Stream: Internet Engineering Task Force (IETF)
RFC: 9704
Category: Standards Track
Published: December 2024
ISSN: 2070-1721
Authors: T. Reddy.K *Nokia*
D. Wing *Citrix*
K. Smith *Vodafone*
B. Schwartz *Meta*

RFC 9704
Establishing Local DNS Authority in Validated Split-Horizon Environments

Abstract

When split-horizon DNS is deployed by a network, certain domain names can be resolved authoritatively by a network-provided DNS resolver. DNS clients that are not configured to use this resolver by default can use it for these specific domains only. This specification defines a mechanism for domain owners to inform DNS clients about local resolvers that are authorized to answer authoritatively for certain subdomains.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9704.

Copyright Notice

Table of Contents

1. Introduction

To resolve a DNS query, there are three main behaviors that an implementation can apply: (1) answer from a local database, (2) query the relevant authorities and their parents, or (3) ask a server to query those authorities and return the final answer. Implementations that use these behaviors are called "authoritative nameservers", "full/recursive resolvers", and "forwarders" (or "stub resolvers"), respectively. However, an implementation can also implement a mixture of these behaviors, depending on local policy, for each query. Such an implementation is termed a "hybrid resolver".

Most DNS resolvers are hybrids of some kind. For example, stub resolvers support a local "hosts file" that preempts query forwarding, and most DNS forwarders and full resolvers can also serve responses from a local zone file. Other standardized hybrid resolution behaviors include using a local root [RFC8806], Multicast DNS (mDNS) [RFC6762], and NXDOMAIN synthesis for .onion [RFC7686].

Networks usually offer clients a DNS resolver using means such as DHCP offers or IPv6 Router Advertisements (RAs). Although this resolver is formally specified as a recursive resolver (e.g., see Section 5.1 of [RFC8106]), some networks provide a hybrid resolver instead. If this resolver acts as an authoritative server for some names and -- depending on the source of the query -- provides different answers for those domains, the network is said to be using "split-horizon DNS", because those names resolve in this way only from inside the network.

DNS clients that use pure stub resolution, sending all queries to the network-provided resolver, will always receive the split-horizon results. Conversely, clients that send all queries to a different resolver or implement pure full resolution locally will never receive them. Clients that strictly implement either of these resolution behaviors are out of scope for this specification. Instead, this specification enables hybrid clients to access split-horizon results from a network-provided hybrid resolver, while using a different resolution method for some or all other names.

There are several existing mechanisms for a network to provide clients with "local domain hints", listing domain names that are given special treatment in this network (e.g., "Recursive DNS Server (RDNSS) selection" [RFC6731], "access network domain name" [RFC5986], and "Client Fully Qualified Domain Name (FQDN)" [RFC4702] [RFC4704] in DHCP; "dnsZones" in Provisioning Domains (PvDs) [RFC8801]; and "INTERNAL_DNS_DOMAIN" [RFC8598] in Internet Key Exchange Protocol Version 2 (IKEv2)). However, none of the local domain hint mechanisms enable clients

to determine whether this special treatment is authorized by the domain owner. Instead, these specifications require clients to make their own determinations about whether to trust and rely on these hints.

This document describes a mechanism between domain names, networks, and clients that allows the network to establish its authority over a domain to a client (Section 5). Clients can use this protocol to confirm that a local domain hint was authorized by the domain owner (Section 6), which might influence its processing of that hint. This process requires cooperation between the local DNS zone and the public zone.

This specification expects that local DNS servers will be securely identified and that each local domain hint will be checked against a globally valid parent zone.

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 [RFC9499], e.g., "global DNS". The following additional terms are used throughout this document:

Encrypted DNS: A DNS protocol that provides an encrypted channel between a DNS client and server (e.g., DNS over TLS (DoT) [RFC7858], DNS (queries) over HTTPS (DoH) [RFC8484], DNS over QUIC (DoQ) [RFC9250]).

Encrypted DNS Resolver: Refers to a DNS resolver that supports any encrypted DNS scheme.

Split-Horizon DNS: The DNS service provided by a resolver that also acts as an authoritative server for some names, providing resolution results that are meaningfully different from those in the global DNS. (See the definition of "split DNS" in Section 6 of [RFC9499].)

Validated Split Horizon: Indicates that a split-horizon configuration for some name is considered "validated" if the client has confirmed that a parent of that name has authorized this resolver to serve its own responses for that name. Such authorization generally extends to the entire subtree of names below the authorization point.

In this document, the terms "owner" and "operator" are used interchangeably and refer to the individual or entity responsible for the management and maintenance of domains.

Lone lines in examples are wrapped using a single backslash ("\") per [RFC8792].

3. Scope

The protocol described in this document is designed to support the ability of a domain owner to create or authorize a split-horizon view of their domain. The protocol does not support split-horizon views created by any other entity. Thus, DNS filtering is not enabled by this protocol.

The protocol is applicable to any type of network offering split-horizon DNS configuration. The endpoint does not need any prior configuration to confirm that a local domain hint was indeed authorized by the domain.

