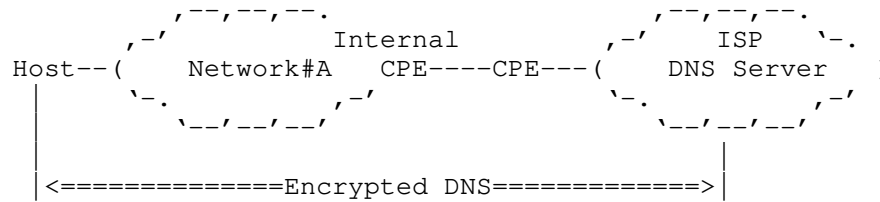


Figure 17: Direct Encrypted DNS Sessions with the ISP DNS Resolver (Internal CPE)

Similar to managed CPEs, a user may modify the default DNS configuration of an unmanaged CPE to use his/her favorite DNS servers instead. Encrypted DNS sessions can be established directly between a host and a 3rd Party DNS server (see Figure 18).

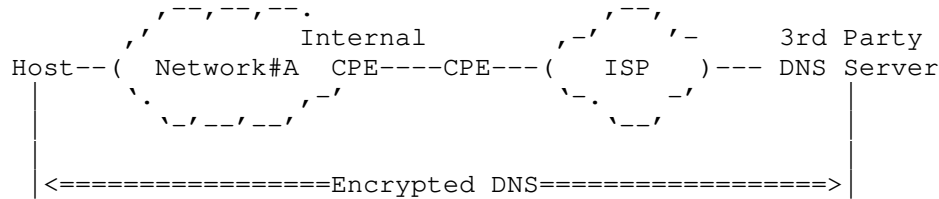


Figure 18: Direct Encrypted DNS Sessions with a Third Party DNS Resolver

Section 8.2 discusses considerations related to hosting a forwarder in the Internal CPE.

#### Appendix B. Make Use of Discovered Encrypted DNS Servers

Even if the use of a discovered encrypted DNS server is beyond the discovery process and falls under encrypted server selection, the following discusses typical conditions under which discovered encrypted DNS server can be used.

- o If the DNS server's IP address discovered by using DHCP/RA is preconfigured in the OS or Browser as a verified resolver (e.g., part of an auto-upgrade program such as [Auto-upgrade]), the DNS client auto-upgrades to use the preconfigured encrypted DNS server tied to the discovered DNS server IP address. In such a case the DNS client will perform additional checks out of band, such as confirming that the Do53 IP address and the encrypted DNS server are owned and operated by the same organisation.
- o Similarly, if the ADN conveyed in DHCP/RA (Sections 4, 5, and 6) is preconfigured in the OS or browser as a verified resolver, the

DNS client auto-upgrades to establish an encrypted a DoH/DoT/DoQ session with the ADN.

In such case, the DNS client matches the domain name in the Encrypted DNS DHCP/RA option with the 'DNS-ID' identifier type within subjectAltName entry in the server certificate conveyed in the TLS handshake.

#### Authors' Addresses

Mohamed Boucadair  
Orange  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com

Tirumaleswar Reddy  
McAfee, Inc.  
Embassy Golf Link Business Park  
Bangalore, Karnataka 560071  
India

Email: TirumaleswarReddy\_Konda@McAfee.com

Dan Wing  
Citrix Systems, Inc.  
USA

Email: dwing-ietf@fuggles.com

Neil Cook  
Open-Xchange  
UK

Email: neil.cook@noware.co.uk

Tommy Jensen  
Microsoft  
USA

Email: tojens@microsoft.com

add  
Internet-Draft  
Intended status: Informational  
Expires: January 29, 2021

D. Migault  
Ericsson  
July 28, 2020

DNS Resolving service Discovery Protocol (DRDP)  
draft-mglt-add-rdp-03

Abstract

This document describes the DNS Resolver Discovery Protocol (DRDP) that enables a DNS client to discover various available local and global resolving service. The discovery is primarily initiated by a DNS client, but a resolving service may also inform the DNS client other resolving services are available and eventually preferred.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 29, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Requirements Notation . . . . .	2
2. Introduction . . . . .	2
3. Terminology . . . . .	3
4. Overview . . . . .	4
5. Pointer to a list of Resolving Domains . . . . .	5
6. Discovery of Resolving Services . . . . .	6
7. TTL . . . . .	7
8. SvcParamKey . . . . .	7
9. Resolver advertising other service sub type . . . . .	8
10. Migration to service sub types . . . . .	8
11. Security Considerations . . . . .	8
11.1. Use of protected channel is RECOMMENDED . . . . .	8
11.2. DNSSEC is RECOMMENDED . . . . .	9
11.3. TLSA is RECOMMENDED . . . . .	10
12. Privacy Considerations . . . . .	10
13. IANA Considerations . . . . .	11
14. Acknowledgments . . . . .	11
15. Appendices . . . . .	12
15.1. DRDP Requirements . . . . .	12
15.2. Discovery of specific service instance . . . . .	13
16. References . . . . .	14
16.1. Normative References . . . . .	14
16.2. Informative References . . . . .	14
Author's Address . . . . .	15

## 1. Requirements Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. Introduction

A DNS client can proceed to DNS resolution using various resolving services. These services can be local or global and can use a wide range of DNS transport protocols such as, for example, standard DNS [RFC1035], DNS over TLS [RFC7858] or DNS over HTTPS [RFC8484]. The local scope of these services may take various forms. For example, it could be associated to a network perspective ( restricted to the network the DNS client is connected to ) or to an application perspective ( restricted to some domain names ).

The purpose of the DNS Resolving service Discovery Protocol (DRDP) is to discover resolving services available to the DNS client. These

available resolving services to a given DNS client may highly depend on its location or browsing activity. The number of resolving services available to the DNS client is expected to remain quite consequent and evolve over time. Similarly, characteristics associated to these resolving services may also evolve over time. As a result, the DNS client is unlikely willing to synchronize such a huge data base of resolving services. DRDP proposes an alternative that consists in adaptively discovering the available resolving services based on the DNS client context.

DRDP adopts a hierarchical approach where the DNS client (or DRDP client) discovers the resolving services from resolving domains (RD) or a pointer to a list of resolving domains (Pointer).

The document does not describe how the DNS client is provisioned with RD or RD\_list. The DNS client may obtain the contextual resolving domains via various way, including a configuration, via DHCP Options [I-D.btw-add-home] or derived from specific procedures [I-D.mglt-add-drdp-isp]. The DNS client is expected to discover resolving services from all RD or RD\_list before proceeding to a selection process. The selection process of the resolving service is out of scope of this document.

### 3. Terminology

**DNS client** the client that sends DNS queries for resolution. In this document the DNS client designates also the end entity that is collecting information about the available Resolving Services and then proceed to the selection of a subset them. The selection is processed according to the DNS client's policy.

**Resolving Service** designates a service that receives DNS queries from a DNS client and resolves them. A Resolving Service is implemented by one or multiple resolvers.

**Resolver:** A resolver designates the software or hardware handling the DNS exchange. See [RFC7719] for more details.

**DNS transport** designates the necessary parameters a DNS client needs to establish a session with a Resolving Service.

