DNS SIG(0)

DNS Public Key Based Authentication/Authorization

draft-eastlake-dnsop-rfc2931bis-sigzero-03

Donald Eastlake 3rd d3e3e3@gmail.com

Johan Stenstam
johan.stenstam@internetstiftelsen.se

SIG(0) & DNS-SD SRP

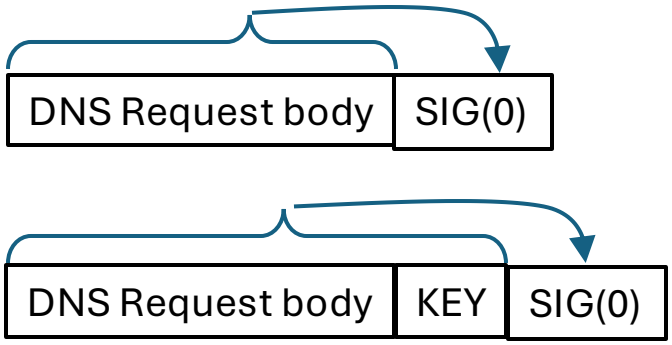
- The Service Registration Protocol (SRP [RFC9665]) uses a variant of the DNS Update request [RFC2136].
- DNS-SD SRP supports First Come, First Served (FCFS) Naming allowing the requester to secure a name and for subsequent requests related to that name to be authenticated as coming from the same requester.
- SRP FCFS Naming uses an initial DNS Update request that contains a KEY RR with the public key to be associated with the name. That message is signed by a SIG(0) using the corresponding private key known only to the update requester.

SIG(0)

- SIG(0) [RFC2931] is a meta DNS Resource Record (RR) providing DNS request and transaction security based on a public/private key signature. It has the structure of the RRSIG RR with a “Type covered” field of zero.
  - Being a meta (pseudo) RR means that it can be included in DNS messages, but it is not included in stored zone data.
- The corresponding public key is in a KEY RR [RFC2535] which has the structure of the DNSKEY RR and can be stored in zone data or included, as is done in DNS-SD Service Registration Protocol (SRP [RFC9665]), in DNS messages.

SIG(0) uses

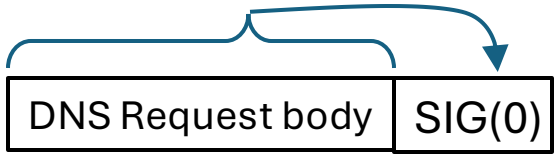
- Request message integrity & origin/authority authentication
  - Ordinary Request
    - Request signed by a private key at the requester
  - SRP FCFS Naming Request
    - Request signed by a private key at the server corresponding to KEY



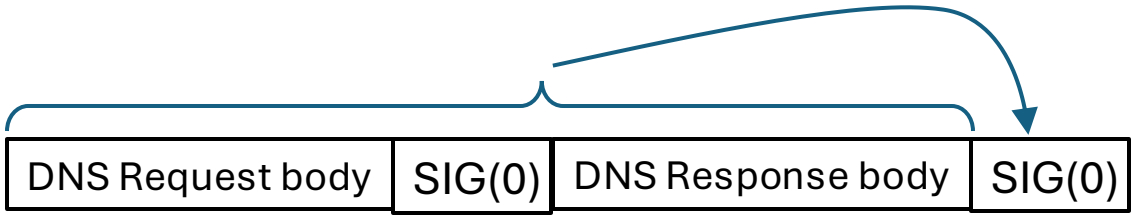
- DNS UPDATE [RFC2136] Request Authorization
  - SIG(0) signed with a key indicating update authority for the zone.
- SRP FCFS Naming Request Authorization
  - SIG(0) signed with a key that will indicate continuity of the same requester [RFC9665].
- DNS TKEY Message Authorization
  - SIG(0) signed with a key indication authority for a host.

SIG(0) uses

- Transaction integrity & authentication
  - Request



  - Transaction
    - Request and response together signed by a private key at the server



TSIG

- TSIG (Transaction Signature, RFC 8945) is a meta RR providing efficient DNS request and transaction authentication based on a keyed hash algorithm and shared secret key. (transaction = concatenated request and response)
- TSIG requires a shared secret key and is not used in SRP. (However, it is possible by using TKEY [RFC2930] RRs and SIG(0) to dynamically configure secure shared secret keys.)

RFC 2931 SIG(0) RR General Problems

- Has no separate extended Error field, unlike TSIG and TKEY. So, an error separate from the overall DNS request error cannot be signaled.
- Has no Original ID field so DNS forwarded authentication not obviously supported. This is not a problem for DNS SD because it does not use general DNS servers that recursively forward.
  - BIND supports multi-hop SIG(0) by forwarding with TCP with the same ID and maintaining a separate ID space per TCP connection.