All of the Special-Use Domain Names registered with IANA [RFC6761], most notably "home.arpa.", "resolver.arpa.", "ipv4only.arpa.", and "local.", are never unique to a specific DNS server's authority. All Special-Use Domain Names are outside the scope of this document and **MUST NOT** be validated using the mechanism described in this document.

The use of this specification is limited to DNS servers that support authenticated encryption and split-horizon DNS names that are rooted in the global DNS.

4. Requirements

This solution seeks to fulfill the following requirements:

No loss of security: No unauthorized party can impersonate a zone unless they could already do so without the use of this specification.

Least privilege: Local resolvers do not hold any secrets that could weaken the security of the public zone if compromised.

Local zone confidentiality: The specification does not leak local network subdomains to anyone outside of the network.

Flexibility: The specification can represent and authorize a split DNS zone structure.

DNSSEC compatibility: The specification supports DNSSEC-based object security for local zone contents per [RFC9364].

5. Establishing Local DNS Authority

A participating network **MUST** offer one or more encrypted resolvers via DHCP and Router Advertisement options for the Discovery of Network-designated Resolvers (DNR) [RFC9463], Discovery of Designated Resolvers (DDR) [RFC9462], or an equivalent mechanism (see Section 10).

To establish local authority, the network **MUST** convey one or more "authorization claims" to the client. An authorization claim is an abstract structure comprising:

- An Authentication Domain Name (ADN) of a local encrypted resolver.
- The DNS name of the authorizing parent zone.
- A set of subdomains of this parent zone that are claimed by the named local resolver (potentially including the entire parent zone). To claim the entire parent zone, the claimed subdomain will be represented as an asterisk symbol ("*").
- A ZONEMD Hash Algorithm (Section 5.3 of [RFC8976]). For interoperability purposes, implementations **MUST** support the "mandatory to implement" hash algorithms defined in Section 2.2.3 of [RFC8976].
- A high-entropy salt, up to 255 octets.

If the local encrypted resolver is identified by name (e.g., DNR), that identifying name **MUST** be the name used in any corresponding authorization claim. Otherwise (e.g., DDR using IP addresses), the resolver **MUST** present a validatable certificate containing a subjectAltName that matches the authorization claim using the validation techniques for matching as described in [RFC9525].

The network then provides each authorization claim to the parent zone operator. If the contents are approved, the parent zone operator computes a "Verification Token" according to the following procedure:

1. Convert all subdomains into canonical form and sort them in canonical order (Section 6 of [RFC4034]).
2. Replace the suffix corresponding to the parent zone with a zero octet.
3. Let $X be the concatenation of the resulting pseudo-FQDNs.
4. Let len($SALT) be the number of octets of salt, as a single octet.
5. Let $TOKEN = hash(len($SALT) || $SALT || $X), where "||" denotes concatenation and hash is the ZONEMD Hash Algorithm.

The zone operator then publishes a "Verification Record" with the following structure, following the best practices outlined in Sections 5.2 and 5.3 of [DOMAIN-VERIFICATION-TECHNIQUES]:

- Type = TXT
- Owner Name = Concatenation of the ADN, "_splitdns-challenge", and the parent zone name
- Contents = "key/value" pairs, e.g., "token=base64url($TOKEN)" (without padding)

By publishing this record, the parent zone authorizes the local encrypted resolver to serve these subdomains authoritatively.

5.1. Example

Consider the following authorization claim:

- ADN = "resolver17.parent.example"

- Parent = "parent.example"
- Subdomains = "payroll.parent.example", "secret.project.parent.example"
- Hash Algorithm = SHA-384 [RFC6234]
- Salt = "example salt octets (should be random)"

To approve this claim, the zone operator would publish the following record:

```
resolver17.parent.example._splitdns-challenge.parent.example. \
IN TXT "token=z1qyK7QWwQPkT-ZmVW-tAQbsNyYenTNBPp5ogYB8S1wesVCR\
-KJDv2eFwfJcWQM"
```

5.2. Conveying Authorization Claims

The authorization claim is an abstract structure that must be encoded in some concrete syntax in order to convey it from the network to the client. This section defines some encodings of the authorization claims.

5.2.1. Using DHCP

In DHCP, each authorization claim is encoded as a DHCP Authentication option ([RFC3118] and Section 21.11 of [RFC8415]), using the Protocol value 4, "Split-horizon DNS". In DHCPv4 [RFC2131], the mechanism for splitting long options as described in Section 8 of [RFC3396] **MUST** be used if the Authentication option exceeds the maximum DHCPv4 option size of 255 octets. The Algorithm field provides the ZONEMD Hash Algorithm, represented by its registered Value. The Replay Detection Method value **MUST** be 0x00. The Authentication Information **MUST** contain the following information, concatenated:

1. The ADN in canonical form.
2. The parent name in canonical form.
3. A one-octet "salt length" field.
4. The salt value.
5. The $X value as defined in Section 5.

5.2.2. Using Provisioning Domains

When using PvDs [RFC8801], the authorization claims are represented by the PvD Additional Information key "splitDnsClaims", whose value is a JSON array. Each entry in the array **MUST** be a JSON object with the following structure:

"resolver": The ADN as a dot-separated name.

"parent": The parent zone name as a dot-separated name.

