Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

D. Wessels
P. Barber
M. Weinberg
Verisign
W. Kumari
Google
W. Hardaker
USC/ISI

July 2, 2018

Message Digest for DNS Zones
draft-wessels-dns-zone-digest-02

Abstract

This document describes a protocol and DNS Resource Record used to provide a message digest over DNS zone data. In particular, it describes how to compute, sign, represent, and use the message digest to verify the contents of a zone for accuracy and completeness. The ZONEMD Resource Record type is introduced for conveying the message digest data.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

In the DNS, a zone is the collection of authoritative resource records (RRs) sharing a common origin ([RFC7719]), which can be distributed from primary to secondary name servers. Zones are often stored as files on disk in the so-called master file format [RFC1034]. Sometimes zones are distributed outside of the DNS, with such protocols as FTP, HTTP, rsync, and so on. Currently there is no standard way to verify the authenticity of a stand-alone zone file.

This document introduces a new RR type that serves as a cryptographic message digest of the data in a zone file. It allows a receiver of the zone file to verify the zone file's authenticity, especially when used in combination with DNSSEC. This technique makes the message digest a part of the zone file itself, allowing anything to verify the zone file as a whole, no matter how it is transmitted.

DNSSEC provides three strong security guarantees relevant to this protocol:

1. whether or not to expect DNSSEC records in the zone,

2. whether or not to expect a ZONEMD record in a signed zone, and

3. whether or not the ZONEMD record has been altered since it was signed.

This specification is OPTIONAL to implement by both publishers and consumers of zone file data.

1.1. Motivation

The motivation and design of this protocol enhancement is tied to the DNS root zone [InterNIC]. The root zone is perhaps the most widely distributed DNS zone on the Internet, served by 930 separate instances [RootServers] at the time of this writing. Additionally,

many organizations configure their own name servers to serve the root zone locally. Reasons for doing so include privacy and reduced access time. [RFC7706] describes one, but not the only, way to do this. As the root zone spreads beyond its traditional deployment boundaries, the need for verification of the completeness of the zone contents becomes increasingly important.

One approach to preventing data tampering and corruption is to secure the distribution channel. The DNS has a number of features that can already be used for channel security. Perhaps the most widely used is DNS transaction signatures (TSIG [RFC2845]). TSIG uses shared secret keys and a message digest to protect individual query and

response messages. It is generally used to authenticate and validate UPDATE [RFC2136], AXFR [RFC5936], and IXFR [RFC1995] messages.

DNS Request and Transaction Signatures (SIG(0) [RFC2931]) is another protocol extension designed to authenticate individual DNS transactions. Whereas SIG records were originally designed to cover specific RR types, SIG(0) is used to sign an entire DNS message. Unlike TSIG, SIG(0) uses public key cryptography rather than shared secrets.

The Transport Layer Security protocol suite is also designed to provide channel security. It is entirely possible, for example, to perform zone transfers using DNS-over-TLS ([RFC7858]). Furthermore, one can easily imagine the distribution of zone files over HTTPS-enabled web servers, as well as DNS-over-HTTPS [dns-over-https].

Unfortunately, the protections provided by these channel security techniques are ephemeral and are not retained after the data transfer is complete. They can ensure that the client receives the data from the expected server, and that the data sent by the server is not modified during transmission. However, they do not guarantee that the server transmits the data as originally published, and do not provide any methods to verify data that is read after transmission is complete. For example, a name server loading saved zone data upon restart cannot guarantee that the on-disk data has not been modified. For these reasons, it is preferable to secure the data itself.

DNSSEC provides certain data security guarantees. For zones that are signed, a recipient can validate all of the signed RRsets.

Additionally, the denial-of-existence records can prove that RRsets have not been added or removed. However, not all RRsets in a zone are signed. The design of DNSSEC stipulates that delegations (non-apex NS records) are not signed, and neither are any glue records. Thus, changes to delegation and glue records cannot be detected by DNSSEC alone. Furthermore, zones that employ NSEC3 with opt-out are susceptible to the removal or addition of names between the signed nodes.

