Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: 5 September 2024

T. Lemon
S. Cheshire
Apple Inc.

4 March 2024

Service Registration Protocol for DNS-Based Service Discovery
draft-ietf-dnssd-srp-25

Abstract

The Service Registration Protocol for DNS-Based Service Discovery uses the standard DNS Update mechanism to enable DNS-Based Service Discovery using only unicast packets. This makes it possible to deploy DNS Service Discovery without multicast, which greatly improves scalability and improves performance on networks where multicast service is not an optimal choice, particularly IEEE 802.11 (Wi-Fi) and IEEE 802.15.4 networks. DNS-SD Service registration uses public keys and SIG(0) to allow services to defend their registrations.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://dnssd-wg.github.io/draft-ietf-dnssd-srp/draft-ietf-dnssd-srp.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-dnssd-srp/.

Discussion of this document takes place on the DNS-SD Working Group mailing list (mailto:dnssd@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/dnssd/. Subscribe at https://www.ietf.org/mailman/listinfo/dnssd/.

Source for this draft and an issue tracker can be found at https://github.com/dnssd-wg/draft-ietf-dnssd-srp.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction 4
2. Conventions and Terminology Used in This Document 6
3. Service Registration Protocol 6
  3.1. Protocol Variants 7
    3.1.1. Full-featured Hosts 7
    3.1.2. Constrained Hosts 7
    3.1.3. Why two variants? 8
  3.2. Protocol Details 8
    3.2.1. What to publish 8
    3.2.2. Where to publish it 9
    3.2.3. How to publish it 10
      3.2.3.1. How the DNS-SD Service Registration process differs from DNS Update as specified in RFC2136 10
      3.2.3.2. Retransmission Strategy 11
      3.2.3.3. Successive Updates 11
    3.2.4. How to secure it 11
      3.2.4.1. First-Come First-Served Naming 11
    3.2.5. SRP Requestor Behavior 12
      3.2.5.1. Public/Private key pair generation and storage . 12
      3.2.5.2. Name Conflict Handling 13
      3.2.5.3. Record Lifetimes 13
      3.2.5.4. Compression in SRV records 13
      3.2.5.5. Removing published services 14
  3.3. Validation and Processing of SRP Updates 15
    3.3.1. Validation of DNS Update Add and Delete RRs 15

3.3.1.1. Service Discovery Instruction 16
3.3.1.2. Service Description Instruction 17
3.3.1.3. Host Description Instruction 17
3.3.2. Valid SRP Update Requirements 18
3.3.3. FCFS Name And Signature Validation 18
3.3.4. Handling of Service Subtypes 19
3.3.5. SRP Update response 20
3.3.6. Optional Behavior 20
4. TTL Consistency 21
5. Maintenance 21
5.1. Cleaning up stale data 22
6. Security Considerations 23
6.1. Source Validation 24
6.2. Other DNS updates 24
6.3. Risks of allowing arbitrary names to be registered in SRP updates 25
6.4. Security of local service discovery 25
6.5. SRP Registrar Authentication 26
6.6. Required Signature Algorithm 26
7. Privacy Considerations 26
8. Domain Name Reservation Considerations 27
8.1. Users 27
8.2. Application Software 27
8.3. Name Resolution APIs and Libraries 27
8.4. Caching DNS Servers 28
8.5. Authoritative DNS Servers 29
8.6. DNS Server Operators 29
8.7. DNS Registries/Registrars 29
9. Delegation of 'service.arpa.' 29
10. IANA Considerations 29
10.1. Registration and Delegation of 'service.arpa' as a Special-Use Domain Name 30
10.2. Subdomains of 'service.arpa.' 30
10.3. Service Name registrations 30
10.4. 'dnssd-srp' Service Name 31
10.5. 'dnssd-srp-tls' Service Name 31
10.6. Anycast Address 32
11. Implementation Status 32
12. Acknowledgments 33
13. Normative References 33
14. Informative References 36
Appendix A. Testing using standard RFC2136-compliant DNS servers 38
Appendix B. How to allow SRP requestors to update standard RFC2136-compliant servers 39
Appendix C. Sample BIND9 configuration for default.service.arpa. 39
Authors' Addresses 40

1. Introduction

DNS-Based Service Discovery [RFC6763] is a component of Zero Configuration Networking [RFC6760] [ZC] [ROADMAP].

This document describes an enhancement to DNS-Based Service Discovery [RFC6763] (DNS-SD) that allows servers to register the services they offer using the DNS protocol rather than using Multicast DNS [RFC6762] (mDNS). There is already a large installed base of DNS-SD clients that can discover services using the DNS protocol (e.g. Android, Windows, Linux, Apple).

This document is intended for three audiences: implementors of software that provides services that should be advertised using DNS-SD, implementors of DNS servers that will be used in contexts where DNS-SD registration is needed, and administrators of networks where DNS-SD service is required. The document is expected to provide sufficient information to allow interoperable implementation of the registration protocol.

DNS-Based Service Discovery (DNS-SD) allows services to advertise the fact that they provide service, and to provide the information required to access that service. DNS-SD clients can then discover the set of services of a particular type that are available. They can then select a service from among those that are available and obtain the information required to use it. Although DNS Service Discovery (DNS-SD) using the DNS protocol (as opposed to mDNS) can be more efficient and versatile, it is not common in practice, because of the difficulties associated with updating authoritative DNS services with service information.

Existing practice for updating DNS zones is to either manually enter new data, or else use DNS Update [RFC2136]. Unfortunately DNS Update requires either that the authoritative DNS server automatically trust updates, or else that the DNS Update requestor have some kind of shared secret or public key that is known to the DNS server and can be used to authenticate the update. Furthermore, DNS Update can be a fairly chatty process, requiring multiple round trips with different conditional predicates to complete the update process.

The Service Registration Protocol (SRP) adds a set of default heuristics for processing DNS updates that eliminates the need for DNS update conditional predicates: instead, the SRP registrar (a DNS server that supports SRP updates) has a set of default predicates that are applied to the update, and the update either succeeds entirely, or fails in a way that allows the requestor to know what went wrong and construct a new update.

SRP also adds a feature called First-Come, First-Served (FCFS) Naming, which allows the requestor to claim a name that is not yet in use, and, using SIG(0) [RFC2931], to authenticate both the initial claim and subsequent updates. This prevents name conflicts, since a second SRP requestor attempting to claim the same name will not possess the SIG(0) key used by the first requestor to claim it, and so its claim will be rejected and the second requestor will have to choose a new name.

It is important to understand that "authenticate" here just means that we can tell that an update came from the same source as the original registration. We have not established trust. This has important implications for what we can and can't do with data the client sends us. You will notice as you read this document that we only support adding a very restricted set of records, and the content of those records is further constrained.

The reason for this is precisely that we have not established trust. So we can only publish information that we feel safe in publishing even though we do not have any basis for trusting the requestor. We reason that mDNS [RFC6762] allows arbitrary hosts on a single IP link to advertise services [RFC6763], relying on whatever service is advertised to provide authentication as a part of its protocol rather than in the service advertisement.

This is considered reasonably safe because it requires physical presence on the network in order to advertise. An off-network mDNS attack is simply not possible. Our goal with this specification is to impose similar constraints. Because of this you will see in Section 3.3.1 that a very restricted set of records with a very restricted set of relationships are allowed. You will also see in Section 6.1 that we give advice on how to prevent off-network attacks.

This leads us to the disappointing observation that this protocol is not a mechanism for adding arbitrary information to DNS zones. We have not evaluated the security properties of adding, for example, an SOA record, an MX record, or a CNAME record, and so these are forbidden. A future protocol specification might include analyses for other records, and extend the set of records that can be registered here. Or it might require establishment of trust, and add an authorization model to the authentication model we now have. But this is work for a future document.

Finally, SRP adds the concept of a 'lease,' similar to leases in Dynamic Host Configuration Protocol [RFC8415]. The SRP registration itself has a lease which may be on the order of an hour; if the requestor does not renew the lease before it has elapsed, the

registration is removed. The claim on the name can have a longer lease, so that another requestor cannot claim the name, even though the registration has expired.

The Service Registration Protocol for DNS-SD (SRP), specified in this document, provides a reasonably secure mechanism for publishing this information. Once published, these services can be readily discovered by DNS-SD clients using standard DNS lookups.