"subdomains": An array containing the claimed subdomains, as dot-separated names with the parent suffix already removed, in canonical order. To claim the entire parent zone, the claimed subdomain will be represented as an asterisk symbol ("*").

"algorithm": The hash algorithm, represented by its "Mnemonic" string from the "ZONEMD Hash Algorithms" registry (Section 5.3 of [RFC8976]).

"salt": The salt, encoded in base64url [RFC4648].

Future specifications aiming to define new keys will need to add them to the IANA registry defined in Section 13.3. DNS client implementations will ignore any keys they don't recognize but may also report unknown keys.

6. Validating Authority over Local Domain Hints

To validate an authorization claim provided by the network, DNS clients **MUST** resolve the Verification Record for that name. If the resolution produces an RRset containing the expected token for this claim, the client **SHALL** regard the named resolver as authoritative for the claimed subdomains. Clients **MUST** ignore any unrecognized keys in the Verification Record.

Each validation of authority applies only to a specific ADN. If a network offers multiple encrypted resolvers, each claimed subdomain may be authorized for a distinct subset of the network-provided resolvers.

A zone is termed a "Validated Split-Horizon zone" after successful validation using a "tamperproof" DNS resolution method, i.e., a method that is not subject to interference by the local network operator. Two possible tamperproof resolution methods are presented below.

6.1. Using a Preconfigured External Resolver

This method applies only if the client is already configured with a default resolution strategy that sends queries to a resolver outside of the network over an encrypted transport. That resolution strategy is considered tamperproof because any actor who could modify the response could already modify all of the user's other DNS responses. If the client cannot obtain a response from the external resolver within a reasonable timeout period, it **MUST** consider the verification process to have failed.

To ensure that this assumption holds, clients **MUST NOT** relax the acceptance rules they would otherwise apply when using this resolver. For example, if the client would check the Authenticated Data (AD) bit or validate RRSIGs locally when using this resolver, it must also do so when resolving TXT records for this purpose. Alternatively, a client might perform DNSSEC validation for the verification query even if it has disabled DNSSEC validation for other DNS queries.

6.2. Using DNSSEC

The client resolves the Verification Record using any resolution method of its choice (e.g., querying one of the network-provided resolvers, performing iterative resolution locally) and performs full DNSSEC validation locally [RFC6698]. The result is processed based on its DNSSEC validation state (Section 4.3 of [RFC4035]):

Secure: The response is used for validation.

Bogus or **Indeterminate**: The response is rejected, and validation is considered to have failed.

Insecure: The client **SHOULD** retry the validation process using a different method, such as the method described in Section 6.1, to ensure compatibility with unsigned names. If the client chooses not to retry (e.g., no configured policy to validate the authorization claim using an external resolver), it **MUST** consider validation to have failed.

7. Delegating DNSSEC Across Split DNS Boundaries

When the local zone can be signed with globally trusted keys for the parent zone, support for DNSSEC can be accomplished by simply placing a zone cut at the parent zone and including a suitable DS record for the local resolver's DNSKEY. Zones in this configuration appear the same to validating stubs whether or not they implement this specification.

To enable DNSSEC validation of local DNS names without requiring the local resolver to hold DNSSEC private keys that are valid for the parent zone, parent zones **MAY** add a "ds=..." key to the Verification Record whose value is the RDATA of a single DS record, encoded in base64url. This DS record authorizes a DNSKEY whose owner name is "resolver.arpa."

To validate DNSSEC, the client first fetches and validates the Verification Record. If it is valid and contains a "ds" key, the client **MAY** send a DNSKEY query for "resolver.arpa." to the local encrypted resolver. At least one resulting DNSKEY Resource Record (RR) **MUST** match the DS RDATA from the "ds" key in the Verification Record. All local resolution results for subdomains in this claim **MUST** offer RRSIGs that chain to a DNSKEY whose RDATA is identical to one of these approved DNSKEYs.

The "ds" key **MAY** appear multiple times in a single Verification Record, in order to authorize multiple DNSKEYs for this local encrypted resolver. If the "ds" key is not present in a valid Verification Record, the client **MUST** disable DNSSEC validation when resolving the claimed subdomains via this local encrypted resolver.