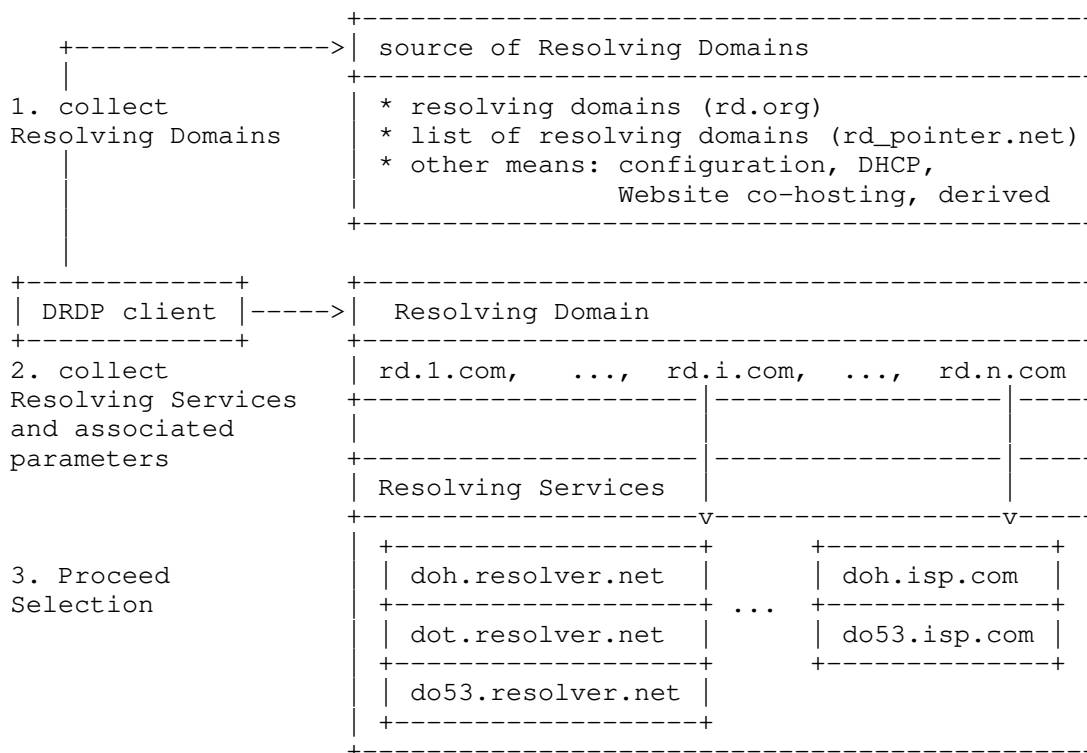
**Resolving Domain** a DNS domain that hosts one or multiple resolving services.



#### 4. Overview

DRDP is a DNS based protocol that addresses the requirements listed in Section 15.1. The figure below represents provides a high level description of DRDP.

1. The DRDP client considers the source of resolving domains (RD). This document defines two ways to collect RDs: RD are directly provisioned - in our case rd.org) , or RD are retrieved from a Pointer - in our case rd\_pointer.org. In the later case RD are collected from a DNS request of type PTR. In the case of Pointer being rd\_pointer.org, the QNAME of the PTR request would be b.\_dns.rd\_pointer.org.
2. The DRDP client collects the resolving services and associated parameters under the umbrella of each RD. The resolving services offered by the RD are collected via a DNS request of type SVCB and the associated parameters are provided through SvcParameters. In the case of the RD being rd.org, the QNAME of the SVCB request would be \_dns.rd.org.



## 5. Pointer to a list of Resolving Domains

A Pointer is a FQDN that points to a list of FQDN that designates RD. If Pointer is represented by `rd_pointer.net`, the associated RDs are retrieved by the DNS query of type PTR for `b._dns.rd_pointer.org`.

The zone file below is inspired from DNS-SD where `b` indicates a browsing domain, `_dns` indicates the DNS resolving service, `rd_pointer.org` the Pointer and `rd.1.com`, ... `rd.i.com` the associated RDs. Note that they do not necessarily need to share a TLD. The order of the resolving domains is irrelevant, and the zone administrator SHOULD regularly reorder them. The RRsets MUST be signed with DNSSEC.

```
b._dns.rd_pointer.net  PTR rd.1.com
[...]
b._dns.rd_pointer.net  PTR rd.n.com
```

Using the DNS provides the advantage to retrieve the resolving domain without requiring other libraries than DNS as well as benefit from the DNS caching infrastructure including the use of the TTL.

An EDNS buffer size of 1232 bytes will avoid fragmentation on nearly all current networks. This is based on an MTU of 1280, which is required by the IPv6 specification, minus 48 bytes for the IPv6 and UDP headers. This document RECOMMENDS that the number of RDs associated to a Pointer do not generate fragmentation of the DNS UDP packet. It is believed to address most common needs or expectation from a vast majority of stub DNS client.

When the number of RD exceeds this limit, the DNS client may carry this over TCP which is likely to be supported by DNS client willing to upgrade to DoH or DoT resolving services. However, the transfer of large number of RDs is considered as an application specificity that would benefit from the compression of the transferred data provided by ftp or http. In such case, these application may define there own specific mechanism to provision the RDs.

As of July 27 2020, 1232 bytes correspond to the 94 first most popular FQDN listed by [moz.com]. The current size of such lists [curl][dnsprivacy.org] have less than 50 resolving domains. Other lists such as [public-dns.info] have as much as 11.000 entries, but such lists seems to contain open resolvers which is out side of the scope of a selection process.

Web browser (Google Chrome) also have lists over 10.000 entries, but in case a significant number of entries seems to be IP addresses that have a very limited network scope ( e.g. limited to the ISP ). The length of the list in scope to the DNS client is in fact significant

smaller in term of IP addresses and even smaller if resolving domain are able to represent multiple IP addresses. Overall, the size of such lists are currently due to the absence of discovery protocols.

## 6. Discovery of Resolving Services

The discovery of resolving services is performed by the RDP client with all the available RDs. Given a RD `rd.org`, a DRDP client sends a DNS request of type SVCB for `_dns.rd.org`.

The example below presents the use of an AliasForm followed by a ServiceForm which allows an indirection. The Alias form is not mandatory and instead only ServiceForm associated to `_dns.rd.org` could have been used instead.

The `SvcFieldPriority` indicates the preference of the RD. It typically enables an operator to indicate that an encrypted DNS is preferred.

The `SvcParamKey alpn` MUST be present when TLS is used as its presence and value indicates the DNS transport. The absence of the `alpn SvcParamKey` indicates Do53, `alpn` set to `dot` indicates DoT is served while `h*` indicates DoH is served. Note that the port value (53, 853, 443) is not used to determine the DNS transport as non standard port MAY be used. The example below uses an non standard port 5353 for illustrative purpose.

Other `SvcParam` are detailed in Section 8 and are optional. A `SvcParam` not understood by the DNS client MUST be ignored.

The RRsets MUST be protected with DNSSEC and when `alpn` is provided a TLSA RRset SHOULD be present. These RRsets have been omitted for clarity.

```
## Discovery of all service instances
_dns.rd.org. 7200 IN SVCB 0 svc.example.com.
svc.example.com. 7200 IN SVCB 12 ( svc0.example.net.
                                port="5353" user-display="Legacy Resolver"
)
svc.example.com. 7200 IN SVCB 1 ( svc1.example.net. alpn="dot"
                                port="5353" esniconfig="..."
                                user-display="Preferred Example's Choice" )
svc.example.com. 7200 IN SVCB 3 ( svc2.example.net. alpn="h2"
                                port="5353" esniconfig="..." user-display=
)
svc.example.com. 7200 IN SVCB 2 ( svc3.example.net. alpn="h3"
                                port="5353" esniconfig="..." user-display=
)
```

Note that Section 15.2 provides another variant to perform RDP. Such variant is left for further discussion and address the need to be able to narrow down the discovery to a subset of resolving services such as DoH-only or DoT-only services.

Some notes:

1. `_domain` uses SVCB but does not have TLS. While SVCB has been created essentially for TLS based service, this does not appear to be mandatory.
2. Should we have some constraints regarding the `SvcDomainName` and `QNAME` ?
3. do we need the service subsets

#### 7. TTL

The DNS client SHOULD perform DRDP at regular intervals as indicated by its policy.