The DNS-SD specification ([RFC6763], Section 10, Populating the DNS with Information), briefly discusses ways that servers can publish their information in the DNS namespace. In the case of mDNS, it allows servers to publish their information on the local link, using names in the ".local" namespace, which makes their services directly discoverable by peers attached to that same local link.

RFC6763 also allows clients to discover services using the DNS protocol [RFC1035]. This can be done by having a system administrator manually configure service information in the DNS, but manually populating DNS authoritative server databases is costly and potentially error-prone, and requires a knowledgeable network administrator. Consequently, although all DNS-SD client implementations of which we are aware support DNS-SD using DNS queries, in practice it is used much less frequently than mDNS.

The Discovery Proxy [RFC8766] provides one way to automatically populate the DNS namespace, but is only appropriate on networks where services are easily advertised using mDNS. This document describes a solution more suitable for networks where multicast is inefficient, or where sleepy devices are common, by supporting both offering of services, and discovery of services, using unicast.

2. Conventions and Terminology Used in This Document

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.

3. Service Registration Protocol

Services that implement SRP use DNS Update [RFC2136] [RFC3007] to publish service information in the DNS. Two variants exist, one for full-featured hosts, and one for devices designed for "Constrained-Node Networks" [RFC7228]. An SRP registrar is most likely an authoritative DNS server, or else is updating an authoritative DNS server. There is no requirement that the server that is receiving

SRP updates be the same server that is answering queries that return records that have been registered.

3.1. Protocol Variants

3.1.1. Full-featured Hosts

Full-featured hosts either are configured manually with a registration domain, or discover the default registration domain as described in Section 11 of [RFC6763]. If this process does not produce a default registration domain, the Service Registration protocol is not discoverable on the local network using this mechanism. Other discovery mechanisms are possible, but are out of scope for this document.

Manual configuration of the registration domain can be done either by querying the list of available registration domains ("r._dns-sd._udp") and allowing the user to select one from the UI, or by any other means appropriate to the particular use case being addressed. Full-featured devices construct the names of the SRV, TXT, and PTR records describing their service(s) as subdomains of the chosen service registration domain. For these names they then discover the zone apex of the closest enclosing DNS zone using SOA queries Section 6.1 of [RFC8765]. Having discovered the enclosing DNS zone, they query for the "_dnssd-srp._tcp.<zone>" SRV record to discover the server to which they can send SRP updates. Hosts that support SRP Updates using TLS use the "_dnssd-srp-tls._tcp.<zone>" SRV record instead.

Examples of full-featured hosts include devices such as home computers, laptops, powered peripherals with network connections such as printers, home routers, and even battery-operated devices such as mobile phones that have long battery lives.

3.1.2. Constrained Hosts

For devices designed for Constrained-Node Networks [RFC7228] some simplifications are available. Instead of being configured with (or discovering) the service registration domain, the special-use domain name (see [RFC6761]) "default.service.arpa" is used. The details of how SRP registrar(s) are discovered will be specific to the constrained network, and therefore we do not suggest a specific mechanism here.

SRP requestors on constrained networks are expected to receive from the network a list of SRP registrars with which to register. It is the responsibility of a Constrained-Node Network supporting SRP to provide one or more registrar addresses. It is the responsibility of

the registrar supporting a Constrained-Node Network to handle the updates appropriately. In some network environments, updates may be accepted directly into a local "default.service.arpa" zone, which has only local visibility. In other network environments, updates for names ending in "default.service.arpa" may be rewritten by the registrar to names with broader visibility.

3.1.3. Why two variants?

The reason for these different variants is that low-power devices that typically use Constrained-Node Networks may have very limited battery storage. The series of DNS lookups required to discover an SRP registrar and then communicate with it will increase the energy required to advertise a service; for low-power devices, the additional flexibility this provides does not justify the additional use of energy. It is also fairly typical of such networks that some network service information is obtained as part of the process of joining the network, and so this can be relied upon to provide nodes with the information they need.

Networks that are not constrained networks can have more complicated topologies at the IP layer. Nodes connected to such networks can be assumed to be able to do DNS-SD service registration domain discovery. Such networks are generally able to provide registration domain discovery and routing. This creates the possibility of off-network spoofing, where a device from a foreign network registers a service on the local network in order to attack devices on the local network. To prevent such spoofing, TCP is required for such networks.

3.2. Protocol Details

We will discuss several parts to this process: how to know what to publish, how to know where to publish it (under what name), how to publish it, and how to secure its publication. In Section 5, we specify how to maintain the information once published.

3.2.1. What to publish

SRP Updates are sent by SRP requestors to SRP registrars. Three types of instructions appear in an SRP update: Service Discovery instructions, Service Description instructions, and Host Description instructions. These instructions are made up of DNS Update RRs that are either adds or deletes. The types of records that are added, updated and removed in each of these instructions, as well as the constraints that apply to them, are described in Section 3.3. An SRP Update is a DNS Update message that is constructed so as to meet the constraints described in that section. The following is a brief

overview of what is included in a typical SRP Update:

* PTR Resource Record (RR) for services, which map from a generic service type (or subtype) name to a specific Service Instance Name.
* For any Service Instance Name ([RFC6763], Section 4.1), an SRV RR, one or more TXT RRs, and a KEY RR. Although in principle DNS-SD Service Description records can include other record types with the same Service Instance Name, in practice they rarely do. SRP does not permit other record types. The KEY RR is used to support FCFS naming, and has no specific meaning for DNS-SD lookups. SRV records for all services described in an SRP update point to the same hostname.
* There is never more than one hostname in a single SRP update. The hostname has one or more address RRs (AAAA or A) and a KEY RR (used for FCFS naming). Depending on the use case, an SRP requestor may be required to suppress some addresses that would not be usable by hosts discovering the service through the SRP registrar. The exact address record suppression behavior required may vary for different types of SRP requestors. An example of such advice can be found in Section 5.5.2 of [RFC8766].

[RFC6763] describes the details of what each of these types of RR mean, with the exception of the KEY RR, which is defined in [RFC2539]. These RFCs should be considered the definitive source for information about what to publish; the reason for summarizing this here is to provide the reader with enough information about what will be published that the service registration process can be understood at a high level without first learning the full details of DNS-SD. Also, the "Service Instance Name" is an important aspect of FCFS naming, which we describe later on in this document.

3.2.2. Where to publish it

Multicast DNS uses a single namespace, ".local", which is valid on the local link. This convenience is not available for DNS-SD using the DNS protocol: services must exist in some specific DNS namespace that is chosen either by the network operator, or automatically.

As described above, full-featured devices are responsible for knowing the domain in which to register their services. Such devices MAY optionally support configuration of a registration domain by the operator of the device. However, such devices MUST support registration domain discovery as described in Section 11 of [RFC6763], "Discovery of Browsing and Registration Domains".

Devices made for Constrained-Node Networks register in the special use domain name [RFC6761] "default.service.arpa", and let the SRP registrar handle rewriting that to a different domain if necessary.

3.2.3. How to publish it

It is possible to issue a DNS Update that does several things at once; this means that it's possible to do all the work of adding a PTR resource record to the PTR RRset on the Service Name, and creating or updating the Service Instance Name and Host Description, in a single transaction.

An SRP Update takes advantage of this: it is implemented as a single DNS Update message that contains a service's Service Discovery records, Service Description records, and Host Description records.

Updates done according to this specification are somewhat different than regular DNS Updates as defined in [RFC2136]. The [RFC2136] update process can involve many update attempts: you might first attempt to add a name if it doesn't exist; if that fails, then in a second message you might update the name if it does exist but matches certain preconditions. Because the registration protocol uses a single transaction, some of this adaptability is lost.

In order to allow updates to happen in a single transaction, SRP Updates do not include update prerequisites. The requirements specified in Section 3.3 are implicit in the processing of SRP Updates, and so there is no need for the SRP requestor to put in any explicit prerequisites.

3.2.3.1. How the DNS-SD Service Registration process differs from DNS Update as specified in RFC2136

DNS-SD Service Registration is based on standard RFC2136 DNS Update, with some differences:

* It implements first-come first-served name allocation, protected using SIG(0) [RFC2931].
* It enforces policy about what updates are allowed.
* It optionally performs rewriting of "default.service.arpa" to some other domain.
* It optionally performs automatic population of the address-to-name reverse mapping domains.
* An SRP registrar is not required to implement general DNS Update prerequisite processing.
* Constrained-Node SRP requestors are allowed to send updates to the generic domain "default.service.arpa."

