Workgroup: Internet Engineering Task Force
Published: 14 September 2022
Intended Status: Standards Track
Expires: 18 March 2023
Authors: T. Lemon   S. Cheshire
         Apple Inc.   Apple Inc.

Service Registration Protocol for DNS-Based Service Discovery

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 802.11 (Wi-Fi) and 802.15.4 (IoT) networks. DNS-SD Service registration uses public keys and SIG(0) to allow services to defend their registrations against attack.

Status of This Memo

Copyright Notice

Table of Contents

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.

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 intended 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-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 SRP protocol 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.

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), described 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 knowledgable 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. 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.

2.1. Protocol Variants

2.1.1. 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 should send SRP updates. Hosts that support SRP Updates using TLS use the "_dnssd-srp-tls._tcp.<zone>" SRV record instead.

2.1.2. 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 (proposed) 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 SRP registrar addresses. It is the responsibility of the SRP 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 internally to names with broader visibility.

2.1.3. 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 Internet layer. Nodes connected to such networks can be assumed to be able to do DNSSD service registration domain discovery. Such networks are generally able to provide registration domain discovery and routing. This creates the possibility of off-network spoofing. To prevent such spoofing, TCP is required for such networks.

2.2. 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 4, we specify how to maintain the information once published.

2.2.1. 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 adds and removes. The types of records that are updated and removed in each of these instructions, as well as the constraints that apply to them, are described in Section 2.3. 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. In principle Service Description records can include other record types, with the same Service Instance Name, though in practice they rarely do. SRP does not support 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 first-come, first-serve naming, which we describe later on in this document.

2.2.2. 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 unicast namespace.

As described above, full-featured devices are responsible for knowing in what domain they should register their services. Devices made for Constrained-Node Networks register in the (proposed) special use domain name [RFC6761] "default.service.arpa", and let the SRP registrar handle rewriting that to a different domain if necessary.

2.2.3. 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 2.3 are implicit in the processing of SRP Updates, and so there is no need for the SRP requestor to put in any explicit prerequisites.

2.2.3.1. How DNS-SD Service Registration differs from standard RFC2136 DNS Update

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."

2.2.4. How to secure it

Traditional DNS update is secured using the TSIG protocol, [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 what we describe here is an improvement over the security of mDNS. The goal is not to provide the level of security of a network managed by a skilled operator.

2.2.4.1. 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 the key used to authenticate the DNS Update. 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]), its name is no longer protected. FCFS naming is used to protect both the Service Description and the Host Description.

2.2.5. SRP Requestor Behavior

2.2.5.1. Public/Private key pair generation and storage

The requestor generates a public/private key pair (See Section 5.3). 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 to erase the old key and install a new one. Therefore, the SRP requestor on the device SHOULD provide a mechanism to overwrite the key, for example as the result of a "factory reset."

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 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.

2.2.5.2. 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 registrar to have a conflict with an existing name, the registrar will respond to the SRP Update with a YXDomain rcode. In this case, the requestor MUST either abandon the service registration attempt, or else choose a new name.

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.

2.2.5.3. 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.

2.2.5.4. Compression in SRV records

Although [RFC2782] requires that the target name in the SRV record not be compressed, an SRP requestor SHOULD 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 potentially saves substantial space in the SRP Update. SRP registrars MUST support compression of SRV RR targets.

2.2.5.5. Removing published services

2.2.5.5.1. Removing all published services

To remove all the services registered to a particular host, the SRP requestor retransmits its most recent update 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 have 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 a 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 2.3.1.3).

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.

2.2.5.5.2. 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 2.3.1.1) 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 2.3.1.2) 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 2.2.5.5.1, which deletes the host and all services that have that host as their target.

2.3. Validation and Processing of SRP Updates

2.3.1. Validation of Adds and Deletes

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.

2.3.1.1. 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 remove, the above constraints must be met.

2.3.1.2. Service Description Instruction

An instruction is a Service Description Instruction if, for the appropriate Service Instance Name, it contains

* exactly one "Delete all RRsets from a name" update for the service instance name ([RFC2136], Section 2.5.3),
* zero or one "Add to an RRset" SRV RR,
* 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),

* 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.
* Service Descriptions Instructions do not modify any other resource records.