Note that in this configuration, any claimed subdomains **MUST** be marked as unsigned in the public DNS. Otherwise, resolution results would be rejected by validating stubs that do not implement this specification.

```
;; Parent zone.
$ORIGIN parent.example.

; Parent zone's public Key Signing Key (KSK)
; and Zone Signing Key (ZSK).
@ IN DNSKEY 257 3 5 ABCD...=
@ IN DNSKEY 256 3 5 DCBA...=

; Verification Record containing DS RDATA for the local
; resolver's KSK.  This is an ordinary public TXT record,
; secured by RRSIGs from the public ZSK.
resolver.example._splitdns-challenge IN TXT "token=abc...,ds=QWE..."

; NSEC record indicating that unsigned delegations are permitted at
; this subdomain.  This is required for compatibility with
; non-split-aware validating stub resolvers.  If the claimed label is
; confidential, the parent zone can conceal it using NSEC3 (with or
; without "opt-out").
@ IN NSEC subdomain.parent.example. NS

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;; Local zone, claiming "subdomain.parent.example".

; The local resolver's KSK, validated by the Verification Record.
; It may not have a corresponding RRSIG.
resolver.arpa. IN DNSKEY 257 3 5 ASDF...=

; Each claimed subdomain duplicates the local resolver's KSK at its
; zone apex and uses it to sign the ZSK.
subdomain.parent.example.        IN DNSKEY 257 3 5 ASDF...=
subdomain.parent.example.        IN DNSKEY 256 3 5 FDSA...=
subdomain.parent.example         IN RRSIG DNSKEY 5 3 ...  \
        (KSK key tag) subdomain.parent.example. ...
subdomain.parent.example.        IN AAAA 2001:db8::17
subdomain.parent.example         IN RRSIG AAAA 5 3 ...    \
        (ZSK key tag) subdomain.parent.example. ...
deeper.subdomain.parent.example. IN AAAA 2001:db8::18
deeper.subdomain.parent.example  IN RRSIG AAAA 5 3 ...    \
        (ZSK key tag) subdomain.parent.example. ...
```

Figure 1: Example Use of "ds=..."

8. Example Split-Horizon DNS Configuration

Consider an organization that operates "example.com" and runs a different version of its global domain on its internal network.

First, the host and network both need to support one of the discovery mechanisms described in Section 5. Figure 2 shows discovery using DNR and PvD information.

Validation is then performed using either an external resolver (Section 8.1) or DNSSEC (Section 8.2).

Steps 1-2: The client determines the network's DNS server (dns.example.net) and PvD information (`pvd.example.com`) using DNR [RFC9463] and PvDs [RFC8801], using one of the following: DNR Router Solicitation, DHCPv4, or DHCPv6.

Steps 3-5: The client connects to dns.example.net using an encrypted transport as indicated in DNR [RFC9463], authenticating the server to its name using TLS (Section 8 of [RFC8310]), and sends it a query for the address of `pvd.example.com`.

Steps 6-7: The client connects to the PvD server, validates its certificate, and retrieves the PvD JSON information indicated by the associated PvD. The PvD contains:

```
{
  "identifier": "pvd.example.com",
  "expires": "2025-05-23T06:00:00Z",
  "prefixes": ["2001:db8:1::/48", "2001:db8:4::/48"],
  "splitDnsClaims": [{
    "resolver": "dns.example.net",
    "parent": "example.com",
    "subdomains": ["*"],
    "algorithm": "SHA384",
    "salt": "abc...123"
  }]
}
```

The JSON keys "identifier", "expires", and "prefixes" are defined in [RFC8801].