3.2.3.2. Retransmission Strategy

The DNS protocol, including DNS updates, can operate over UDP or TCP. When using UDP, reliable transmission must be guaranteed by retransmitting if a DNS UDP message is not acknowledged in a reasonable interval. Section 4.2.1 of [RFC1035] provides some guidance on this topic, as does Section 1 of [RFC1536]. Section 3.1.3 of [RFC8085] also provides useful guidance that is particularly relevant to DNS.

3.2.3.3. Successive Updates

Service Registration Protocol does not require that every update contain the same information. When an SRP requestor needs to send more than one SRP update to the SRP registrar, it MUST send these sequentially: until an earlier update has been successfully acknowledged, the requestor MUST NOT begin sending a subsequent update.

3.2.4. How to secure it

DNS update as described in [RFC2136] is secured using Secret Key Transaction Signatures, [RFC8945], which uses a secret key shared between the DNS Update requestor (which issues the update) and the server (which authenticates it). This model does not work for automatic service registration.

The goal of securing the DNS-SD Registration Protocol is to provide the best possible security given the constraint that service registration has to be automatic. It is possible to layer more operational security on top of what we describe here, but FCFS naming is already an improvement over the security of mDNS.

3.2.4.1. First-Come First-Served Naming

First-Come First-Serve naming provides a limited degree of security: a server that registers its service using DNS-SD Registration protocol is given ownership of a name for an extended period of time based on a lease specific to the key used to authenticate the DNS Update, which may be longer than the lease associated with the registered records. As long as the registration service remembers the name and the key used to register that name, no other server can add or update the information associated with that. If the server fails to renew its service registration before the KEY lease (Section 4 of [I-D.ietf-dnssd-update-lease]) expires, its name is no longer protected. FCFS naming is used to protect both the Service Description and the Host Description.

3.2.5. SRP Requestor Behavior

3.2.5.1. Public/Private key pair generation and storage

The requestor generates a public/private key pair (See Section 6.6). This key pair MUST be stored in stable storage; if there is no writable stable storage on the SRP requestor, the SRP requestor MUST be pre-configured with a public/private key pair in read-only storage that can be used. This key pair MUST be unique to the device. A device with rewritable storage SHOULD retain this key indefinitely. When the device changes ownership, it may be appropriate for the former owner to erase the old key pair, which would then require the new owner to install a new one. Therefore, the SRP requestor on the device SHOULD provide a mechanism to erase the key, for example as the result of a "factory reset," and to generate a new key.

The policy described here for managing keys assumes that the keys are only used for SRP. If a key that is used for SRP is also used for other purposes, the policy described here is likely to be insufficient. The policy stated here is NOT RECOMMENDED in such a situation: a policy appropriate to the full set of uses for the key must be chosen. Specifying such a policy is out of scope for this document.

When sending DNS updates, the requestor includes a KEY record containing the public portion of the key in each Host Description Instruction and each Service Description Instruction. Each KEY record MUST contain the same public key. The update is signed using SIG(0), using the private key that corresponds to the public key in the KEY record. The lifetimes of the records in the update is set using the EDNS(0) Update Lease option [I-D.ietf-dnssd-update-lease].

The format of the KEY resource record in the SRP Update is defined in [RFC3445]. Because the KEY RR used in TSIG is not a zone-signing key, the flags field in the KEY RR MUST be all zeroes.

The KEY record in Service Description updates MAY be omitted for brevity; if it is omitted, the SRP registrar MUST behave as if the same KEY record that is given for the Host Description is also given for each Service Description for which no KEY record is provided. Omitted KEY records are not used when computing the SIG(0) signature.

3.2.5.2. Name Conflict Handling

Both Host Description RR adds and Service Description RR adds can have names that result in name conflicts. Service Discovery record adds cannot have name conflicts. If any Host Description or Service Description record is found by the SRP registrar to have a conflict with an existing name, the registrar will respond to the SRP Update with a YXDomain RCODE (Section 2.2 of [RFC2136]). In this case, the requestor MUST choose a new name or give up.

There is no specific requirement for how this is done; typically, however, the requestor will append a number to the preferred name. This number could be sequentially increasing, or could be chosen randomly. One existing implementation attempts several sequential numbers before choosing randomly. So for instance, it might try host.default.service.arpa, then host-1.default.service.arpa, then host-2.default.service.arpa, then host-31773.default.service.arpa.

3.2.5.3. Record Lifetimes

The lifetime of the DNS-SD PTR, SRV, A, AAAA and TXT records [RFC6763] uses the LEASE field of the Update Lease option, and is typically set to two hours. This means that if a device is disconnected from the network, it does not appear in the user interfaces of devices looking for services of that type for too long.

The lifetime of the KEY records is set using the KEY-LEASE field of the Update Lease Option, and SHOULD be set to a much longer time, typically 14 days. The result of this is that even though a device may be temporarily unplugged, disappearing from the network for a few days, it makes a claim on its name that lasts much longer.

This means that even if a device is unplugged from the network for a few days, and its services are not available for that time, no other device can come along and claim its name the moment it disappears from the network. In the event that a device is unplugged from the network and permanently discarded, then its name is eventually cleaned up and made available for re-use.

3.2.5.4. Compression in SRV records

Although [RFC2782] requires that the target name in the SRV record not be compressed, an SRP requestor MAY compress the target in the SRV record. The motivation for _not_ compressing in [RFC2782] is not stated, but is assumed to be because a caching resolver that does not understand the format of the SRV record might store it as binary data and thus return an invalid pointer in response to a query. This does not apply in the case of SRP: an SRP registrar needs to understand

SRV records in order to validate the SRP Update. Compression of the target can save space in the SRP Update, so we want clients to be able to assume that the registrar will handle this. Therefore, SRP registrars MUST support compression of SRV RR targets.

Note that this does not update [RFC2782]: DNS servers still MUST NOT compress SRV record targets. The requirement to accept compressed SRV records in updates only applies to SRP registrars, and SRP registrars that are also DNS servers still MUST NOT compress SRV record targets in DNS responses. We note also that [RFC6762] recomments that SRV records be compressed in mDNS messages, so [RFC2782] does not apply to mDNS messages.

In addition, we note that an implementor of an SRP requestor might update existing code that creates SRV records or compresses DNS messages so that it compresses the target of an SRV record. Care must be taken if such code is used both in requestors and in DNS servers that the code only compresses in the case where a requestor is generating an SRP update.

3.2.5.5. Removing published services

3.2.5.5.1. Removing all published services

To remove all the services registered to a particular host, the SRP requestor transmits an SRP update for that host with an Update Lease option that has a LEASE value of zero. If the registration is to be permanently removed, KEY-LEASE SHOULD also be zero. Otherwise, it SHOULD be set to the same value it had previously; this holds the name in reserve for when the SRP requestor is once again able to provide the service.

SRP requestors are normally expected to remove all service instances when removing a host. However, in some cases an SRP requestor may not have retained sufficient state to know that some service instance is pointing to a host that it is removing. This method of removing services is intended for the case where the requestor is going offline and does not want its services advertised. Therefore, it is sufficient for the requestor to send the Host Description Instruction (Section 3.3.1.3).

To support this, when removing services based on the lease time being zero, an SRP registrar MUST remove all service instances pointing to a host when a host is removed, even if the SRP requestor doesn't list them explicitly. If the KEY lease time is nonzero, the SRP registrar MUST NOT delete the KEY records for these SRP requestors.

3.2.5.5.2. Removing some published services

In some use cases a requestor may need to remove some specific service, without removing its other services. This can be accomplished in one of two ways. To simply remove a specific service, the requestor sends a valid SRP Update where the Service Discovery Instruction (Section 3.3.1.1) contains a single Delete an RR from an RRset ([RFC2136], Section 2.5.4) update that deletes the PTR record whose target is the service instance name. The Service Description Instruction (Section 3.3.1.2) in this case contains a single Delete all RRsets from a Name ([RFC2136], Section 2.5.3) update to the service instance name.

The second alternative is used when some service is being replaced by a different service with a different service instance name. In this case, the old service is deleted as in the first alternative. The new service is added, just as it would be in an update that wasn't deleting the old service. Because both the removal of the old service and the add of the new service consist of a valid Service Discovery Instruction and a valid Service Description Instruction, the update as a whole is a valid SRP Update, and will result in the old service being removed and the new one added, or, to put it differently, in the old service being replaced by the new service.