There are existing tools and protocols that provide data security, such as OpenPGP [RFC4880] and S/MIME [RFC3851]. In fact, the internic.net site publishes PGP signatures along side the root zone and other files available there. However, this is a detached signature with no strong association to the corresponding zone file other than its timestamp. Non-detached signatures are, of course, possible, but these necessarily change the format of the file being distributed. That is, a zone file signed with OpenPGP or S/MIME no longer looks like a zone file and could not directly be loaded into a

name server. Once loaded the signature data is lost, so it does not survive further propagation.

It seems the desire for data security in DNS zones was envisioned as far back as 1997. [RFC2065] is an obsoleted specification of the first generation DNSSEC Security Extensions. It describes a zone transfer signature, aka AXFR SIG, which is similar to the technique proposed by this document. That is, it proposes ordering all RRsets in a zone, hashing their signatures, and then signing the zone hash. The AXFR SIG is described only for use during zone transfers. It did not postulate the need to validate zone data distributed outside of the DNS. Furthermore, its successor, [RFC2535], omits the AXFR SIG, while at the same time introducing an IXFR SIG.

1.2. Design Overview

This document introduces a new Resource Record type designed to convey a message digest of the content of a zone file. The digest is calculated at the time of zone publication. Ideally the zone is signed with DNSSEC to guarantee that any modifications of the digest can be detected.

The zone digest is designed to be used on zones that are relatively stable and have infrequent updates. As currently specified, the digest is re-calculated over the entire zone content each time. This specification does not provide an efficient mechanism for incremental updates of zone data. The authors believe that the incremental updates represent significant complexity which could be a barrier to implementation at this time (see DNS Camel). Nothing in this specification prevents future work to support incremental zone digest algorithms (e.g. using Merkle trees and a different RR type).

The cryptographic algorithms available for zone digest are exactly the same as for DS records. This avoids the need for a separate digest algorithm registry. Any updates to the DS algorithms automatically updates the algorithm status for zone digests.

It is expected that verification of a zone digest will be implemented in name server software. That is, a name server can verify the zone data it was given and refuse to serve a zone which fails verification. For signed zones, the name server would need a trust anchor for DNSSEC validation. For signed non-root zones, the name server may need to send queries to validate a chain-of-trust. Digest verification may also be performed externally.

1.3. Requirements Language

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

2. The ZONEMD Resource Record

This section describes the ZONEMD Resource Record, including its fields, wire format, and presentation format. The Type value for the ZONEMD RR is TBD. The ZONEMD RR is class independent. The RDATA of the resource record consists of three fields: Serial, Digest Type, and Digest.

FOR DISCUSSION: This document is currently written as though a zone MUST NOT contain more than one ZONEMD RR. Having exactly one ZONEMD record per zone simplifies this protocol and eliminates confusion around downgrade attacks, at the expense of algorithm agility.

2.1. ZONEMD RDATA Wire Format

The ZONEMD RDATA wire format is encoded as follows:

```
                        1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Serial                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Digest Type  |                                               |
   +-+-+-+-+-+-+-+-+             Digest                            +
   /                                                               /
   /                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

2.1.1. The Serial Field

The Serial field is a 32-bit unsigned integer in network order. It is equal to the serial number from the zone's SOA record ([RFC1035] section 3.3.13) for which the message digest was generated.

2.1.2. The Digest Type Field

The Digest Type field is an 8-bit unsigned integer, with meaning equivalent to the Digest Type of the DS resource record, as defined in section 5.1.3 of [RFC4034].

The status of ZONEMD digest types (e.g., mandatory, optional, deprecated) SHALL always match the status for DS records. This information can be found in the IANA protocol registry for DS digest types [iana-ds-digest-types].

At the time of this writing the following digest types are defined:

```
            +-------+-----------------+-----------+-----------+
```

Value	Description	Status	Reference
1	SHA1	Mandatory	[RFC3658]
2	SHA256	Mandatory	[RFC4509]
3	GOST R 34.11-94	Optional	[RFC5933]
4	SHA384	Optional	[RFC6605]

Table 1: Digest Types

2.1.3. The Digest Field

The Digest field is a variable-length sequence of octets containing the message digest. Section 3 describes how to calculate the digest for a zone. Section 4 describes how to use the digest to verify the contents of a zone.

2.2. ZONEMD Presentation Format

The presentation format of the RDATA portion is as follows:

The Serial field MUST be represented as an unsigned decimal integer.

The Digest Type field MUST be represented as an unsigned decimal integer.

The Digest MUST be represented as a sequence of case-insensitive hexadecimal digits. Whitespace is allowed within the hexadecimal text.

2.3. ZONEMD Example

The following example shows a ZONEMD RR.