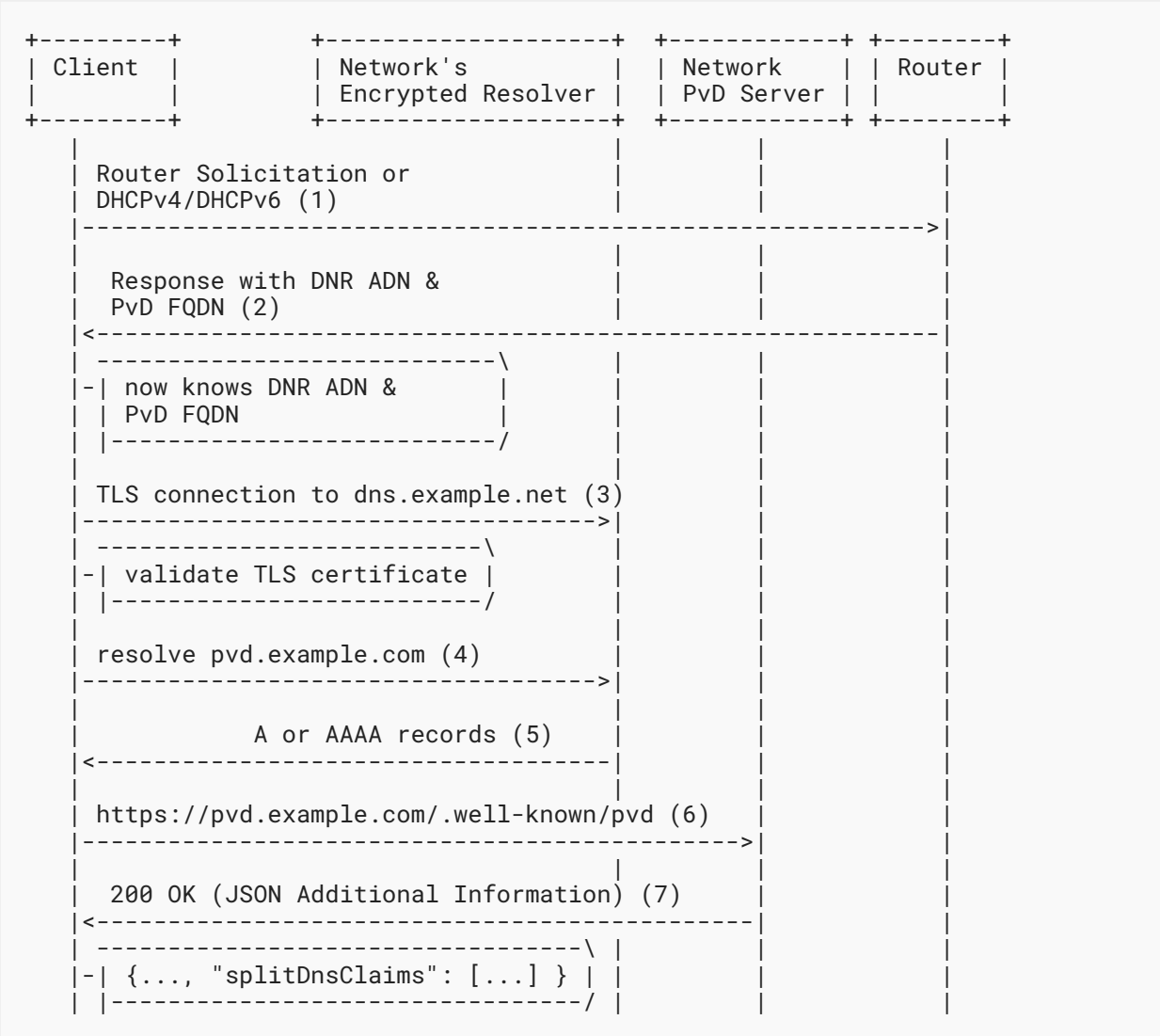

Figure 2: An Example of Learning Local Claims of DNS Authority

8.1. Verification Using an External Resolver

Figure 3 shows the steps performed to verify the local claims of DNS authority using an external resolver.

Steps 1-2: The client uses an encrypted DNS connection to an external resolver to issue TXT queries for the Verification Records. The TXT lookup returns a token that matches the claim.

Step 3: The client has validated that `example.com` has authorized `dns.example.net` to serve `example.com`. When the client connects using an encrypted transport as indicated in DNR [RFC9463], it will authenticate the server to its name using TLS (Section 8 of [RFC8310]) and send queries to resolve any names that fall within the claimed zones.

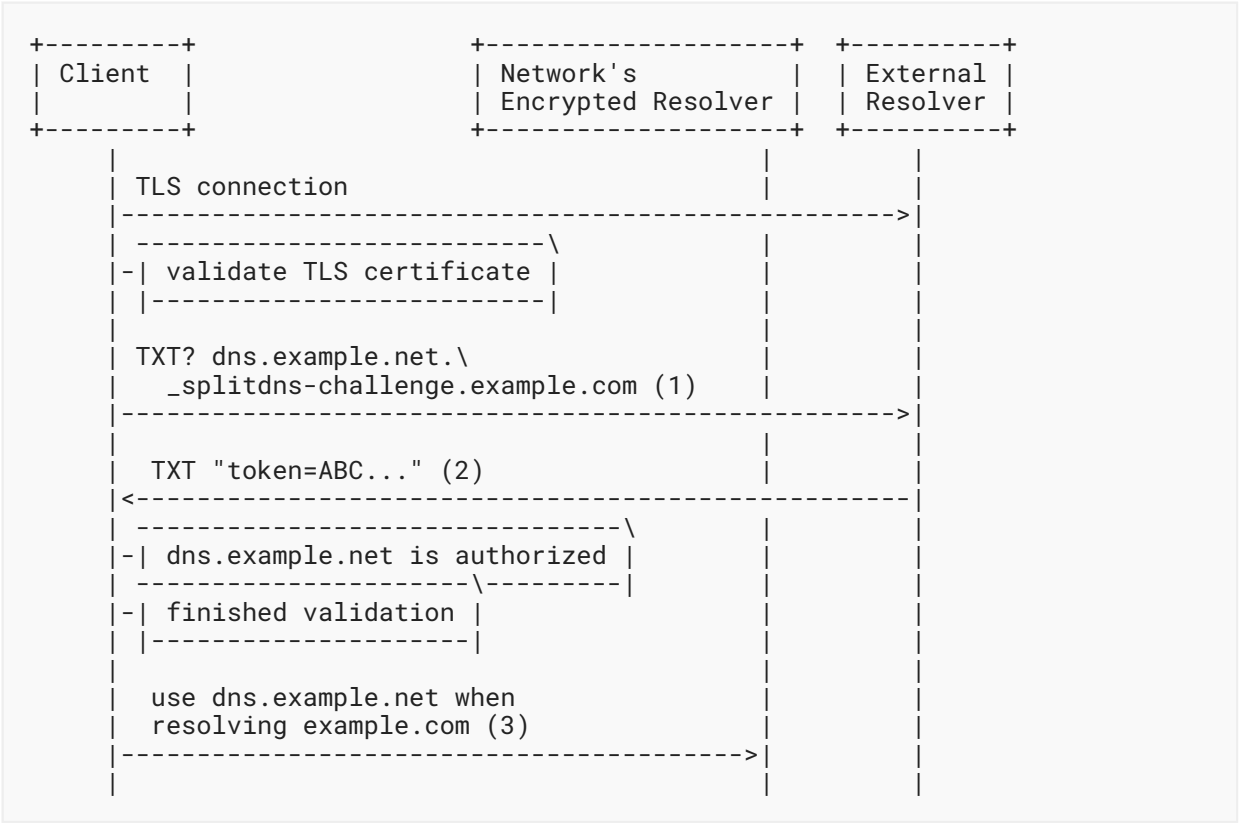

Figure 3: Verifying Claims Using an External Resolver

8.2. Verification Using DNSSEC

Figure 4 shows the steps performed to verify the local claims of DNS authority using DNSSEC.