It is perhaps worth noting that if a service is being updated without the service instance name changing, that will look very much like the second alternative above. The difference is that because the target for the PTR record in the Service Discovery Instruction is the same for both the Delete An RR From An RRset update and the Add To An RRSet update, there is no way to tell whether they were intended to be one or two Instructions. The same would be true of the Service Description Instruction.

Whichever of these two alternatives is used, the host lease will be updated with the lease time provided in the SRP update. In neither of these cases is it permissible to delete the host. All services must point to a host. If a host is to be deleted, this must be done using the method described in Section 3.2.5.5.1, which deletes the host and all services that have that host as their target.

3.3. Validation and Processing of SRP Updates

3.3.1. Validation of DNS Update Add and Delete RRs

The SRP registrar first validates that the DNS Update is a syntactically and semantically valid DNS Update according to the rules specified in [RFC2136].

SRP Updates consist of a set of _instructions_ that together add or remove one or more services. Each instruction consists of some combination of delete updates and add updates. When an instruction contains a delete and an add, the delete MUST precede the add.

The SRP registrar checks each instruction in the SRP Update to see that it is either a Service Discovery Instruction, a Service Description Instruction, or a Host Description Instruction. Order matters in DNS updates. Specifically, deletes must precede adds for records that the deletes would affect; otherwise the add will have no effect. This is the only ordering constraint; aside from this constraint, updates may appear in whatever order is convenient when constructing the update.

Because the SRP Update is a DNS update, it MUST contain a single question that indicates the zone to be updated. Every delete and update in an SRP Update MUST be within the zone that is specified for the SRP Update.

3.3.1.1. Service Discovery Instruction

An instruction is a Service Discovery Instruction if it contains

* exactly one "Add to an RRSet" ([RFC2136], Section 2.5.1) or exactly one "Delete an RR from an RRSet" ([RFC2136], Section 2.5.4) RR update,
* which updates a PTR RR,
* the target of which is a Service Instance Name
* for which name a Service Description Instruction is present in the SRP Update, and:
  - if the RR Update is an "Add to an RRSet" instruction, that Service Description Instruction contains an "Add to an RRset" RR update for the SRV RR describing that service and no other "Delete from an RRset" instructions for that Service Instance Name; or
  - if the RR Update is a "Delete an RR from an RRSet" instruction, that Service Description Instruction contains a "Delete from an RRset" RR update and no other "Add to an RRset" instructions for that Service Instance Name.
* and contains no other add or delete RR updates for the same name as the PTR RR Update.

Note that there can be more than one Service Discovery Instruction for the same name if the SRP requestor is advertising more than one service of the same type, or is changing the target of a PTR RR. This is also true for SRP subtypes (Section 7.1 of [RFC6763]). For each such PTR RR add or delete, the above constraints must be met.

3.3.1.2. Service Description Instruction

An instruction is a Service Description Instruction if, for the appropriate Service Instance Name, the following are true:

* It contains exactly one "Delete all RRsets from a name" update for the service instance name ([RFC2136], Section 2.5.3),
* It contains zero or one "Add to an RRset" SRV RR,
* It contains zero or one "Add to an RRset" KEY RR that, if present, contains the public key corresponding to the private key that was used to sign the message (if present, the KEY MUST match the KEY RR given in the Host Description),
* It contains zero or more "Add to an RRset" TXT RRs,
* If there is one "Add to an RRset" SRV update, there MUST be at least one "Add to an RRset" TXT update.
* The target of the SRV RR Add, if present points to a hostname for which there is a Host Description Instruction in the SRP Update, or
* If there is no "Add to an RRset" SRV RR, then either:
  - the name to which the "Delete all RRsets from a name" applies does not exist, or

  - there is an existing KEY RR on that name, which matches the key with which the SRP Update was signed.
* No other resource records on the Service Instance Name are modified.

An SRP registrar MUST correctly handle compressed names in the SRV target.

3.3.1.3. Host Description Instruction

An instruction is a Host Description Instruction if, for the appropriate hostname, it contains

* exactly one "Delete all RRsets from a name" RR,
* one or more "Add to an RRset" RRs of type A and/or AAAA,
* exactly one "Add to an RRset" RR that adds a KEY RR that contains the public key corresponding to the private key that was used to sign the message,
* Host Description Instructions do not modify any other resource records.

A and/or AAAA records that are not of sufficient scope to be validly published in a DNS zone MAY be ignored by the SRP registrar, which could result in a host description effectively containing zero reachable addresses even when it contains one or more addresses.

For example, if a link-scope address or IPv4 autoconfiguration address is provided by the SRP requestor, the SRP registrar could not publish this in a DNS zone. However, in some situations, the registrar might make the records available through a mechanism such as an advertising proxy only on the specific link from which the SRP update originated; in such a situation, locally-scoped records are still valid.

3.3.2. Valid SRP Update Requirements

An SRP Update MUST contain exactly one Host Description Instruction. In addition, there MUST NOT be any Service Description Instruction to which no Service Discovery Instruction points. A DNS Update that contains any additional adds or deletes that cannot be identified as Service Discovery, Service Description or Host Description Instructions is not an SRP Update. A DNS update that contains any prerequisites is not an SRP Update.

An SRP Update MUST include an EDNS(0) Update Lease option [I-D.ietf-dnssd-update-lease]. The LEASE time specified in the Update Lease option MUST be less than or equal to the KEY-LEASE time. A DNS update that does not include the Update Lease option, or that includes a KEY-LEASE value that is less than the LEASE value, is not an SRP update.

When an SRP registrar receives a DNS Update that is not an SRP update, it MAY process the update as regular RFC2136 updates, including access control checks and constraint checks, if supported. Otherwise the SRP registrar MUST reject the DNS Update with the Refused RCODE.

If the definitions of each of these instructions are followed carefully and the update requirements are validated correctly, many DNS Updates that look very much like SRP Updates nevertheless will fail to validate. For example, a DNS update that contains an Add to an RRset instruction for a Service Name and an Add to an RRset instruction for a Service Instance Name, where the PTR record added to the Service Name does not reference the Service Instance Name, is not a valid SRP Update message, but may be a valid RFC2136 update.

3.3.3. FCFS Name And Signature Validation

Assuming that a DNS Update message has been validated with these conditions and is a valid SRP Update, the SRP registrar checks that the name in the Host Description Instruction exists. If so, then the registrar checks to see if the KEY record on that name is the same as the KEY record in the Host Description Instruction. The registrar performs the same check for the KEY records in any Service

Description Instructions. For KEY records that were omitted from Service Description Instructions, the KEY from the Host Description Instruction is used. If any existing KEY record corresponding to a KEY record in the SRP Update does not match the KEY record in the SRP Update (whether provided or taken from the Host Description Instruction), then the SRP registrar MUST reject the SRP Update with the YXDomain RCODE.

Otherwise, the SRP registrar validates the SRP Update using SIG(0) against the public key in the KEY record of the Host Description Instruction. If the validation fails, the registrar MUST reject the SRP Update with the Refused RCODE. Otherwise, the SRP Update is considered valid and authentic, and is processed according to the method described in RFC2136.

KEY record updates omitted from Service Description Instruction are processed as if they had been explicitly present: every Service Description that is updated MUST, after the SRP Update has been applied, have a KEY RR, and it must be the same KEY RR that is present in the Host Description to which the Service Description refers.

[RFC3445] states that the flags field in the KEY RR MUST be zero except for bit 7, which can be one in the case of a zone key. However, the SRP registrar MUST NOT validate the flags field.

3.3.4. Handling of Service Subtypes

SRP registrars MUST treat the update instructions for a service type and all its subtypes as atomic. That is, when a service and its subtypes are being updated, whatever information appears in the SRP Update is the entirety of information about that service and its subtypes. If any subtype appeared in a previous update but does not appear in the current update, then the SRP registrar MUST remove that subtype.

Similarly, there is no mechanism for deleting subtypes. A delete of a service deletes all of its subtypes. To delete an individual subtype, an SRP Update must be constructed that contains the service type and all subtypes for that service except for the one to be deleted.