The selection process MAY remove resolving services with short TTL lower than a day as it indicates some source of instability. Given a subset of selected resolving services, the DNS client may perform DRDP 1 hour before an SVB RRset expires.

#### 8. SvcParamKey

This section defines a set of SvcParamKey that MAY be use to carry the necessary informations for the selection process.

`alpn` :

`esniconfig` :

`port` :

`user-display` indicates a strings in UTF-8 that is expected to be representative to a potential end user. Though there is no restriction in the scope of that string. The string is likely to represent the service within the resolving domain.

`uri_template` designates the URI template for DoH. This key MUST NOT be present on non DoH services and MUST be ignored by the DNS client on non DoH resolving Services.

`auth_domain` indicates the list of authoritative domain name the resolving service has strong relation with. It is expected that a

resolving service may prefer to perform DNS resolution over these domains to that specific resolving service as to preserve its privacy. This information MUST be verified and validated.

`scope_domain` indicates the limitation of resolved domains. When present DNS request sent to the resolution service MUST belong to that domain.

`filtering` indicates the presence of a filtering service

`ip_subnet` indicates a subnetwork restriction. This is mostly useful for resolving services that are not globally.

`dnssec` indicates whether dnssec is enabled or not.

#### 9. Resolver advertising other service sub type

A resolving service receiving a DNS request over a service sub type MAY be willing to advertise the DNS client that other sub service type are available. This is especially useful, when, for example, a resolver wants that the DNS resolver switches to other service sub types that are more secure.

In order to do so the resolver MAY provide in the additional data field the `_dns SRVCB` of `ServiceForm`.

#### 10. Migration to service sub types

The principle of the discovery mechanism is that the resolver indicates the available service sub types and let the DNS client chose which sub type it prefers. On the other hand, the resolver MAY also indicate a preference using the priority and weight fields. Redirection MAY especially be needed when a DNS client is using the Do53 and the resolver would like to upgrade the DNS client session to a more secure session. This MAY require a specific ERROR code that will request the DNS client to perform service discovery.

It is expected that DRDP MUST always be available via Do53. However, this does not mean that a resolver is expected to implement the Do53 sub type service for a resolving service.

#### 11. Security Considerations

##### 11.1. Use of protected channel is RECOMMENDED

When available, it is recommended to chose a protected version of the `rdns` service. More specifically, the use of end-to-end protection ensures that the DNS client is connected to the expected platform and

that its traffic cannot be intercepted on path. Typically, the selection of resolver on the Internet (and not on your ISP network) and the use of a non protected channel enables an attacker to monitor your DNS traffic. The similar observation remains true if you are connected to the resolver of your ISP. It is commonly believed that trusting your ISP (that is your first hop) makes encryption unnecessary. Trusting your ISP is mandatory in any case, but the associated level of trust with an protected channel is restricted to the operation of the DNS platform. With non protected channel the trust is extended to any segment between the DNS client and the resolver, which is consequently larger. The use of a protected channel is recommended as it will prevent anyone on path to monitor your traffic.

#### 11.2. DNSSEC is RECOMMENDED

The exchanges SHOULD be protected with DNSSEC to ensure integrity of the information between the authoritative servers and the DNS client. Without DNSSEC protection, DNS messages may be tampered typically when they are transmitted over an unprotected channel either between the DNS client and the resolver or between the resolver and the authoritative servers. The messages may be tampered by an online attacker intercepting the messages or by the intermediary devices. It is important to realize that protection provided by TLS is limited to the channel between the DNS client and the resolver. There are a number of cases where the trust in the resolver is not sufficient which justify the generalization of the use of DNSSEC. The following examples are illustrative and are intended to be exhaustive.

First, the discovery exchanges may happen over an unprotected channel, in which case, the messages exchanged may be tampered by anyone on-path between the DNS client and the resolver as well as between the resolver and the authoritative servers - including the resolver. When TLS is used between the DNS client and the resolver, this does not necessarily mean the DNS client trusts the resolver. Typically, the TLS session may be established with a self-signed certificate in which case the session is basically protected by a proof-of-ownership. In other cases, the session may be established based on Certificate Authorities (CA) that have been configured into the TLS client, but that are not necessarily trusted by the DNS client. In such cases, the connected resolver may be used to discover resolvers from another domain. In this case, the resolver is probably interacting with authoritative servers using untrusted and unprotected channels. Integrity protection relies on DNSSEC.

### 11.3. TLSA is RECOMMENDED

When TLS is used to protect the DNS exchanges, certificates or fingerprint SHOULD be provided to implement trust into the communication between the DNS client and the resolver. The TLS session and the association of the private key to a specific identity can be based on two different trust model. The Web PKI that will rely on CA provisioned in the TLS library or the TA provided to the DNS client. A DNS client SHOULD be able to validate the trust of a TLS session based on the DNSSEC trust model using DANE.

When the DNS client is protecting its session to the resolver via TLS, the DNS client may initiate an TLS session that is not validated by a CA or a TLSA RRsets. The DNS client MUST proceed to the discovery process and validate the certificate match the TLSA RRset. In case of mismatch the DNS client MUST abort the session.

### 12. Privacy Considerations

When the discovery protocol is performed using a resolver that belongs to one domain for another domain, or over an unprotected channel, the DNS client must be conscious that its is revealing to the resolver its intention to use another resolver. More specifically, suppose an resolver is complying some legal requirements that DNS traffic must be unencrypted. Using this resolver to perform a resolver discovery reveals the intention of potentially using alternative resolvers. Alternatively, narrowing down the discovery over a specific sub type of resolver (DoT, or DoH) may reveal to that resolver the type of communication. As result, when performing a discovery over a domain that differs to the domain the resolver belongs to, it is RECOMMENDED to request the SRV RRsets associated to all different sub type of proposed services.

The absence of traffic that results from switching completely to a newly discovered resolver right after the discovery process provides an indication to the resolver the DNS client is switching to. It is hard to make that switch unnoticed to the initial resolver and the DNS resolver MUST assume this will be noticed. The information of switching may be limited by sharing the traffic between different resolvers, however, the traffic pattern associated to each resolver may also reveal the switch. In addition, when the initial resolver is provided by the ISP, the ISP is also able to monitor the IP traffic and infer the switch. As a result, the DNS client SHOULD assume the switch will be detected.

With DoT or DoH, the selection of port 443 will make the traffic indistinguishable from HTTPS traffic. This means that an observer will not be able to tell whether the traffic carries web traffic or

DNS traffic. Note that it presents an interest if the server offers both a web service as well as a resolution service. Note that many resolvers have a dedicated IP address for the resolution service, in which case, the information will be inferred from the IP address. Note also that traffic analysis may infer this as well. Typically suppose an IP address hosts one or multiple web sites that are not popular as well as a resolving service. If this IP address is associated frequent short size exchanges, it is likely that these exchanges will be DNS exchanges rather than Web traffic. The size of the packet may also be used as well as many other patterns. As a result, the use port 443 to hide the DNS traffic over web traffic should be considered as providing limited privacy.

### 13. IANA Considerations

This document requests the IANA the creation of the following underscored node names in the Underscored and Globally Scoped DNS Node Names registry <https://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-14>

RR Type	_NODE NAME	Reference
SRVCB	_dns	RFC-TBD

SvcParamKey	NAME	Meaning	Reference
7	user-display	User friendly string (UTF8) to represent the resolver	RFC-TBD
	uri_template	URI template	
	auth_domain	Domains the resolving service is authoritative	
	filetring	Filetring services provided	
	ip_subnet	ip ranges accepted.	
	dnssec	DNSSEC validation enabled	

### 14. Acknowledgments

We would like thank Mirja Kuehlewind as well as the GSMA IG for their comments. We also thank Ted Hardie and Paul Hoffman for their feedbacks regarding the dns schemes for DoT and DoH. We thank Ben Schwartz for the comments on the list size. We thank Harald Alvestrand for its recommendation on having a model that enable multiple third party providers to provide their own list of resolving domains. We thank Stephan Bortzmeyer, Ralf Weber, Chris Box for its clarifications.