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

2.3.1.3. 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 of sufficient scope to be validly published in a DNS zone can be ignored by the SRP server, 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 SRP registrar may 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.

2.3.2. 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 registrar MAY either process such messages are either processed as regular RFC2136 updates, including access control checks and constraint checks, if supported, or MAY reject them with 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.

2.3.3. FCFS Name And Signature Validation

Assuming that a DNS Update message has been validated with these conditions and is a valid SRP Update, the 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 registrar MUST reject the SRP Update with the YXDomain RCODE.

Otherwise, the 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.

2.3.4. 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.

2.3.5. SRP Update response

The status that is returned depends on the result of processing the update, and can be either NoError or ServFail: all other possible outcomes should already have been accounted for when applying the constraints that qualify the update as an SRP Update.

2.3.6. Optional Behavior

The 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 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 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 Refused.

3. 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 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 never be longer than the lease time (Section 4.1). 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.

4. Maintenance

4.1. 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 should be assumed to be valid. The Key Lease time represents the time after the update during which the KEY record should be assumed to be valid.

The reasoning behind the different lease times is discussed in the section on first-come, first-served naming (Section 2.2.4.1). 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 SRP 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 time-it 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. 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. Dual-use servers 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'.

5. Security Considerations

5.1. Source Validation

SRP Updates have no authorization semantics other than first-come, first-served. This means that if an attacker from outside of the administrative domain of the registrar knows the registrar's IP address, it can in principle send updates to the registrar that will be processed successfully. 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 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]

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 to update records added by SRP requestors.

5.2. 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.

5.3. 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.

6. Privacy Considerations

Because DNSSD 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. Registrar implementations MUST offer TLS support. The use of TLS with DNS is described in [RFC7858].

Hosts that implement TLS support SHOULD NOT fall back to TCP; since 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.

7. 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 should 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]

8. IANA Considerations

8.1. 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 7.

IANA is further requested to add a new entry to the "Transport-Independent Locally-Served Zones" subregistry of the 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.

8.2. 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". This registry shall begin as the following table:

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

Table 1

8.3. 'dnssd-srp' Service Name

IANA is also requested to add a new entry to the Service Names and Port Numbers registry for dnssd-srp with a transport type of tcp. No port number is to be assigned. The reference should be to this document, and the Assignee and Contact information should reference the authors of this document. The Description should be "DNS-SD Service Registration."

8.4. 'dnssd-srp-tls' Service Name

IANA is also requested to add a new entry to the Service Names and Port Numbers registry for dnssd-srp-tls with a transport type of tcp. No port number is to be assigned. The reference should be to this document, and the Assignee and Contact information should reference the authors of this document. The Description should be "DNS-SD Service Registration (TLS).

8.5. 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 should be replaced with the actual allocation in the table that follows. 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 2

9. 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.

10. 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 a 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.

11. Normative References

[I-D.ietf-dnssd-update-lease] Cheshire, S. and T. Lemon, "An EDNS0 option to negotiate Leases on DNS Updates", Work in Progress, Internet-Draft, draft-ietf-dnssd-update-lease-02, 11 July 2022, <https://www.ietf.org/archive/id/draft-ietf-dnssd-update-lease-02.txt>.

[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>.

[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>.

[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>.

[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>.

[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>.

[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/>.

12. 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>.

[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>.

[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>.

[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://www.ietf.org/archive/id/draft-cheshire-dnssd-roadmap-03.txt>.

[AP] Cheshire, S. and T. Lemon, "Advertising Proxy for DNS-SD Service Registration Protocol", Work in Progress, Internet-Draft, draft-ietf-dnssd-advertising-proxy-01, 11

July 2022, <https://www.ietf.org/archive/id/draft-ietf-dnssd-advertising-proxy-01.txt>.

[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 2.2.4.1).

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 should 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 should 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 should 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 master;
  file "/etc/bind/master/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 513 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