3.3.5. SRP Update response

The status that is returned depends on the result of processing the update, and can be either NoError, ServFail, Refused or YXDomain: all other possible outcomes will already have been accounted for when applying the constraints that qualify the update as an SRP Update. The meanings of these responses are explained in Section 2.2 of [RFC2136].

In the case of a response other than NoError, Section 3.8 of [RFC2136] specifies that the server is permitted to respond either with no RRs or to copy the RRs sent by the client into the response. The SRP Requestor MUST NOT attempt to validate any RRs that are included in the response. It is possible that a future SRP extension may include per-RR indications as to why the update failed, but at present this is not specified, so if a client were to attempt to validate the RRs in the response, it might reject such a response, since it would contain RRs, but probably not a set of RRs identical to what was sent in the SRP Update.

3.3.6. Optional Behavior

The SRP registrar MAY add a Reverse Mapping (Section 3.5 of [RFC1035], Section 2.5 of [RFC3596]) that corresponds to the Host Description. This is not required because the Reverse Mapping serves no protocol function, but it may be useful for debugging, e.g. in annotating network packet traces or logs. In order for the registrar to do a reverse mapping update, it must be authoritative for the zone that would need to be updated, or have credentials to do the update. The SRP requestor MAY also do a reverse mapping update if it has credentials to do so.

The SRP registrar MAY apply additional criteria when accepting updates. In some networks, it may be possible to do out-of-band registration of keys, and only accept updates from pre-registered keys. In this case, an update for a key that has not been registered SHOULD be rejected with the Refused RCODE.

There are at least two benefits to doing this rather than simply using normal SIG(0) DNS updates. First, the same registration protocol can be used in both cases, so both use cases can be addressed by the same SRP requestor implementation. Second, the registration protocol includes maintenance functionality not present with normal DNS updates.

Note that the semantics of using SRP in this way are different than for typical RFC2136 implementations: the KEY used to sign the SRP Update only allows the SRP requestor to update records that refer to its Host Description. RFC2136 implementations do not normally provide a way to enforce a constraint of this type.

The SRP registrar could also have a dictionary of names or name patterns that are not permitted. If such a list is used, updates for Service Instance Names that match entries in the dictionary are rejected with a Refused RCODE.

4. TTL Consistency

All RRs within an RRset are required to have the same TTL (Clarifications to the DNS Specification [RFC2181], Section 5.2). In order to avoid inconsistencies, SRP places restrictions on TTLs sent by requestors and requires that SRP registrars enforce consistency.

Requestors sending SRP Updates MUST use consistent TTLs in all RRs within the SRP Update.

SRP registrars MUST check that the TTLs for all RRs within the SRP Update are the same. If they are not, the SRP update MUST be rejected with a Refused RCODE.

Additionally, when adding RRs to an RRset, for example when processing Service Discovery records, the SRP registrar MUST use the same TTL on all RRs in the RRset. How this consistency is enforced is up to the implementation.

TTLs sent in SRP Updates are advisory: they indicate the SRP requestor's guess as to what a good TTL would be. SRP registrars may override these TTLs. SRP registrars SHOULD ensure that TTLs are reasonable: neither too long nor too short. The TTL SHOULD NOT ever be longer than the lease time (Section 5.1). Shorter TTLs will result in more frequent data refreshes; this increases latency on the DNS-SD client side, increases load on any caching resolvers and on the authoritative server, and also increases network load, which may be an issue for constrained networks. Longer TTLs will increase the likelihood that data in caches will be stale. TTL minimums and maximums SHOULD be configurable by the operator of the SRP registrar.

5. Maintenance

5.1. Cleaning up stale data

Because the DNS-SD registration protocol is automatic, and not managed by humans, some additional bookkeeping is required. When an update is constructed by the SRP requestor, it MUST include an EDNS(0) Update Lease Option [I-D.ietf-dnssd-update-lease]. The Update Lease Option contains two lease times: the Lease Time and the KEY Lease Time.

These leases are promises, similar to DHCP leases [RFC2131], from the SRP requestor that it will send a new update for the service registration before the lease time expires. The Lease time is chosen to represent the time after the update during which the registered records other than the KEY record can be assumed to be valid. The KEY lease time represents the time after the update during which the KEY record can be assumed to be valid.

The reasoning behind the different lease times is discussed in the section on FCFS naming (Section 3.2.4.1). SRP registrars may be configured with limits for these values. A default limit of two hours for the Lease and 14 days for the SIG(0) KEY are currently thought to be good choices. Constrained devices with limited battery that wake infrequently are likely to request longer leases; registrars that support such devices may need to set higher limits. SRP requestors that are going to continue to use names on which they hold leases SHOULD update well before the lease ends, in case the registrar is unavailable or under heavy load.

The lease time applies specifically to the host. All service instances, and all service entries for such service instances, depend on the host. When the lease on a host expires, the host and all services that reference it MUST be removed at the same timeit is never valid for a service instance to remain when the host it references has been removed. If the KEY record for the host is to remain, the KEY record for any services that reference it MUST also remain. However, the service PTR record MUST be removed, since it has no key associated with it, and since it is never valid to have a service PTR record for which there is no service instance on the target of the PTR record.

SRP registrars MUST also track a lease time per service instance. The reason for doing this is that a requestor may re-register a host with a different set of services, and not remember that some different service instance had previously been registered. In this case, when that service instance lease expires, the SRP registrar MUST remove the service instance (although the KEY record for the service instance SHOULD be retained until the KEY lease on that service expires). This is beneficial because otherwise if the SRP requestor continues to renew the host, but never mentions the stale service again, the stale service will continue to be advertised.

The SRP registrar MUST include an EDNS(0) Update Lease option in the response if the lease time proposed by the requestor has been shortened or lengthened by the registrar. The requestor MUST check for the EDNS(0) Update Lease option in the response and MUST use the lease times from that option in place of the options that it sent to the registrar when deciding when to renew its registration. The times may be shorter or longer than those specified in the SRP Update; the SRP requestor must honor them in either case.

SRP requestors SHOULD assume that each lease ends N seconds after the update was first transmitted, where N is the lease duration. SRP Registrars SHOULD assume that each lease ends N seconds after the update that was successfully processed was received. Because the registrar will always receive the update after the SRP requestor sent it, this avoids the possibility of misunderstandings.

SRP registrars MUST reject updates that do not include an EDNS(0) Update Lease option. DNS authoritative servers that allow both SRP and non-SRP DNS updates MAY accept updates that don't include leases, but SHOULD differentiate between SRP Updates and other updates, and MUST reject updates that would otherwise be SRP Updates if they do not include leases.

Lease times have a completely different function than TTLs. On an authoritative DNS server, the TTL on a resource record is a constant: whenever that RR is served in a DNS response, the TTL value sent in the answer is the same. The lease time is never sent as a TTL; its sole purpose is to determine when the authoritative DNS server will delete stale records. It is not an error to send a DNS response with a TTL of 'n' when the remaining time on the lease is less than 'n'.

6. Security Considerations

6.1. Source Validation

SRP Updates have no authorization semantics other than FCFS. This means that if an attacker from outside of the administrative domain of the SRP registrar knows the registrar's IP address, it can in principle send updates to the registrar that will be processed successfully. SRP Registrars SHOULD therefore be configured to reject updates from source addresses outside of the administrative domain of the registrar.

For TCP updates, the initial SYN-SYN+ACK handshake prevents updates being forged by an off-network attacker. In order to ensure that this handshake happens, SRP registrars relying on three-way-handshake validation MUST NOT accept TCP Fast Open [RFC7413] payloads. If the network infrastructure allows it, an SRP registrar MAY accept TCP Fast Open payloads if all such packets are validated along the path, and the network is able to reject this type of spoofing at all ingress points.

For UDP updates from constrained devices, spoofing would have to be prevented with appropriate source address filtration on routers [RFC2827]. This would ordinarily be accomplished by measures such as are described in Section 4.5 of [RFC7084]. For example, a stub router [I-D.ietf-snac-simple] for a constrained network might only accept UDP updates from source addresses known to be on-link on that stub network, and might further validate that the UDP update was actually received on the stub network interface and not the interface connected to the adjacent infrastructure link.

6.2. Other DNS updates

Note that these rules only apply to the validation of SRP Updates. A server that accepts updates from SRP requestors may also accept other DNS updates, and those DNS updates may be validated using different rules. However, in the case of a DNS server that accepts SRP updates, the intersection of the SRP Update rules and whatever other update rules are present must be considered very carefully.