## 15. Appendices

### 15.1. DRDP Requirements

This section lists the DRDP requirements.

REQ 1: DRDP MUST enable a DNS client to discover the available resolving services with their associated characteristics in order to proceed to a selection process. The selection process takes resolving services identities and associated parameters and proceed to the selection.

Any sort of resolving service selection is outside the scope of DRDP.

REQ 2: While the discovery process is triggered by the DNS client, a third party MUST be able to provide necessary input information so a resolving service discovery process can be triggered within a specific context.

Provisioning protocols to provide this information is not as per say in scope of DRDP. DRDP defines the format of the format for such input as well as a set of such inputs.

REQ 3: Any information used in DRDP MUST be authenticated by its owner. In particular, the characteristics associated to the resolving service MUST be certified by the resolving service operator / owner and MUST be associated a validity period. In addition, a third party providing a set of inputs MUST authenticate that set.

REQ 4: Information associated to the resolving services is intended to enable the selection process that is assumed to match the end user or application policy. The selection process is out of scope of DRDP. Information may carry some characteristics of a resolving service or hints that will help the selection. In particular an operator of multiple resolving service MUST be able to associate a preference to the proposed resolving services. To ease automation of the selection as well as to make multiple applications benefit from DRDP the information MUST be carried over a standardized format.

REQ 5: DRDP MUST be designed to be used indifferently by a DNS client using any DNS transport protocol (Do53, DoT, DoH, ...). However, DRDP SHOULD be able to restrict the information retrieved to a certain type of resolving service, i.e. Do53, DoT, DoH.

REQ 6: DRDP deployment MUST NOT be disruptive for the legacy DNS client or infrastructure and legacy client SHOULD be able to incrementally include DRDP.

## 15.2. Discovery of specific service instance

To reduce the size of the messages, the DNS client MAY also prefer to query information of resolving services using a specific transport (DNS, DoT, DoH) that are designated as sub sets. A DNS client MAY list the different subsets of that resolving domain with a PTR query. This document defines the following subsets `_53._dns` for DNS, `_853._dns` for DoT and `_443._dns` for DoH. Other subsets MAY be defined in the future. A DNS client that does not understand a subset SHOULD ignore it and maybe proceed to the discovery as defined in Section 6.

All subsets MUST share the same resolving domain and be listed with a PTR RRs. The DNS client MAY NOT performed a DNS query of type PTR, for example, if it has a previous knowledge of the existence of the subset or if indicated by its policy. In this it MAY directly proceed to the SRVCB resolution.

The same restrictions as defined in section Section 6 apply.

Note that while the `SvcFieldPriority` indicates the priority within a subservice, this field MUST have a coherence across subservices. The priority provided SHOULD be coherent with the case of a `_dns` SRVCB query of section Section 6.

The figure below illustrates an example of zone file. RRSIG and TLSA have been omitted for the purpose of clarity.

```
### Definition of the resolving service subsets
_dns.example.com PTR _53._dns.example.com
_dns.example.com PTR _853._dns.example.com
_dns.example.com PTR _443._dns.example.com

### services instances per service subset
_53._dns.example.com. 7200 IN SVCB 0 svc0.example.com.
svc0.example.com.    7200 IN SVCB 12 ( svc0.example.net.
                                port="5353" user-display="Legacy Resolver"
                                )
_853._dns.example.com. 7200 IN SVCB 0 svc1.example.com.
svc1.example.com.    7200 IN SVCB 1 ( svc1.example.net. alpn="dot"
                                port="5353" esniconfig="..."
                                user-display="Preferred Example's Choice" )
_443._dns.example.com. 7200 IN SVCB 0 svc4.example.net.
svc4.example.com.    7200 IN SVCB 3 ( svc2.example.net. alpn="h2"
                                port="5353" esniconfig="..." user-display=
                                )
svc4.example.com.    7200 IN SVCB 2 ( svc3.example.net. alpn="h3"
                                port="5353" esniconfig="..."
                                user-display="Testing QUIC")
```

## 16. References

### 16.1. Normative References

- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.

### 16.2. Informative References

- [curl] "Publicly available servers", n.d., <<https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers>>.
- [dnsprivacy.org] "DNS Privacy Test Servers", n.d., <<https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+Test+Servers#DNSPrivacyTestServers-Publicresolvers>>.
- [I-D.btw-add-home] Boucadair, M., Reddy, K. T., Wing, D., and N. Cook, "Encrypted DNS Discovery and Deployment Considerations for Home Networks", draft-btw-add-home-07 (work in progress), July 2020.
- [moz.com] "The Moz Top 500 Websites", n.d., <<https://moz.com/top500>>.

[public-dns.info]  
"Public DNS Server List", n.d.,  
<<https://public-dns.info/>>.

[RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <<https://www.rfc-editor.org/info/rfc7719>>.

Author's Address

Daniel Migault  
Ericsson  
8275 Trans Canada Route  
Saint Laurent, QC 4S 0B6  
Canada

EMail: [daniel.migault@ericsson.com](mailto:daniel.migault@ericsson.com)

ADD WG  
Internet-Draft  
Intended status: Standards Track  
Expires: April 10, 2022

T. Reddy  
Akamai  
D. Wing  
Citrix  
M. Richardson  
Sandelman Software Works  
M. Boucadair  
Orange  
October 7, 2021

DNS Server Selection: DNS Server Information with Assertion Token  
draft-reddy-add-server-policy-selection-09

## Abstract

The document defines a mechanism that is meant to communicate DNS resolver information to DNS clients for use as a criteria for server selection decisions. Such an information that is cryptographically signed to attest its authenticity is used for the selection of DNS resolvers. Typically, evaluating the resolver information and the signatory, DNS clients with minimal or no human intervention can select the DNS servers for resolving domain names.

This assertion is useful for encrypted DNS (e.g., DNS-over-TLS, DNS-over-HTTPS, or DNS-over-QUIC) servers that are either public resolvers or discovered in a local network.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 10, 2022.

## Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	3
2. Terminology . . . . .	3
3. Resolver Assertion Token (REAT): Overview . . . . .	4
4. REAT Header . . . . .	6
4.1. 'typ' (Type) Header Parameter . . . . .	6
4.2. 'alg' (Algorithm) Header Parameter . . . . .	6
4.3. 'x5u' (X.509 URL) Header Parameter . . . . .	6
4.4. An Example of REAT Header . . . . .	7
5. REAT Payload . . . . .	7
5.1. JWT Defined Claims . . . . .	7
5.1.1. 'iat' - Issued At Claim . . . . .	7
5.1.2. 'exp' - Expiration Time Claim . . . . .	7
5.2. REAT Specific Claims . . . . .	8
5.2.1. DNS Server Identity Claims . . . . .	8
5.2.2. 'resinfo' (Resolver Information) Claim . . . . .	8
5.2.3. An Example . . . . .	9
6. REAT Signature . . . . .	9
7. Extending REAT . . . . .	10
8. Deterministic JSON Serialization . . . . .	10
8.1. Example REAT Deterministic JSON Form . . . . .	11
9. Using RESINFO Responses . . . . .	11
10. Security Considerations . . . . .	12
11. IANA Considerations . . . . .	12
11.1. Media Type Registration . . . . .	12
11.1.1. Media Type Registry Contents Additions Requested . .	12
11.2. JSON Web Token Claims Registration . . . . .	13
11.2.1. Registry Contents Additions Requested . . . . .	13
11.3. DNS Resolver Information Registration . . . . .	14
12. Acknowledgments . . . . .	14
13. References . . . . .	14
13.1. Normative References . . . . .	14