```
example.com. 86400 IN ZONEMD ( 2018031500 4 FEBE3D4CE2EC2FFA4BA9
                                            9D46CD69D6D29711E552
                                            17057BEE7EB1A7B641A4
                                            7BA7FED2DD5B97AE499F
                                            AFA4F22C6BD647DE )
```


3. Calculating the Digest

3.1. Canonical Format and Ordering

Calculation of the zone digest REQUIRES the RRs in a zone to be in a consistent format and ordering. Correct ordering of the zone depends on (1) ordering of owner names in the zone, (2) ordering of RRsets with the same owner name, and (3) ordering of RRs within an RRset.

This specification adopts DNSSEC's canonical ordering for names (Section 6.1 of [RFC4034]), and canonical ordering for RRs within an RRset (Section 6.3 of [RFC4034]). It also adopts DNSSEC's canonical RR form (Section 6.2 of [RFC4034]). However, since DNSSEC does not define a canonical ordering for RRsets having the same owner name, that ordering is defined here.

3.1.1. Order of RRsets Having the Same Owner Name

For the purposes of calculating the zone digest, RRsets having the same owner name MUST be numerically ordered by their numeric RR TYPE.

3.1.2. Special Considerations for SOA RRs

When AXFR is used to transfer zone data, the first and last records are always the SOA RR ([RFC5936] Section 2.2). Because of this, zone files on disk often contain two SOA RRs. When calculating the zone digest, the first SOA RR MUST be included and any subsequent SOA RRs MUST NOT be included.

Additionally, per established practices, the SOA record is generally the first record in a zone file. However, according to the requirement to sort RRsets with the same owner name by type, the SOA RR (type value 6) will not be first in the digest calculation. The zone's NS RRset (type value 2) at the apex MUST be processed before the SOA RR.

3.2. Add ZONEMD Placeholder

In preparation for calculating the zone digest, any existing ZONEMD records MUST first be deleted from the zone.

Prior to calculation of the digest, and prior to signing with DNSSEC, a placeholder ZONEMD record MUST be added to the zone. This serves two purposes: (1) it allows the digest to cover the Serial and Digest Type field values, and (2) ensures that appropriate denial-of-existence (NSEC, NSEC3) records are created if the zone is signed with DNSSEC.

In the placeholder record, the Serial field MUST be set to the current SOA Serial. The Digest Type field MUST be set to the value for the chosen digest algorithm. The Digest field MUST be set to all zeroes and of length appropriate for the chosen digest algorithm.

3.3. Optionally Sign the Zone

Following addition of the placeholder record, the zone MAY be signed with DNSSEC. Note that when the digest calculation is complete, and the ZONEMD record is updated, the signature(s) for that record MUST be recalculated and updated as well. Therefore, the signer is not required to calculate a signature over the placeholder record at this step in the process, but it is harmless to do so.

3.4. Calculate the Digest

The zone digest is calculated by concatenating the canonical on-the-wire form of all RRs in the zone, in the order described above, subject to the inclusion/exclusion rules described below, and then applying the digest algorithm:

```
digest = digest_algorithm( RR(1) | RR(2) | RR(3) | ... )
```

where "|" denotes concatenation, and