For example, a normal, authenticated DNS update to any RR that was added using SRP, but that is authenticated using a different key, could be used to override a promise made by the SRP registrar to an SRP requestor, by replacing all or part of the service registration information with information provided by an authenticated DNS update requestor. An implementation that allows both kinds of updates SHOULD NOT allow DNS Update requestors that are using different authentication and authorization credentials to update records added by SRP requestors.

6.3. Risks of allowing arbitrary names to be registered in SRP updates

It is possible to set up SRP updates for a zone that is used for non-DNSSD services. For example, imagine that you set up SRP service for example.com. SRP hosts can now register names like "www" or "mail" or "smtp" in this domain. In addition, SRP updates using FCFS naming can insert names that are obscene or offensive into the zone. There is no simple solution to these problems. We have two recommendations to address this problem, however:

* Do not provide SRP service in organization-level zones. Use subdomains of the organizational domain for DNS service discovery. This does not prevent registering names as mentioned above, but does ensure that genuinely important names are not accidentally reserved for SRP clients. So for example, the zone "dnssd.example.com" could be used instead of "example.com" for SRP updates. Because of the way that DNS browsing domains are discovered, there is no need for the DNSSD discovery zone that is updated by SRP to have a user-friendly or important-sounding name.
* Configure a dictionary of names that are prohibited. Dictionaries of common obscene and offensive names are no doubt available, and can be augmented with a list of typical "special" names like "www", "mail", "smtp" and so on. Lists of names are generally available, or can be constructed manually.

6.4. Security of local service discovery

Local links can be protected by managed services such as RA Guard [RFC6105], but multicast services like DHCP [RFC2131], DHCPv6 [RFC8415] and IPv6 Neighbor Discovery [RFC4861] are in most cases not authenticated and can’t be controlled on unmanaged networks, such as home networks and small-office networks where no network management staff are present. In such situations, the SRP service has comparatively fewer potential security exposures and hence is not the weak link. This is discussed in more detail in Section 3.2.4.

The fundamental protection for networks of this type is the user’s choice of what devices to add to the network. Work is being done in other working groups and standards bodies to improve the state of the art for network on-boarding and device isolation (e.g., [RFC8520] provides a means for constraining what behaviors are allowed for a device in an automatic way), but such work is out of scope for this document.

6.5. SRP Registrar Authentication

This specification does not provide a mechanism for validating responses from SRP Registrars to SRP requestors. In principle, a KEY RR could be used by a non-constrained SRP requestor to validate responses from the registrar, but this is not required, nor do we specify a mechanism for determining which key to use.

In addition, for DNS-over-TLS connections, out-of-band key pinning as described in [RFC7858], Section 4.2 could be used for authentication of the SRP registrar, e.g. to prevent man-in-the-middle attacks. However the use of such keys is impractical for an unmanaged service registration protocol, and hence is out of scope for this document.

6.6. Required Signature Algorithm

For validation, SRP registrars MUST implement the ECDSAP256SHA256 signature algorithm. SRP registrars SHOULD implement the algorithms specified in [RFC8624], Section 3.1, in the validation column of the table, that are numbered 13 or higher and have a "MUST", "RECOMMENDED", or "MAY" designation in the validation column of the table. SRP requestors MUST NOT assume that any algorithm numbered lower than 13 is available for use in validating SIG(0) signatures.

7. Privacy Considerations

Because DNS-SD SRP Updates can be sent off-link, the privacy implications of SRP are different than for multicast DNS responses. Host implementations that are using TCP SHOULD also use TLS if available. SRP Registrar implementations MUST offer TLS support. The use of TLS with DNS is described in [RFC7858]. Because there is no mechanism for sharing keys, validation of DNS-over-TLS keys is not possible; DNS-over-TLS is used only as described in [RFC7858], Section 4.1

Hosts that implement TLS support SHOULD NOT fall back to TCP; since SRP registrars are required to support TLS, it is entirely up to the host implementation whether to use it.

Public keys can be used as identifiers to track hosts. SRP registrars MAY elect not to return KEY records for queries for SRP registrations. To avoid DNSSEC validation failures, an SRP registrar that signs the zone for DNSSEC but refuses to return a KEY record MUST NOT store the KEY record in the zone itself. Because the KEY record isn't in the zone, the nonexistance of the KEY record can be validated. If the zone is not signed, the server MAY instead return a negative non-error response (either NXDOMAIN or no data).

8. Domain Name Reservation Considerations

This section specifies considerations for systems involved in domain name resolution when resolving queries for names ending with '.service.arpa.'. Each item in this section addresses some aspect of the DNS or the process of resolving domain names that would be affected by this special-use allocation. Detailed explanations of these items can be found in Section 5 of [RFC6761].

8.1. Users

The current proposed use for 'service.arpa' does not require special knowledge on the part of the user. While the 'default.service.arpa.' subdomain is used as a generic name for registration, users are not expected to see this name in user interfaces. In the event that it does show up in a user interface, it is just a domain name, and requires no special treatment by the user. Users are not expected to see this name in user interfaces, although it's certainly possible that they might. If they do, they are not expected to treat it specially.

8.2. Application Software

Application software does not need to handle subdomains of 'service.arpa' specially. 'service.arpa' SHOULD NOT be treated as more trustworthy than any other insecure DNS domain, simply because it is locally-served (or for any other reason). It is not possible to register a PKI certificate for a subdomain of 'service.arpa.' because it is a locally-served domain name. So no such subdomain can be considered as uniquely identifying a particular host, as would be required for such a PKI cert to be issued. If a subdomain of 'service.arpa.' is returned by an API or entered in an input field of an application, PKI authentication of the endpoint being identified by the name will not be possible. Alternative methods and practices for authenticating such endpoints are out of scope for this document.

8.3. Name Resolution APIs and Libraries

Name resolution APIs and libraries MUST NOT recognize names that end in '.service.arpa.' as special and MUST NOT treat them as having special significance, except that it may be necessary that such APIs not bypass the locally configured recursive resolvers.

One or more IP addresses for recursive DNS servers will usually be supplied to the client through router advertisements or DHCP. For an administrative domain that uses subdomains of 'service.arpa.', the recursive resolvers provided by that domain will be able to answer queries for subdomains of 'service.arpa.'; other (non-local) resolvers will not, or they will provide answers that are not correct within that administrative domain.

A host that is configured to use a resolver other than one that has been provided by the local network may be unable to resolve, or may receive incorrect results for, subdomains of 'service.arpa.'. In order to avoid this, it is permissible that hosts use the resolvers that are locally provided for resolving 'service.arpa.', even when they are configured to use other resolvers.

8.4. Caching DNS Servers

There are three considerations for caching DNS servers that follow this specification:

1. For correctness, recursive resolvers at sites using 'service.arpa.' must in practice transparently support DNSSEC queries: queries for DNSSEC records and queries with the DNSSEC OK (DO) bit set (Section 3.2.1 of [RFC4035]). DNSSEC validation is a Best Current Practice [RFC9364]: although validation is not required, a caching recursive resolver that does not validate answers that can be validated may cache invalid data. This, in turn, would prevent validating stub resolvers from successfully validating answers. Hence, as a practical matter, recursive resolvers at sites using 'service.arpa' should do DNSSEC validation.

2. Unless configured otherwise, recursive resolvers and DNS proxies MUST behave as described in Locally Served Zones, Section 3 of [RFC6303]. That is, queries for 'service.arpa.' and subdomains of 'service.arpa.' MUST NOT be forwarded, with one important exception: a query for a DS record with the DO bit set MUST return the correct answer for that question, including correct information in the authority section that proves that the record is nonexistent.

So, for example, a query for the NS record for 'service.arpa.' MUST NOT result in that query being forwarded to an upstream cache nor to the authoritative DNS server for '.arpa.'. However, as necessary to provide accurate authority information, a query for the DS record MUST result in forwarding whatever queries are necessary; typically, this will just be a query for the DS record, since the necessary authority information will be included in the authority section of the response if the DO bit is set.

8.5. Authoritative DNS Servers

No special processing of 'service.arpa.' is required for authoritative DNS server implementations. It is possible that an authoritative DNS server might attempt to check the authoritative servers for 'service.arpa.' for a delegation beneath that name before answering authoritatively for such a delegated name. In such a case, because the name always has only local significance, there will be no such delegation in the 'service.arpa.' zone, and so the server would refuse to answer authoritatively for such a zone. A server that implements this sort of check MUST be configurable so that either it does not do this check for the 'service.arpa.' domain or it ignores the results of the check.