Steps 1-2: The DNSSEC-validating client queries the network's encrypted resolver to issue TXT queries for the Verification Records. The TXT lookup will return a signed response containing the expected token. The client then performs full DNSSEC validation locally.

Step 3: If the DNSSEC validation is successful and the token matches, then this authorization claim is validated. Once the client connects using an encrypted transport as indicated in DNR [RFC9463], it will authenticate the server to its name using TLS (Section 8 of [RFC8310]) and send queries to resolve any names that fall within the claimed zones.

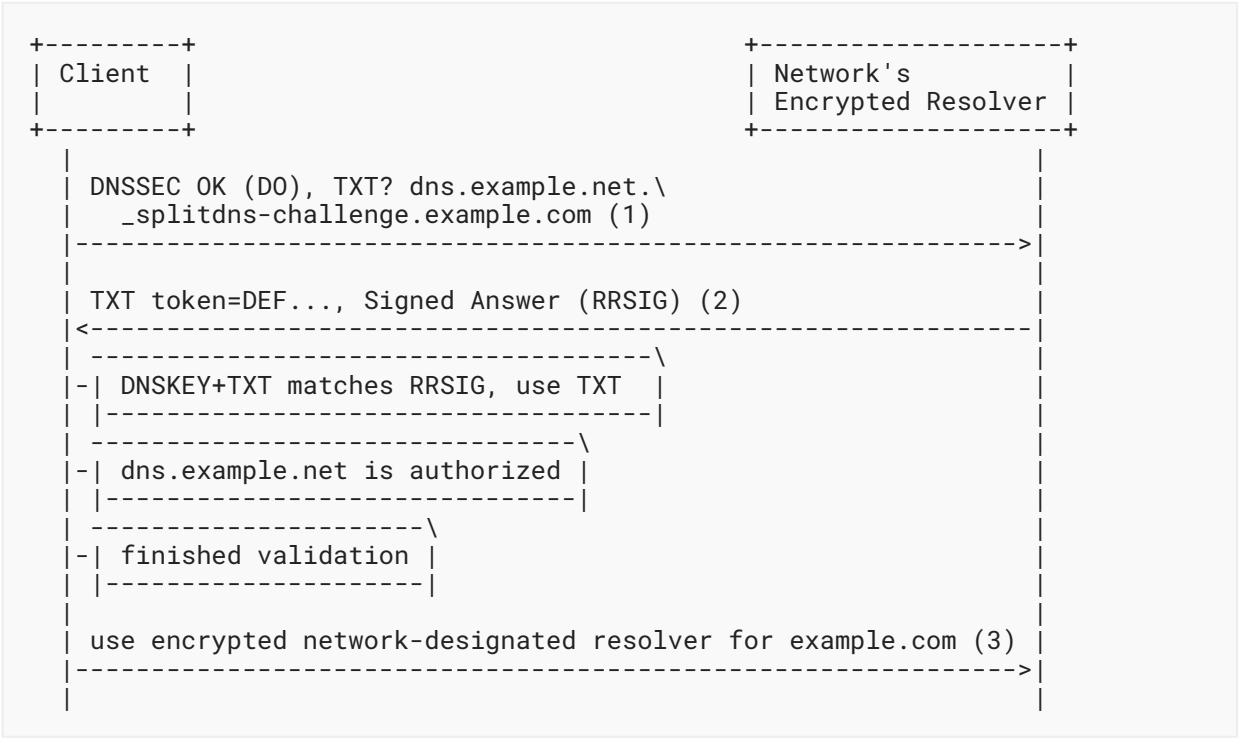

Figure 4: An Example of Verifying Claims Using DNSSEC

9. Operational Efficiency in Split-Horizon Deployments

In many split-horizon deployments, all non-public domain names are placed in a separate child zone (e.g., `internal.example.com`). In this configuration, the message flow is similar to the flow described in Section 8.1, except that queries for hosts not within the subdomain (e.g., `www.example.com`) are sent to the external resolver rather than the resolver for `internal.example.com`.

As specified in Section 8.1, the internal DNS server will need a certificate signed by a Certification Authority (CA) trusted by the client.

Although placing internal domains inside a child domain is unnecessary to prevent leakage, such placement reduces the frequency of changes to the Verification Record. This document recommends that the internal domains be kept in a child zone of the local domain hints advertised by the network. For example, if the PvD "dnsZones" entry is "internal.example.com" and the network-provided DNS resolver is "ns1.internal.example.com", the network operator can structure the internal domain names as "private1.internal.example.com", "private2.internal.example.com", etc. The network-designated resolver will be used to resolve the subdomains of the local domain hint "*.internal.example.com".