13.2. Informative References . . . . .	16
Appendix A. Example of ES256-based REAT JWS Serialization and Signature . . . . .	18
A.1. X.509 Private Key in PKCS#8 Format for ES256 Example** . . . . .	20
A.2. X.509 Public Key for ES256 Example** . . . . .	20
Appendix B. Complete JWS JSON Serialization Representation with multiple Signatures . . . . .	20
B.1. X.509 Private Key in PKCS#8 format for E384 Example** . . . . .	21
B.2. X.509 Public Key for ES384 Example** . . . . .	21
Authors' Addresses . . . . .	21

## 1. Introduction

[RFC7626] discusses DNS privacy considerations in both "on the wire" (Section 2.4 of [RFC7626]) and "in the server" (Section 2.5 of [RFC7626]) contexts. Examples of protocols that provide encrypted channels between DNS clients and servers are DNS-over-HTTPS (DoH) [RFC8484], DNS-over-TLS (DoT) [RFC7858], and DNS-over-QUIC (DoQ) [I-D.ietf-dprive-dnsquic].

DNS clients can discover and authenticate encrypted DNS servers provided by a local network, for example using the techniques proposed in [I-D.ietf-add-dnr] and [I-D.ietf-add-ddr]. If the mechanism used to discover the encrypted DNS server is insecure, the DNS client needs evidence about the encrypted server to assess its trustworthiness and a way to appraise such evidence. The mechanism specified in this document can be used by the DNS client to cryptographically identify if it is connecting to an encrypted DNS server hosted by a specific organization (e.g., ISP or Enterprise). This strengthens the protection as clients can detect and reject connections to encrypted DNS servers hosted by attackers.

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

This document makes use of the terms defined in [RFC8499] and [I-D.ietf-dnsop-terminology-ter].

'Encrypted DNS' refers to a DNS protocol that provides an encrypted channel between a DNS client and server (e.g., DoT, DoH, or DoQ).

The terms 'Evidence', 'Verifier', 'Background Check', 'Relying Party', 'Appraisal Policy', and 'Attestation Results' are defined in [I-D.ietf-rats-architecture].

### 3. Resolver Assertion Token (REAT): Overview

The mechanism used in this specification resembles the Background-Check Model discussed in Sections 5.2 and 5.3 of Remote attestation procedure (RATS) Architecture [I-D.ietf-rats-architecture]. RATS enables a relying party to establish a level of confidence in the trustworthiness of a remote peer through the creation of Evidence to assess the peer's trustworthiness, and an Appraisal Policy for such Evidence.

In this document, the Relying Party is the DNS client and the Attester is the encrypted DNS server. The Encrypted DNS servers MAY use "Domain Validation" (DV) certificates for certificate-based server authentication in TLS connections.

The DNS server's resolver information needs to be validated and signed. This signature is called an Attestation Result [I-D.ietf-rats-architecture]. This validation can be performed by the DNS operator itself (signed by the DNS operator's certificate) acting as a verifier or performed by an external Verifier (signed by that external Verifier). The signing certificate can be an Extended Validation (EV) certificate issued by a public CA in specific scenarios listed below. An EV certificate is issued by the public CA after a thorough Background Check to verify the requesting organization's legal identity. If the signing certificate is a EV certificate, it leaves the client with a better audit trail of the organization hosting the DNS server in comparison with the DV certificate.

The use of EV certificate is needed in the following scenarios:

- o It helps the client to avoid sending DNS queries to an Encrypted DNS server hosted by an attacker discovered insecurely (e.g., using DHCP/RA or DNS). For example, an attacker can get a domain name, domain-validated public certificate from a CA and host a Encrypted DNS server. Furthermore, an attacker can use a public IP address, get an 'IP address'-validated public certificate from a CA and host a Encrypted DNS server.
- o It can be used by the client to identify the Encrypted DNS server is hosted by a legal organization.

The use of EV certificate is not required in the following scenarios:



- o If the Encrypted DNS server can only be discovered securely (e.g., using IKEv2 [I-D.btw-add-ipsecme-ike]), the signing certificate need not be an EV certificate.
- o Secure Zero Touch Provisioning [RFC8572] defines a bootstrapping strategy for enabling a networking device to securely obtain the required configuration information with no user input. If the encrypted DNS server is insecurely discovered and not preconfigured in the networking device, the DNS client on the networking device can validate the Resolver Assertion Token signature using the owner certificate as per Section 3.2 of [RFC8572].

JSON Web Token (JWT) [RFC7519] and JSON Web Signature (JWS) [RFC7515] and related specifications define a standard token format that can be used as a way of encapsulating claimed or asserted information with an associated digital signature using X.509 based certificates. JWT provides a set of claims in JSON format that can accommodate asserted resolver information of the Encrypted DNS server. Additionally, JWS provides a path for updating methods and cryptographic algorithms used for the associated digital signatures.

JWS defines the use of JSON data structures in a specified canonical format for signing data corresponding to JOSE header, JWS Payload, and JWS Signature. The next sections define the header and claims that MUST be minimally used with JWT and JWS for resolver assertion token.

The REsolver Assertion Token (REAT) specifically uses this token format and defines claims that convey the resolver information of Encrypted DNS server.

The client can retrieve the REAT object using the RESINFO RRtype defined in [I-D.reddy-add-resolver-info] and QNAME of the domain name that is used to authenticate the DNS server (referred to as ADN in [RFC8310]). If the special use domain name "resolver.arpa" defined in [I-D.ietf-add-ddr] is used to discover the Encrypted DNS server, the client can retrieve the REAT object using the RESINFO RRtype and QNAME of the special use domain name.

The signature of REAT object MUST be validated by the DNS client. If signature is invalid, the REAT object is rejected. If signature is valid and signer is trusted, the DNS client can use that encrypted DNS server.

#### 4. REAT Header

The JWS token header is a JOSE header (Section 4 of [RFC7515]) that defines the type and encryption algorithm used in the token.

The REAT header MUST include, at a minimum, the header parameters defined in Sections 4.1, 4.2, and 4.3.

##### 4.1. 'typ' (Type) Header Parameter

The 'typ' (Type) Header Parameter is defined in Section 4.1.9 of [RFC7515] to declare the media type of the complete JWS.

For REAT Token the 'typ' header MUST be the string 'rat'. This represents that the encoded token is a JWT of type rat.

##### 4.2. 'alg' (Algorithm) Header Parameter

The 'alg' (Algorithm) Header Parameter is defined in Section 4.1.1 of [RFC7515]. It specifies the JWS signature cryptographic algorithm. It also refers to a list of defined 'alg' values as part of a registry established by JSON Web Algorithms (JWA) [RFC7518] Section 3.1.

For the creation and verification of REAT tokens and their digital signatures, implementations MUST support ES256 as defined in Section 3.4 of [RFC7518]. Implementations MAY support other algorithms registered in the JSON Web Signature and Encryption Algorithms registry created by [RFC7518]. The content of that registry may be updated in the future depending on cryptographic strength requirements guided by current security best practice. The mandatory-to-support algorithm for REAT tokens may likewise be updated in the future.

Implementations of REAT digital signatures using ES256 as defined above SHOULD use deterministic ECDSA when supported for the reasons stated in [RFC6979].

##### 4.3. 'x5u' (X.509 URL) Header Parameter

As defined in Section 4.1.5 of [RFC7515], the 'x5u' header parameter defines a URI [RFC3986] referring to the resource for the X.509 public key certificate or certificate chain [RFC5280] corresponding to the key used to digitally sign the JWS. Generally, as defined in Section 4.1.5 of [RFC7515] this corresponds to an HTTPS or DNSSEC resource using integrity protection.