8.6. DNS Server Operators

DNS server operators MAY configure an authoritative server for 'service.arpa.' for use with SRP. The operator for the DNS servers authoritative for 'service.arpa.' in the global DNS will configure any such servers as described in Section 9.

8.7. DNS Registries/Registrars

'service.arpa.' is a subdomain of the 'arpa' top-level domain, which is operated by IANA under the authority of the Internet Architecture Board according to the rules established in [RFC3172]. There are no other DNS registrars for '.arpa'.

9. Delegation of 'service.arpa.'

In order to be fully functional, the owner of the 'arpa.' zone must add a delegation of 'service.arpa.' in the '.arpa.' zone [RFC3172]. This delegation is to be set up as was done for 'home.arpa', as a result of the specification in Section 7 of [RFC8375]. This is currently the responsibility of the IAB [IAB-ARPA]

10. IANA Considerations

10.1. Registration and Delegation of 'service.arpa' as a Special-Use Domain Name

IANA is requested to record the domain name 'service.arpa.' in the Special-Use Domain Names registry [SUDN]. IANA is requested, with the approval of IAB, to implement the delegation requested in Section 9.

IANA is further requested to add a new entry to the "Transport-Independent Locally-Served Zones" subregistry of the "Locally-Served DNS Zones" registry [LSDZ]. The entry will be for the domain 'service.arpa.' with the description "DNS-SD Service Registration Protocol Special-Use Domain", listing this document as the reference.

10.2. Subdomains of 'service.arpa.'

This document only makes use of the 'default.service.arpa' subdomain of 'service.arpa.' Other subdomains are reserved for future use by DNS-SD or related work. The IANA is requested to create a registry, the "service.arpa Subdomain" registry. The IETF shall have change control for this registry. New entries may be added either as a result of Standards Action Section 4.9 of [RFC8126] or with IESG approval Section 4.10 of [RFC8126], provided that a specification exists Section 4.6 of [RFC8126].

The IANA shall group the "service.arpa Subdomain" registry with the "Locally-Served DNS Zones" registry. The registry shall be a table with three columns: the subdomain name (expressed as a fully-qualified domain name), a brief description of how it is used, and a reference to the document that describes its use in detail.

This registry shall begin as the following table:

Subdomain Name	Description	reference
default.service.arpa.	Default domain for SRP updates	[THIS DOCUMENT]

Table 1

10.3. Service Name registrations

IANA is requested to add two new entries to the Service Names and Port Numbers registry. The following sections contain tables with the fields required by Section 8.1.1 of [RFC6335].

10.4. 'dnssd-srp' Service Name

Field Name	Value
Service Name	dnssd-srp
Transport Protocol	TCP
Assignee	IESG <iesg@ietf.org>
Contact	IETF Chair <chair@ietf.org>
Description	DNS-SD Service Registration
Reference	this document
Port Number	None
Service Code	None

Table 2

10.5. 'dnssd-srp-tls' Service Name

Field Name	Value
Service Name	dnssd-srp-tls
Transport Protocol	TCP
Assignee	IESG
Contact	IETF Chair
Description	DNS-SD Service Registration (TLS)
Reference	this document
Port Number	None
Service Code	None

Table 3

10.6. Anycast Address

IANA is requested to allocate an IPv6 Anycast address from the IPv6 Special-Purpose Address Registry, similar to the Port Control Protocol anycast address, 2001:1::1. The value TBD is to be replaced with the actual allocation in the table that follows. The purpose of this allocation is to provide a fixed anycast address that can be commonly used as a destination for SRP updates when no SRP registrar is explicitly configured. The values for the registry are:

Attribute	value
Address Block	2001:1::TBD/128
Name	DNS-SD Service Registration Protocol Anycast Address
RFC	[this document]
Allocation Date	[date of allocation]
Termination Date	N/A
Source	True
Destination	True
Forwardable	True
Global	True
Reserved-by-protocol	False

Table 4

11. Implementation Status

[Note to the RFC Editor: please remove this section prior to publication.]

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in RFC 7942. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation

here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to RFC 7942, "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

There are two known independent implementations of SRP requestors:

* SRP Client for OpenThread: https://github.com/openthread/openthread/pull/6038

* mDNSResponder open source project: https://github.com/Abhayakara/mdnsresponder

There are two related implementations of an SRP registrar. One acts as a DNS Update proxy, taking an SRP Update and applying it to the specified DNS zone using DNS update. The other acts as an Advertising Proxy [AP]. Both are included in the mDNSResponder open source project mentioned above.

12. Acknowledgments

Thanks to Toke Høiland-Jørgensen, Jonathan Hui, Esko Dijk, Kangping Dong and Abtin Keshavarzian for their thorough technical reviews. Thanks to Kangping and Abtin as well for testing the document by doing an independent implementation. Thanks to Tamara Kemper for doing a nice developmental edit, Tim Wattenberg for doing an SRP requestor proof-of-concept implementation at the Montreal Hackathon at IETF 102, and Tom Pusateri for reviewing during the hackathon and afterwards. Thanks to Esko for a really thorough second last call review. Thanks also to Nathan Dyck, Gabriel Montenegro, Kangping Dong, Martin Turon, and Michael Cowan for their detailed second last call reviews. Thanks to Patrik Fältström, Dhruv Dhody, David Dong, Joey Salazar, Jean-Michel Combes, and Joerg Ott for their respective directorate reviews. Thanks to Paul Wouters for a _really_ detailed IESG review! Thanks also to the other IESG members who provided comments or simply took the time to review the document.

13. Normative References

[I-D.ietf-dnssd-update-lease]
Cheshire, S. and T. Lemon, "An EDNS(0) option to negotiate Leases on DNS Updates", Work in Progress, Internet-Draft, draft-ietf-dnssd-update-lease-08, 7 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnssd-update-lease-08>.

[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>.

[RFC1536] Kumar, A., Postel, J., Neuman, C., Danzig, P., and S. Miller, "Common DNS Implementation Errors and Suggested Fixes", RFC 1536, DOI 10.17487/RFC1536, October 1993, <https://www.rfc-editor.org/info/rfc1536>.

[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>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, DOI 10.17487/RFC2136, April 1997, <https://www.rfc-editor.org/info/rfc2136>.

[RFC2181] Elz, R. and R. Bush, "Clarifications to the DNS Specification", RFC 2181, DOI 10.17487/RFC2181, July 1997, <https://www.rfc-editor.org/info/rfc2181>.

[RFC2539] Eastlake 3rd, D., "Storage of Diffie-Hellman Keys in the Domain Name System (DNS)", RFC 2539, DOI 10.17487/RFC2539, March 1999, <https://www.rfc-editor.org/info/rfc2539>.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, DOI 10.17487/RFC2931, September 2000, <https://www.rfc-editor.org/info/rfc2931>.

[RFC3172] Huston, G., Ed., "Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa")", BCP 52, RFC 3172, DOI 10.17487/RFC3172, September 2001, <https://www.rfc-editor.org/info/rfc3172>.

[RFC3445] Massey, D. and S. Rose, "Limiting the Scope of the KEY Resource Record (RR)", RFC 3445, DOI 10.17487/RFC3445, December 2002, <https://www.rfc-editor.org/info/rfc3445>.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", STD 88, RFC 3596, DOI 10.17487/RFC3596, October 2003, <https://www.rfc-editor.org/info/rfc3596>.

[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>.

[RFC6303] Andrews, M., "Locally Served DNS Zones", BCP 163, RFC 6303, DOI 10.17487/RFC6303, July 2011, <https://www.rfc-editor.org/info/rfc6303>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[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>.

[RFC8085] Eggert, L., Fairhurst, G., and G. Shepherd, "UDP Usage Guidelines", BCP 145, RFC 8085, DOI 10.17487/RFC8085, March 2017, <https://www.rfc-editor.org/info/rfc8085>.

[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>.

[RFC8375] Pfister, P. and T. Lemon, "Special-Use Domain 'home.arpa.'", RFC 8375, DOI 10.17487/RFC8375, May 2018, <https://www.rfc-editor.org/info/rfc8375>.