10. Validation with IKEv2

When the endpoint is using a VPN tunnel and the tunnel is IPsec, the encrypted DNS resolver hosted by the VPN service provider can be securely discovered by the endpoint using the ENCDNS_IP* IKEv2 Configuration Payload Attribute Types defined in [RFC9464]. The VPN client can use the mechanism defined in Section 6 to validate that the discovered encrypted DNS resolver is authorized to answer for the claimed subdomains.

Other VPN tunnel types have similar configuration capabilities. Note that those capabilities are not discussed in this document.

11. Authorization Claim Update

A Verification Record is only valid until it expires. Expiry occurs when the Time To Live (TTL) or DNSSEC signature validity period ends. Shortly before Verification Record expiry, clients **MUST** fetch the Verification Records again and repeat the verification procedure. This ensures the availability of updated and valid Verification Records.

A new Verification Record must be added to the RRset before the corresponding authorization claim is updated. After the claim is updated, the following procedures can be used:

1. DHCP reconfiguration can be initiated by a DHCP server that has previously communicated with a DHCP client and negotiated for the DHCP client to listen for Reconfigure messages, to prompt the DHCP client to dynamically request the updated authorization claim. This process avoids the need for the client to wait for its current lease to complete and request a new one, enabling the lease renewal to be driven by the DHCP server.
2. The sequence number in the RA PvD option will be incremented, requiring clients to fetch PvD Additional Information from the HTTPS server due to the updated sequence number in the new RA (Section 4.1 of [RFC8801]).
3. The old Verification Record needs to be maintained until the DHCP lease or PvD Additional Information expires.

12. Security Considerations

The ADNs of authorized local encrypted resolvers are revealed in the owner names of Verification Records. This makes it easier for domain owners to understand which resolvers they are currently authorizing to implement split DNS. However, this could create a confidentiality issue if the local encrypted resolver's name contains sensitive information or is part of a secret subdomain. To mitigate the impact of such leakage, local resolvers should be given names that do not reveal any sensitive information.

The security properties of hashing algorithms are not fixed. Algorithm agility (see [RFC7696]) is achieved by providing implementations with the flexibility to choose hashing algorithms from the "ZONEMD Hash Algorithms" registry (Section 5.3 of [RFC8976]).

The entropy of a salt depends on a high-quality pseudorandom number generator. For further discussion on random number generation, see [RFC4086]. The salt **MUST** be regenerated whenever the authorization claim is updated.

13. IANA Considerations

13.1. New DHCP Authentication Algorithm for Split DNS

IANA has added the following entry to the "Protocol Name Space Values" registry in the "Dynamic Host Configuration Protocol (DHCP) Authentication Option Name Spaces" registry group:

Value: 4

Description: Split-horizon DNS

Reference: RFC 9704

13.2. New PvD Additional Information Type for Split DNS

IANA has added the following entry to the "Additional Information PvD Keys" registry in the "Provisioning Domains (PvDs)" registry group:

JSON key: splitDnsClaims

Description: Verifiable locally served domains

Type: Array of Objects

Example:

```
[{
  "resolver": "dns.example.net",
  "parent": "example.com",
  "subdomains": ["sub"],
  "algorithm": "SHA384",
  "salt": "abc...123"
}]
```

Reference: RFC 9704

13.3. New PvD Split DNS Claims Registry

IANA has created a new registry called "PvD Split DNS Claims" within the "Provisioning Domains (PvDs)" registry group. This new registry reserves JSON keys for use in sub-dictionaries under the splitDnsClaims JSON key. The initial contents of this registry, as discussed in Section 5.2.2, are listed below and have been added to the registry:

JSON key	Description	Type	Example	Reference
resolver	The Authentication Domain Name	String	"dns.example.net"	RFC 9704
parent	The parent zone name	String	"example.com"	RFC 9704
subdomains	An array containing the claimed subdomains	Array of Strings	["sub"]	RFC 9704
algorithm	The hash algorithm	String	"SHA384"	RFC 9704
salt	The salt (base64url)	String	"abc...123"	RFC 9704

Table 1: Split DNS Claims

The keys defined in this document are mandatory. Any new assignments of keys will be considered as optional for the purpose of the mechanism described in this document.

New assignments in the "PvD Split DNS Claims" registry will be administered by IANA through Expert Review [RFC8126]. Experts are requested to ensure that defined keys do not overlap in names or semantics.

13.3.1. Guidelines for the Designated Experts

It is suggested that multiple designated experts be appointed for registry change requests.

Criteria that should be applied by the designated experts include determining whether the proposed registration duplicates existing entries and whether the registration description is clear and fits the purpose of this registry.

Registration requests are evaluated within a three-week review period on the advice of one or more designated experts. Within the review period, the designated experts will either approve or deny the registration request, communicating this decision to IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful.

13.4. DNS Underscore Name

IANA has added the following entry to the "Underscored and Globally Scoped DNS Node Names" registry in the "Domain Name System (DNS) Parameters" registry group:

RR Type: TXT

_NODE NAME: _splitdns-challenge

Reference: RFC 9704

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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

[RFC3118] Droms, R., Ed. and W. Arbaugh, Ed., "Authentication for DHCP Messages", RFC 3118, DOI 10.17487/RFC3118, June 2001, <https://www.rfc-editor.org/info/rfc3118>.