#### 4.4. An Example of REAT Header

An example of the REAT header is shown in Figure 1. It includes the specified REAT type, ES256 algorithm, and an URI referencing the network location of the certificate needed to validate the REAT signature.

```
{
  "typ":"rat",
  "alg":"ES256",
  "x5u":"https://cert.example.com/rat.cer"
}
```

Figure 1: A REAT Header Example

### 5. REAT Payload

The token claims consist of the resolver information of the DNS server that needs to be verified at the DNS client. These claims follow the definition of a JWT claim (Section 4 of [RFC7519]) and are encoded as defined by the JWS Payload (Section 3 of [RFC7515]).

REAT defines the use of a standard JWT-defined claim as well as custom claims corresponding to the DoT or DoH servers.

Claim names MUST use the US-ASCII character set. Claim values MAY contain characters that are outside the ASCII range, however they MUST follow the default JSON serialization defined in Section 7 of [RFC7519].

#### 5.1. JWT Defined Claims

##### 5.1.1. 'iat' - Issued At Claim

The JSON claim MUST include the 'iat' (Section 4.1.6 of [RFC7519]) defined claim "Issued At". The 'iat' should be set to the date and time of issuance of the JWT. The time value should be of the format (NumericDate) defined in Section 2 of [RFC7519].

##### 5.1.2. 'exp' - Expiration Time Claim

The JSON claim MUST include the 'exp' (Section 4.1.4 of [RFC7519]) defined "claim Expiration Time". The 'exp' should be set to specify the expiration time on or after which the JWT is not accepted for processing. The REAT object should expire after a reasonable duration. A short expiration time for the REAT object periodically reaffirms the resolver information of the DNS server to the DNS client and ensures the DNS client does not use outdated resolver

information. If the DNS client knows the REAT object has expired, it should make another request to get the new REAT object from the DNS server.

## 5.2. REAT Specific Claims

### 5.2.1. DNS Server Identity Claims

The DNS server identity is represented by a claim that is required for REAT: the 'server' claim. The 'server' MUST contain claim values that are identity claim JSON objects where the child claim name represents an identity type and the claim value is the identity string, both defined in subsequent subsections.

These identities can be represented as either authentication domain name (ADN) (defined in [RFC8310]) or Uniform Resource Indicators (URI).

The DNS client constructs a reference identifier for the DNS server based on the ADN or the domain portion in the URI of the DNS server identity. The domain name in the DNS-ID identifier type within subjectAltName entry in the DNS server certificate conveyed in the TLS handshake is matched with the reference identifier. If the match is not successful, the client MUST not accept the REAT for further processing.

#### 5.2.1.1. 'adn' - Authentication Domain Name Identity

If the DNS server identity is an ADN, the claim name representing the identity MUST be 'adn'. The claim value for the 'adn' claim is the ADN.

#### 5.2.1.2. 'uri' - URI Identity

If the DNS server identity is of the form URI Template, as defined in [RFC6570], the claim name representing the identity MUST be 'uri' and the claim value is the URI Template form of the DNS server identity.

As a reminder, if DoH is supported by the DNS server, the DNS client uses the URI Template (Section 3 of [RFC8484]).

### 5.2.2. 'resinfo' (Resolver Information) Claim

The 'resinfo' claim contains the resolver information of the DNS server defined in Section 5 of [I-D.reddy-add-resolver-info].

### 5.2.3. An Example

Figure 2 shows an example of resolver information.

```
{
  "server":{
    "adn":"example.com"
  },
  "iat":1443208345,
  "exp":1443640345,
  "resinfo": {
    "qnameminimization":false,
  }
}
```

Figure 2: An Example of Resolver Information

## 6. REAT Signature

The signature of the REAT is created as specified in Section 5.1 of [RFC7515] (Steps 1 through 6). REAT MUST use the JWS Protected Header.

For the JWS Payload and the JWS Protected Header, the lexicographic ordering and white space rules described in Section 4 and Section 5, and JSON serialization rules in Section 8 MUST be followed.

The REAT is cryptographically signed by the domain hosting the DNS server and optionally by a third party who performed privacy and security audit of the DNS server.

The resolver information is attested using "Extended Validation" (EV) certificate to avoid bad actors taking advantage of this mechanism to advertise encrypted DNS servers for illegitimate and fraudulent purposes meant to trick DNS clients into believing that they are using a legitimate encrypted DNS server hosted to provide privacy for DNS transactions.

Alternatively, a DNS client has to be configured to trust the leaf of the signer of the REAT object. That is, trust of the signer MUST NOT be determined by validating the signer via the OS or the browser trust chain because that would allow any arbitrary entity to operate a DNS server and assert any sort of resolver information.

Appendix A provides an example of how to follow the steps to create the JWS Signature.

JWS JSON serialization (Step 7 in Section 5.1 of [RFC7515]) is supported for REAT to enable multiple signatures to be applied to the REAT object. For example, the REAT object can be cryptographically signed by the domain hosting the DNS server and by a third party who performed privacy and security audit of the DNS server.

Appendix B includes an example of the full JWS JSON serialization representation with multiple signatures.

Section 5.1 of [RFC7515] (Step 8) describes the method to create the final JWS Compact Serialization form of the REAT Token.

## 7. Extending REAT

REAT includes the minimum set of claims needed to securely assert the resolver information of the DNS server. JWT supports a mechanism to add additional asserted or signed information by simply adding new claims. REAT can be extended beyond the defined base set of claims to represent other DNS server information requiring assertion or validation. Specifying new claims follows the baseline JWT procedures (Section 10.1 of [RFC7519]). Understanding new claims on the DNS client is optional. The creator of a REAT object cannot assume that the DNS client will understand the new claims.

## 8. Deterministic JSON Serialization

JSON objects can include spaces and line breaks, and key value pairs can occur in any order. It is therefore a non-deterministic string format. In order to make the digital signature verification work deterministically, the JSON representation of the JWS Protected Header object and JWS Payload object MUST be computed as follows.

The JSON object MUST follow the following rules. These rules are based on the thumbprint of a JSON Web Key (JWK) as defined in Section 3 of [RFC7638] (Step 1).

1. The JSON object MUST contain no whitespace or line breaks before or after any syntactic elements.
2. JSON objects MUST have the keys ordered lexicographically by the Unicode [UNICODE] code points of the member names.
3. JSON value literals MUST be lowercase.
4. JSON numbers are to be encoded as integers unless the field is defined to be encoded otherwise.

5. Encoding rules MUST be applied recursively to member values and array values.

#### 8.1. Example REAT Deterministic JSON Form

This section demonstrates the deterministic JSON serialization for the example REAT Payload shown in Section 5.2.3.

The initial JSON object is shown in Figure 3.

```
{
  "server":{
    "adn":"example.com"
  },
  "iat":1443208345,
  "exp":1443640345,
  "resinfo": {
    "qnameminimization":false,
  }
}
```

Figure 3: Initial JSON Object

The parent members of the JSON object are as follows, in lexicographic order: "exp", "iat", "resinfo", "server".

The final constructed deterministic JSON serialization representation, with whitespace and line breaks removed, (with line breaks used for display purposes only) is:

```
{"exp":1443640345,"iat":1443208345,
"resinfo":{"qnameminimization":false},
"server":{"adn":"example.com"}}
```

Figure 4: Deterministic JSON Form

#### 9. Using RESINFO Responses

This document defines the following entry for the IANA DNS Resolver Information Registry that is defined in [I-D.reddy-add-resolver-info].