RFC 2931 SIG(0) RR Problems (continued)

- Not extensible like TSIG and TKEY.
- More thane one SIG(0) in a message is prohibited.
- Has no way to indicate the state of the keys being used. See draft-berra-dnsop-keystate:
  - Examples in requests:
    - Manual bootstrap request
    - Key inquiry
    - Policy inquiry
  - Example in responses
    - Key is known and trusted
    - Key is unknown
    - Key is invalid
    - Key is refused
    - Known Key but didn’t validate
    - Etc.

RFC 2931 SIG(0) RR Solutions

1. Create a new RR with additional fields somewhat like TSIG.
2. Overload fields in existing SIG(0).
   - Error Code(16 bits) and Original ID (16 bits) → Original TTL (32 bits)
   - Key State (8 bits) → CLASS (16 bits) or Labels (8 bits)
3. Extend with EDNS Options that link to the SIG(0) using Key Tag or the like.

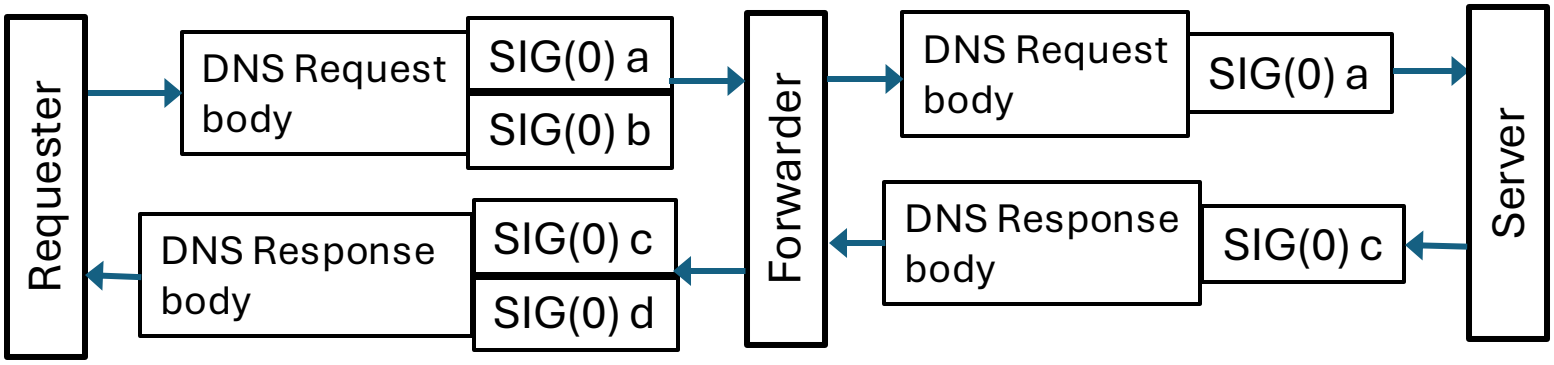
RFC 2931 SIG(0) RR Solutions Pro/Con

1. New RR: Clean solution but no backwards compatibility and perhaps some deployment difficulty.
2. Overload fields in existing SIG(0): Easiest. Possible backwards compatibility depending on browser code and specific field re-use. Does not get extensibility.
3. Extend with EDNS Option(s): Possible problems linking OPT to SIG(0) if multiple SIG(0)s allowed.

SIG(0) Changes In Drafts

- draft-eastlake-dnsop-rfc2931bis-sigzero-03
  - Removed statement that TCP support of SIG(0) is optional.
  - Added an EDNS(0) option to carry the Original ID and an error value.
  - Added section on considerations for forwarding DNS servers.
  - Permits multiple SIG(0)s in a DNS message. (But not a SIG(0) and a TSIG and not multiple TSIGs.)
- draft-berra-dnsop-keystate-01 (expired)
  - Add an EDNS(0) option that can be used to synchronize SIG(0) key state.

Example of Multiple and Forwarded SIG(0)s



a) Authority and integrity for action at Server.

b) Integrity and possibly authorization at Forwarder.

c) Transaction integrity from Server.

d) Transaction integrity from Forwarder.

Questions?

END

Donald Eastlake 3rd d3e3e3@gmail.com

Johan Stenstam
johan.stenstam@internetstiftelsen.se

Historical RR Type Note

- Both DNS transaction security and DNSSEC data security originally used the
  - SIG (type = 24) and
  - KEY (type = 25) RRs [RFC2535].
- DNSSEC was changed to use the
  - RRSIG (type = 46) and
  - DNSKEY (type = 48) RRs [RFC4034].
- The corresponding RRs have the same field structure as each other. Transaction security continues to use the SIG and KEY RRs.