[RFC8624] Wouters, P. and O. Sury, "Algorithm Implementation Requirements and Usage Guidance for DNSSEC", RFC 8624, DOI 10.17487/RFC8624, June 2019, <https://www.rfc-editor.org/info/rfc8624>.

[RFC8765] Pusateri, T. and S. Cheshire, "DNS Push Notifications", RFC 8765, DOI 10.17487/RFC8765, June 2020, <https://www.rfc-editor.org/info/rfc8765>.

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

14. Informative References

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

[RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <https://www.rfc-editor.org/info/rfc2827>.

[RFC3007] Wellington, B., "Secure Domain Name System (DNS) Dynamic Update", RFC 3007, DOI 10.17487/RFC3007, November 2000, <https://www.rfc-editor.org/info/rfc3007>.

[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>.

[RFC6105] Levy-Abegnoli, E., Van de Velde, G., Popoviciu, C., and J. Mohacsi, "IPv6 Router Advertisement Guard", RFC 6105, DOI 10.17487/RFC6105, February 2011, <https://www.rfc-editor.org/info/rfc6105>.

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6760] Cheshire, S. and M. Krochmal, "Requirements for a Protocol to Replace the AppleTalk Name Binding Protocol (NBP)", RFC 6760, DOI 10.17487/RFC6760, February 2013, <https://www.rfc-editor.org/info/rfc6760>.

[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>.

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

[RFC7084] Singh, H., Beebee, W., Donley, C., and B. Stark, "Basic Requirements for IPv6 Customer Edge Routers", RFC 7084, DOI 10.17487/RFC7084, November 2013, <https://www.rfc-editor.org/info/rfc7084>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/info/rfc7228>.

[RFC7413] Cheng, Y., Chu, J., Radhakrishnan, S., and A. Jain, "TCP Fast Open", RFC 7413, DOI 10.17487/RFC7413, December 2014, <https://www.rfc-editor.org/info/rfc7413>.

[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>.

[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>.

[RFC8766] Cheshire, S., "Discovery Proxy for Multicast DNS-Based Service Discovery", RFC 8766, DOI 10.17487/RFC8766, June 2020, <https://www.rfc-editor.org/info/rfc8766>.

[RFC8945] Dupont, F., Morris, S., Vixie, P., Eastlake 3rd, D., Gudmundsson, O., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", STD 93, RFC 8945, DOI 10.17487/RFC8945, November 2020, <https://www.rfc-editor.org/info/rfc8945>.

[ROADMAP] Cheshire, S., "Service Discovery Road Map", Work in Progress, Internet-Draft, draft-cheshire-dnssd-roadmap-03, 23 October 2018, <https://datatracker.ietf.org/doc/html/draft-cheshire-dnssd-roadmap-03>.

[AP] Cheshire, S. and T. Lemon, "Advertising Proxy for DNS-SD Service Registration Protocol", Work in Progress, Internet-Draft, draft-ietf-dnssd-advertising-proxy-03, 28 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-dnssd-advertising-proxy-03>.

[I-D.ietf-snac-simple] Lemon, T. and J. Hui, "Automatically Connecting Stub Networks to Unmanaged Infrastructure", Work in Progress, Internet-Draft, draft-ietf-snac-simple-03, 30 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-snac-simple-03>.

[SUDN] "Special-Use Domain Names Registry", July 2012, <https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml>.

[LSDZ] "Locally-Served DNS Zones Registry", July 2011, <https://www.iana.org/assignments/locally-served-dns-zones/locally-served-dns-zones.xhtml>.

[IAB-ARPA] "Internet Architecture Board statement on the registration of special use names in the ARPA domain", March 2017, <https://www.iab.org/documents/correspondence-reports-documents/2017-2/iab-statement-on-the-registration-of-special-use-names-in-the-arpa-domain/>.

[ZC] Cheshire, S. and D.H. Steinberg, "Zero Configuration Networking: The Definitive Guide", O'Reilly Media, Inc. , ISBN 0-596-10100-7, December 2005.

Appendix A. Testing using standard RFC2136-compliant DNS servers

It may be useful to set up an authoritative DNS server for testing that does not implement SRP. This can be done by configuring the server to listen on the anycast address, or advertising it in the _dnssd-srp._tcp.<zone> SRV and _dnssd-srp-tls._tcp.<zone> record. It must be configured to be authoritative for "default.service.arpa", and to accept updates from hosts on local networks for names under "default.service.arpa" without authentication, since such servers will not have support for FCFS authentication (Section 3.2.4.1).

An authoritative DNS server configured in this way will be able to successfully accept and process SRP Updates from requestors that send SRP updates. However, no prerequisites will be applied, and this means that the test server will accept internally inconsistent SRP Updates, and will not stop two SRP Updates, sent by different services, that claim the same name(s), from overwriting each other.

Since SRP Updates are signed with keys, validation of the SIG(0) algorithm used by the requestor can be done by manually installing the requestor's public key on the DNS server that will be receiving the updates. The key can then be used to authenticate the SRP update, and can be used as a requirement for the update. An example configuration for testing SRP using BIND 9 is given in Appendix C.

Appendix B. How to allow SRP requestors to update standard RFC2136-compliant servers

Ordinarily SRP Updates will fail when sent to an RFC 2136-compliant server that does not implement SRP because the zone being updated is "default.service.arpa", and no DNS server that is not an SRP registrar would normally be configured to be authoritative for "default.service.arpa". Therefore, a requestor that sends an SRP Update can tell that the receiving server does not support SRP, but does support RFC2136, because the RCODE will either be NotZone, NotAuth or Refused, or because there is no response to the update request (when using the anycast address)

In this case a requestor MAY attempt to register itself using regular RFC2136 DNS updates. To do so, it must discover the default registration zone and the DNS server designated to receive updates for that zone, as described earlier, using the _dns-update._udp SRV record. It can then send the update to the port and host pointed to by the SRV record, and is expected to use appropriate prerequisites to avoid overwriting competing records. Such updates are out of scope for SRP, and a requestor that implements SRP MUST first attempt to use SRP to register itself, and only attempt to use RFC2136 backwards compatibility if that fails. Although the owner name for the SRV record specifies the UDP protocol for updates, it is also possible to use TCP, and TCP SHOULD be required to prevent spoofing.

Appendix C. Sample BIND9 configuration for default.service.arpa.

```
zone "default.service.arpa." {
  type primary;
  file "/etc/bind/primary/service.db";
  allow-update { key demo.default.service.arpa.; };
};
```

Figure 1: Zone Configuration in named.conf

```
$ORIGIN .
$TTL 57600  ; 16 hours
default.service.arpa IN SOA          ns3.default.service.arpa.
                                     postmaster.default.service.arpa. (
                2951053287 ; serial
                3600       ; refresh (1 hour)
                1800       ; retry (30 minutes)
                604800     ; expire (1 week)
                3600       ; minimum (1 hour)
)
                        NS           ns3.default.service.arpa.
                        SRV 0 0 53   ns3.default.service.arpa.
$ORIGIN default.service.arpa.
$TTL 3600   ; 1 hour
_ipps._tcp              PTR          demo._ipps._tcp
$ORIGIN _ipps._tcp.default.service.arpa.
demo                    TXT          "0"
                        SRV 0 0 9992 demo.default.service.arpa.
$ORIGIN _udp.default.service.arpa.
$TTL 3600   ; 1 hour
_dns-update             PTR          ns3.default.service.arpa.
$ORIGIN _tcp.default.service.arpa.
_dnssd-srp              PTR          ns3.default.service.arpa.
$ORIGIN default.service.arpa.
$TTL 300    ; 5 minutes
ns3                     AAAA         2001:db8:0:1::1
$TTL 3600   ; 1 hour
demo                    AAAA         2001:db8:0:2::1
                        KEY 0 3 13 (
                           qweEmaaq0FAWok5//ftuQtZgiZoiFSUsm0srWREdywQU
                           9dpvtOhrdKWUuPT3uEFF5TZU6B4q1z1I662GdaUwqg==
                        ); alg = ECDSAP256SHA256 ; key id = 15008
                        AAAA    ::1
```

Figure 2: Example Zone file

Authors' Addresses

Ted Lemon
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America
Email: mellon@fugue.com

Stuart Cheshire
Apple Inc.
One Apple Park Way
Cupertino, California 95014
United States of America
Phone: +1 408 974 3207
Email: cheshire@apple.com