- o The "attested-resinfo" name contains the full REAT object. The REAT header, REAT payload, and REAT signature components comprise a full REAT object. If the "attested-resinfo" name is conveyed to the client, the server need not convey the attributes "resinfourl", "identityurl", "extendeddnserver" and

"qnameminimization" attributes separately as that resolver information will be extracted by the client from the REAT payload.

## 10. Security Considerations

The use of REAT object based on the validation of the digital signature and the associated certificate requires consideration of the authentication and authority or reputation of the signer to attest the resolver information of the DNS server being asserted. Bad actors can host encrypted DNS servers to invade the privacy of the user. Bad actor can get a domain name, host encrypted DNS servers, and get the DNS server certificate signed by a CA. The resolver information will have to be attested using EV certificates or a REAT object signer trusted by the DNS client to prevent the attack.

The CA that issued the EV certificate does not attest the resolver information. The organization hosting the DNS server attests the resolver information using the EV certificate and the client uses the EV certificate to identify the organization (e.g., ISP or Enterprise) hosting the DNS server.

If the REAT object is asserted by a third party, it can do a "time of check" but the DNS server is susceptible of "time of use" attack. For example, changes to the DNS server can cause a disagreement between the auditor and the DNS server operation, hence the REAT object needs to be also asserted by the domain hosting the DNS server. In addition, the REAT object needs to have a short expiration time (e.g., 7 days) to ensure the DNS server's domain re-asserts the resolver information and limits the damage from change in behaviour and mis-issuance.

## 11. IANA Considerations

### 11.1. Media Type Registration

#### 11.1.1. Media Type Registry Contents Additions Requested

This section registers the 'application/rat' media type [RFC2046] in the 'Media Types' registry in the manner described in [RFC6838], which can be used to indicate that the content is a REAT defined JWT.

- o Type name: application
- o Subtype name: rat
- o Required parameters: n/a



- o Optional parameters: n/a
- o Encoding considerations: 8bit; application/rat values are encoded as a series of base64url-encoded values (some of which may be the empty string) separated by period ('.') characters..
- o Security considerations: See the Security Considerations Section of [RFC7515].
- o Interoperability considerations: n/a
- o Published specification: [THIS\_DOCUMENT]
- o Applications that use this media type: DNS
- o Fragment identifier considerations: n/a
- o Additional information:  
  
Magic number(s): n/a File extension(s): n/a Macintosh file type code(s): n/a
- o Person & email address to contact for further information: Tirumaleswar Reddy, kondtir@gmail.com
- o Intended usage: COMMON
- o Restrictions on usage: none
- o Author: Tirumaleswar Reddy, kondtir@gmail.com
- o Change Controller: IESG
- o Provisional registration? No

## 11.2. JSON Web Token Claims Registration

### 11.2.1. Registry Contents Additions Requested

IANA is requested to assign the following claims in the registry maintained in: <https://www.iana.org/assignments/jwt/jwt.xhtml>.

- o Claim Name: 'server'
- o Claim Description: DNS server identity
- o Change Controller: IESG

- o Specification Document(s): Section 5.2.1 of [THIS\_DOCUMENT]
- o Claim Name: 'resinfo'
- o Claim Description: Resolver information of DNS server.
- o Change Controller: IESG
- o Specification Document(s): Section 5.2.2 of [THIS\_DOCUMENT]

### 11.3. DNS Resolver Information Registration

IANA will add the name "attested-resinfo" to the DNS Resolver Information registry defined in Section 7.2 of [I-D.reddy-add-resolver-info].

### 12. Acknowledgments

This specification leverages some of the work that has been done in [RFC8225]. Thanks to Tommy Jensen, Ted Lemon, Paul Wouters, Neil Cook, Vittorio Bertola, Vinny Parla, Chris Box, Ben Schwartz and Shashank Jain for the discussion and comments.

### 13. References

#### 13.1. Normative References

[I-D.ietf-add-ddr]

Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", draft-ietf-add-ddr-03 (work in progress), October 2021.

[I-D.ietf-add-dnr]

Boucadair, M., Reddy, T., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", draft-ietf-add-dnr-02 (work in progress), May 2021.

[I-D.reddy-add-resolver-info]

Reddy, T. and M. Boucadair, "DNS Resolver Information", draft-reddy-add-resolver-info-03 (work in progress), April 2021.

[RFC2046]

Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/info/rfc2046>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", RFC 6570, DOI 10.17487/RFC6570, March 2012, <<https://www.rfc-editor.org/info/rfc6570>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <<https://www.rfc-editor.org/info/rfc6838>>.
- [RFC6979] Pornin, T., "Deterministic Usage of the Digital Signature Algorithm (DSA) and Elliptic Curve Digital Signature Algorithm (ECDSA)", RFC 6979, DOI 10.17487/RFC6979, August 2013, <<https://www.rfc-editor.org/info/rfc6979>>.
- [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015, <<https://www.rfc-editor.org/info/rfc7493>>.
- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/info/rfc7515>>.
- [RFC7518] Jones, M., "JSON Web Algorithms (JWA)", RFC 7518, DOI 10.17487/RFC7518, May 2015, <<https://www.rfc-editor.org/info/rfc7518>>.
- [RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, DOI 10.17487/RFC7519, May 2015, <<https://www.rfc-editor.org/info/rfc7519>>.

- [RFC7638] Jones, M. and N. Sakimura, "JSON Web Key (JWK) Thumbprint", RFC 7638, DOI 10.17487/RFC7638, September 2015, <<https://www.rfc-editor.org/info/rfc7638>>.
- [RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <<https://www.rfc-editor.org/info/rfc7858>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8499] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", BCP 219, RFC 8499, DOI 10.17487/RFC8499, January 2019, <<https://www.rfc-editor.org/info/rfc8499>>.

### 13.2. Informative References

- [I-D.btw-add-home] Boucadair, M., Reddy, T., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for Encrypted DNS Discovery", draft-btw-add-home-12 (work in progress), January 2021.
- [I-D.btw-add-ipsecme-ike] Boucadair, M., Reddy, T., Wing, D., and V. Smyslov, "Internet Key Exchange Protocol Version 2 (IKEv2) Configuration for Encrypted DNS", draft-btw-add-ipsecme-ike-03 (work in progress), May 2021.
- [I-D.ietf-dnsop-terminology-ter] Hoffman, P., "Terminology for DNS Transports and Location", draft-ietf-dnsop-terminology-ter-02 (work in progress), August 2020.
- [I-D.ietf-dprive-dnssoquic] Huitema, C., Dickinson, S., and A. Mankin, "Specification of DNS over Dedicated QUIC Connections", draft-ietf-dprive-dnssoquic-04 (work in progress), September 2021.

- [I-D.ietf-rats-architecture]  
Birkholz, H., Thaler, D., Richardson, M., Smith, N., and W. Pan, "Remote Attestation Procedures Architecture", draft-ietf-rats-architecture-12 (work in progress), April 2021.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", RFC 7159, DOI 10.17487/RFC7159, March 2014, <<https://www.rfc-editor.org/info/rfc7159>>.
- [RFC7626] Bortzmeyer, S., "DNS Privacy Considerations", RFC 7626, DOI 10.17487/RFC7626, August 2015, <<https://www.rfc-editor.org/info/rfc7626>>.
- [RFC7816] Bortzmeyer, S., "DNS Query Name Minimisation to Improve Privacy", RFC 7816, DOI 10.17487/RFC7816, March 2016, <<https://www.rfc-editor.org/info/rfc7816>>.
- [RFC8225] Wendt, C. and J. Peterson, "PASSporT: Personal Assertion Token", RFC 8225, DOI 10.17487/RFC8225, February 2018, <<https://www.rfc-editor.org/info/rfc8225>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8310] Dickinson, S., Gillmor, D., and T. Reddy, "Usage Profiles for DNS over TLS and DNS over DTLS", RFC 8310, DOI 10.17487/RFC8310, March 2018, <<https://www.rfc-editor.org/info/rfc8310>>.
- [RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <<https://www.rfc-editor.org/info/rfc8572>>.
- [RFC8914] Kumari, W., Hunt, E., Arends, R., Hardaker, W., and D. Lawrence, "Extended DNS Errors", RFC 8914, DOI 10.17487/RFC8914, October 2020, <<https://www.rfc-editor.org/info/rfc8914>>.
- [UNICODE] The Unicode Consortium, "The Unicode Standard", June 2016, <<http://www.unicode.org/versions/latest/>>.