[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>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC4086] Eastlake 3rd, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, DOI 10.17487/RFC4086, June 2005, <https://www.rfc-editor.org/info/rfc4086>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC6761] Cheshire, S. and M. Krochmal, "Special-Use Domain Names", RFC 6761, DOI 10.17487/RFC6761, February 2013, <https://www.rfc-editor.org/info/rfc6761>.

[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>.

[RFC8801] Pfister, P., Vyncke, É., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", RFC 8801, DOI 10.17487/RFC8801, July 2020, <https://www.rfc-editor.org/info/rfc8801>.

[RFC8976] Wessels, D., Barber, P., Weinberg, M., Kumari, W., and W. Hardaker, "Message Digest for DNS Zones", RFC 8976, DOI 10.17487/RFC8976, February 2021, <https://www.rfc-editor.org/info/rfc8976>.

[RFC9525] Saint-Andre, P. and R. Salz, "Service Identity in TLS", RFC 9525, DOI 10.17487/RFC9525, November 2023, <https://www.rfc-editor.org/info/rfc9525>.

14.2. Informative References

[DOMAIN-VERIFICATION-TECHNIQUES] Sahib, S. K., Huque, S., Wouters, P., and E. Nygren, "Domain Control Validation using DNS", Work in Progress, Internet-Draft, draft-ietf-dnsop-domain-verification-techniques-06, 21 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-dnsop-domain-verification-techniques-06>.

[RFC4702] Stapp, M., Volz, B., and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option", RFC 4702, DOI 10.17487/RFC4702, October 2006, <https://www.rfc-editor.org/info/rfc4702>.

[RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, DOI 10.17487/RFC4704, October 2006, <https://www.rfc-editor.org/info/rfc4704>.

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, DOI 10.17487/RFC5986, September 2010, <https://www.rfc-editor.org/info/rfc5986>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[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>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

[RFC7686] Appelbaum, J. and A. Muffett, "The ".onion" Special-Use Domain Name", RFC 7686, DOI 10.17487/RFC7686, October 2015, <https://www.rfc-editor.org/info/rfc7686>.

[RFC7696] Housley, R., "Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms", BCP 201, RFC 7696, DOI 10.17487/RFC7696, November 2015, <https://www.rfc-editor.org/info/rfc7696>.

[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>.

[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>.

[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>.

[RFC8598] Pauly, T. and P. Wouters, "Split DNS Configuration for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8598, DOI 10.17487/RFC8598, May 2019, <https://www.rfc-editor.org/info/rfc8598>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

[RFC8806] Kumari, W. and P. Hoffman, "Running a Root Server Local to a Resolver", RFC 8806, DOI 10.17487/RFC8806, June 2020, <https://www.rfc-editor.org/info/rfc8806>.

[RFC9250] Huitema, C., Dickinson, S., and A. Mankin, "DNS over Dedicated QUIC Connections", RFC 9250, DOI 10.17487/RFC9250, May 2022, <https://www.rfc-editor.org/info/rfc9250>.

[RFC9364] Hoffman, P., "DNS Security Extensions (DNSSEC)", BCP 237, RFC 9364, DOI 10.17487/RFC9364, February 2023, <https://www.rfc-editor.org/info/rfc9364>.

[RFC9462] Pauly, T., Kinnear, E., Wood, C. A., McManus, P., and T. Jensen, "Discovery of Designated Resolvers", RFC 9462, DOI 10.17487/RFC9462, November 2023, <https://www.rfc-editor.org/info/rfc9462>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/info/rfc9463>.

[RFC9464] Boucadair, M., Reddy.K, T., Wing, D., and V. Smyslov, "Internet Key Exchange Protocol Version 2 (IKEv2) Configuration for Encrypted DNS", RFC 9464, DOI 10.17487/RFC9464, November 2023, <https://www.rfc-editor.org/info/rfc9464>.

[RFC9499] Hoffman, P. and K. Fujiwara, "DNS Terminology", BCP 219, RFC 9499, DOI 10.17487/RFC9499, March 2024, <https://www.rfc-editor.org/info/rfc9499>.

Acknowledgements

Thanks to Mohamed Boucadair, Jim Reid, Tommy Pauly, Paul Vixie, Michael Richardson, Bernie Volz, Éric Vyncke, and Vinny Parla for the discussion and comments.

Thanks to Tianran Zhou for the opsdir review, Anthony Somerset for the dnsdir review, Watson Ladd for the secdir review, Bob Halley for the intdir review, and Mallory Knodel for the genart review.

Thanks to Mohamed Boucadair for the Shepherd review.

Authors' Addresses

Tirumaleswar Reddy.K
Nokia
India
Email: kondtir@gmail.com

Dan Wing
Citrix Systems, Inc.
4988 Great America Pkwy
Santa Clara, CA 95054
United States of America
Email: danwing@gmail.com

Kevin Smith
Vodafone Group
One Kingdom Street
London
United Kingdom
Email: kevin.smith@vodafone.com

Benjamin Schwartz
Meta Platforms, Inc.
Email: ietf@bemasc.net