```
RR(i) = owner | type | class | TTL | RDATA length | RDATA
```

3.4.1. Inclusion/Exclusion Rules

When calculating the digest, the following inclusion/exclusion rules apply:

- o All records in the zone including glue records MUST be included.
- o More than one SOA MUST NOT be included.
- o The placeholder ZONEMD RR MUST be included.
- o If the zone is signed, DNSSEC RRs MUST be included, except:
- o The RRSIG covering ZONEMD MUST NOT be included.

FOR DISCUSSION: Ambiguities about records that are in/out of zone. For example, see Jinmei message to dnsop 2018-06-01 and followups. BIND will load and axfr data "occluded" by DNAME/NS.

3.5. Update ZONEMD RR

Once the zone digest has been calculated, its value is then copied to the Digest field of the ZONEMD record.

If the zone is signed with DNSSEC, the appropriate RRSIG records covering the ZONEMD record MUST then be added. Because the ZONEMD placeholder was added prior to signing, the zone will already have the appropriate denial-of-existence (NSEC, NSEC3) records.

Some implementations of incremental DNSSEC signing might update the zone's serial number for each resigning. However, to preserve the calculated digest, generation of the ZONEMD signature at this time MUST NOT also result in a change of the SOA serial number.

4. Verifying Zone Message Digest

The recipient of a zone that has a message digest record can verify the zone by calculating the digest as follows:

1. The verifier SHOULD first determine whether or not to expect DNSSEC records in the zone. This can be done by examining locally configured trust anchors, or querying for (and validating) DS RRs in the parent zone. For zones that are provably unsigned, digest validation continues at step 4 below.

2. For zones that are provably signed, the existence of the ZONEMD record MUST be verified. If the ZONEMD record provably does not exist, digest verification cannot be done. If the ZONEMD record does provably exist, but is not found in the zone, digest verification MUST NOT be considered successful.

3. For zones that are provably signed, the SOA RR and ZONEMD RR(set) MUST have valid signatures, chaining up to a trust anchor. If DNSSEC validation of the SOA or ZONEMD records fails, digest verification MUST NOT be considered successful.

4. The SOA Serial field MUST exactly match the ZONEMD Serial field. If the fields to not match, digest verification MUST NOT be

considered successful.

5. The ZONEMD Digest Type field MUST be checked. If the verifier does not support the given digest type, it SHOULD report that the zone digest could not be verified due to an unsupported algorithm.

6. The zone digest is calculated using the algorithm described in Section 3.4. Note in particular that digested ZONEMD RRs MUST be placeholders and their RRSIGs are not included in the digest.

7. The calculated digest is compared to the received digest. If the two digest values match, verification is considered successful. Otherwise, verification MUST NOT be considered successful.

8. If the zone is to be served and transferred, the original (not placeholder) ZONEMD RRs MUST be sent to recipients so that downstream clients can verify the zone.

5. IANA Considerations

5.1. ZONEMD RRtype

This document uses a new DNS RR type, ZONEMD, whose value TBD has been allocated by IANA from the "Resource Record (RR) TYPEs" subregistry of the "Domain Name System (DNS) Parameters" registry.

5.2. ZONEMD Digest Type

The ZONEMD Digest Type field has the same semantics as the DS RR Digest Type field. Thus, it does not add new IANA protocol registry requirements.

6. Security Considerations

6.1. Attacks Against the Zone Digest

The zone digest allows the receiver to verify that the zone contents haven't been modified since the zone was generated/published.

Verification is strongest when the zone is also signed with DNSSEC. An attacker, whose goal is to modify zone content before it is used by the victim, may consider a number of different approaches.

The attacker might perform a downgrade attack to an unsigned zone. This is why Section 4 RECOMMENDS that the verifier determine whether or not to expect DNSSEC signatures for the zone in step 1.

The attacker might perform a downgrade attack by removing the ZONEMD record. This is why Section 4 REQUIRES that the verifier checks DNSSEC denial-of-existence proofs in step 2.

The attacker might alter the Digest Type or Digest fields of the ZONEMD record. Such modifications are detectable only with DNSSEC validation.

6.2. Attacks Utilizing the Zone Digest