## Appendix A. Example of ES256-based REAT JWS Serialization and Signature

For REAT, there will always be a JWS with the following members:

- o 'protected', with the value BASE64URL(UTF8(JWS Protected Header))
- o 'payload', with the value BASE64URL (JWS Payload)
- o 'signature', with the value BASE64URL(JWS Signature)

This example will follow the steps in JWS [RFC7515] Section 5.1, steps 1-6 and 8 and incorporates the additional serialization steps required for REAT.

Step 1 for JWS references the JWS Payload, an example REAT Payload is as follows:

```
{
  "server":{
    "adn":"example.com"
  },
  "iat":1443208345,
  "exp":1443640345,
  "resinfo": {
    "qnameminimization":false
  }
}
```

This would be serialized to the form (with line break used for display purposes only):

```
{"exp":1443640345,"iat":1443208345,"resinfo":{"qnameminimization":false},"server":{"adn":"example.com"}}
```

Step 2 Computes the BASE64URL(JWS Payload) producing this value (with line break used for display purposes only):

```
eyJleHAiOjE0NDM2NDZNDUsImhhbmFtZSI6MTQ0MzIwODM0NSwicmVzaW5mbyI6eyJxYXNmFtZWlwbmltaXphdGlvb20ifX0
```

For Step 3, an example REAT Protected Header comprising the JOSE Header is as follows:

```
{
  "alg": "ES256",
  "typ": "rat",
  "x5u": "https://cert.example.com/rat.cer"
}
```

This would be serialized to the form (with line break used for display purposes only):

```
{"alg": "ES256", "typ": "rat", "x5u": "https://cert.example.com/rat.cer"}
```

Step 4 Performs the BASE64URL(UTF8(JWS Protected Header)) operation and encoding produces this value (with line break used for display purposes only):

```
eyJhbGciOiJFUzI1NiIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5leGFtcGxlLmNvbS9yYXQuY2VyIn0
```

Step 5 and Step 6 performs the computation of the digital signature of the REAT Signing Input ASCII(BASE64URL(UTF8(JWS Protected Header)) || '.' || BASE64URL(JWS Payload)) using ES256 as the algorithm and the BASE64URL(JWS Signature).

```
dlg7szj0roHsWe8psCzYVl4QdN2b7pQnq8EJhc4j3GOJj2NE6M9Em6aidtycnFJ5mRj3ojiUfVF6rK5RksD0rg
```

Step 8 describes how to create the final REAT token, concatenating the values in the order Header.Payload.Signature with period ('.') characters. For the above example values this would produce the following (with line breaks between period used for readability purposes only):

```
eyJhbGciOiJFUzI1NiIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5leGFtcGxlLmNvbS9yYXQuY2VyIn0
.
eyJhbGciOiJFUzI1NiIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5leGFtcGxlLmNvbS9yYXQuY2VyIn0
.
dlg7szj0roHsWe8psCzYVl4QdN2b7pQnq8EJhc4j3GOJj2NE6M9Em6aidtycnFJ5mRj3ojiUfVF6rK5RksD0rg
```

## A.1. X.509 Private Key in PKCS#8 Format for ES256 Example\*\*

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgeVZzL1gdAFr88hb2
OF/2NxApJCzGCEDdfSp6VQO30hyhRANCAAQRWz+jn65BtOMvdyHKcvjBeBSDZH2r
1RTwjmYSi9R/zpBnuQ4EiMnCqfMPWiZqB4QdbAd0E7oH50VpuZ1P087G
-----END PRIVATE KEY-----
```

## A.2. X.509 Public Key for ES256 Example\*\*

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEEVs/o5+uQbTjL3chynL4wXgUg2R9
q9UU8I5mEovUf86QZ7kOBIjJwqnzDlomagEHWWHdBO6B+dFabmdT9POxg==
-----END PUBLIC KEY-----
```

## Appendix B. Complete JWS JSON Serialization Representation with multiple Signatures

The JWS payload used in this example as follows.

```
{
  "server":{
    "adn":"example.com"
  },
  "iat":1443208345,
  "exp":1443640345,
  "resinfo": {
    "qnameminimization":false
  }
}
```

This would be serialized to the form (with line break used for display purposes only):

```
{"exp":1443640345,"iat":1443208345,"resinfo":{"qnameminimization":false},"server":{"adn":"example.com"}}
```

The JWS protected Header value used for the first signature is same as that used in the example in Appendix A. The X.509 private key used for generating the first signature is same as that used in the example in Appendix A.1.

The JWS Protected Header value used for the second signature is:



```
{
  "alg": "ES384",
  "typ": "rat",
  "x5u": "https://cert.audit-example.com/rat.cer"
}
```

The complete JWS JSON Serialization for these values is as follows  
(with line breaks within values for display purposes only):

```
{
  "payload":
    "eyJhbGciOiJFUzI1NiIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5leGFtcGxlLmNvbS9yYXQuY2VyIn0",
  "signatures": [
    {
      "protected": "eyJhbGciOiJFUzI1NiIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5leGFtcGxlLmNvbS9yYXQuY2VyIn0",
      "signature": "dlg7szj0roHsWe8psCzYVl4QdN2b7pQnq8EJhc4j3GOJj2NE6M9Em6aidtycnFJ5mRj3ojiUfVF6rK5RksD0rg"
    },
    {
      "protected": "eyJhbGciOiJFUzM4NCIsInR5cCI6InJhdCI6InglbSI6Imh0dHBzOi8vY2VydC5hdWRpdC1leGFtcGxlLmNvbS9yYXQuY2VyIn0",
      "signature": "GnKuEEFql_Y8HdZl_mqd027DlziGRXFHvjMoY_ukX-M0k5v2jSLvsQAYOGdKFnt3JY6t938HfBVlonsWerNhgceMJpx5hAsl-xus3fmNY8K1g6QK39hn2Dhbleeeyp0f"
    }
  ]
}
```

#### B.1. X.509 Private Key in PKCS#8 format for E384 Example\*\*

```
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGBYqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgeVZzLlgdAFr88hb2
OF/2NxApJCzGCEDdfSp6VQO30hyhRANCAAQRWz+jn65BtOMvdyHKcvjBeBSDZH2r
1RTwjmYSi9R/zpBnuQ4EiMnCqfMPWiZqB4QdbAd0E7oH50VpuZ1P087G
-----END PRIVATE KEY-----
```

#### B.2. X.509 Public Key for ES384 Example\*\*

```
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEEVs/o5+uQbTjL3chynL4wXgUg2R9
q9UU8I5mEovUf86QZ7kOBIjJwqnzDlomagEHWWHdBO6B+dFabmdT9POxg==
-----END PUBLIC KEY-----
```

Authors' Addresses

Tirumaleswar Reddy  
Akamai  
Embassy Golf Link Business Park  
Bangalore, Karnataka 560071  
India

Email: kondtir@gmail.com

Dan Wing  
Citrix Systems, Inc.  
USA

Email: dwing-ietf@fuggles.com

Michael C. Richardson  
Sandelman Software Works  
USA

Email: mcr+ietf@sandelman.ca

Mohamed Boucadair  
Orange  
Rennes 35000  
France

Email: mohamed.boucadair@orange.com