Nothing in this specification prevents clients from making, and servers from responding to, ZONEMD queries. One might consider how well ZONEMD responses could be used in a distributed denial-of-service amplification attack.

The ZONEMD RR is moderately sized, much like the DS RR. A single ZONEMD RR contributes approximately 40 to 65 octets to a DNS response, for currently defined digest types. Certainly other query types result in larger amplification effects (i.e., DNSKEY).

FOR DISCUSSION: The primary purpose of the ZONEMD record is to verify a zone file prior to being loaded or served by a name server. We could allow a name server implementation to respond to ZONEMD queries with the REFUSED RCODE without loss of functionality. Note that refusal would prevent ensuring that a zone-walk is complete.

7. Privacy Considerations

This specification has no impacts on user privacy.

8. Acknowledgments

The authors wish to thank David Blacka, Scott Hollenbeck, and Rick

Wilhelm for providing feedback on early drafts of this document. Additionally, they thank Mark Andrews, Olafur Gudmundsson, Shumon Huque, Tatuya Jinmei, Shane Kerr, Mukund Sivaraman, Petr Spacek, and other members of the dnsop working group for their input.

9. Implementation Status

9.1. Authors' Implementation

The authors are currently working on an implementation in C, using the ldns library [ldns]. This implementation is able to perform the following functions:

- o Read input zone file, output zone file with ZONEMD placeholder.
- o Compute zone digest over signed zone file and update ZONEMD record.
- o Re-compute DNSSEC signature over ZONEMD record.
- o Verify zone digest from input zone file.

The authors expect to be able to release this implementation as open source following submission of this Internet-Draft.

10. Change Log

RFC Editor: Please remove this section.

This section lists substantial changes to the document as it is being worked on.

From -00 to -01:

- o Removed requirement to sort by RR CLASS.
- o Added Kumari and Hardaker as coauthors.
- o Added Change Log section.

o Minor clarifications and grammatical edits.

11. References

11.1. Normative References

[iana-ds-digest-types]
IANA, "Delegation Signer (DS) Resource Record (RR) Type Digest Algorithms", April 2012, <https://www.iana.org/assignments/ds-rr-types/ds-rr-types.xhtml>.

[ldns] NLNet Labs, "The ldns Library", March 2018, <https://www.nlnetlabs.nl/projects/ldns/>.

[RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <https://www.rfc-editor.org/info/rfc1034>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3658] Gudmundsson, O., "Delegation Signer (DS) Resource Record (RR)", RFC 3658, DOI 10.17487/RFC3658, December 2003, <https://www.rfc-editor.org/info/rfc3658>.

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

[RFC4509] Hardaker, W., "Use of SHA-256 in DNSSEC Delegation Signer (DS) Resource Records (RRs)", RFC 4509,

DOI 10.17487/RFC4509, May 2006, <https://www.rfc-editor.org/info/rfc4509>.

[RFC5933] Dolmatov, V., Ed., Chuprina, A., and I. Ustinov, "Use of GOST Signature Algorithms in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5933, DOI 10.17487/RFC5933, July 2010, <https://www.rfc-editor.org/info/rfc5933>.

[RFC6605] Hoffman, P. and W. Wijngaards, "Elliptic Curve Digital Signature Algorithm (DSA) for DNSSEC", RFC 6605, DOI 10.17487/RFC6605, April 2012, <https://www.rfc-editor.org/info/rfc6605>.

[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>.

11.2. Informative References

[dns-over-https] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", draft-ietf-doh-dns-over-https-12 (work in progress), June 2018, <https://tools.ietf.org/html/draft-ietf-doh-dns-over-https-12>.

[InterNIC] ICANN, "InterNIC FTP site", May 2018, <ftp://ftp.internic.net/domain/>.

[RFC1995] Ohta, M., "Incremental Zone Transfer in DNS", RFC 1995, DOI 10.17487/RFC1995, August 1996, <https://www.rfc-editor.org/info/rfc1995>.

[RFC2065] Eastlake 3rd, D. and C. Kaufman, "Domain Name System Security Extensions", RFC 2065, DOI 10.17487/RFC2065, January 1997, <https://www.rfc-editor.org/info/rfc2065>.

[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>.

[RFC2535] Eastlake 3rd, D., "Domain Name System Security Extensions", RFC 2535, DOI 10.17487/RFC2535, March 1999, <https://www.rfc-editor.org/info/rfc2535>.

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, DOI 10.17487/RFC2845, May 2000, <https://www.rfc-editor.org/info/rfc2845>.

[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>.

[RFC3851] Ramsdell, B., Ed., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", RFC 3851, DOI 10.17487/RFC3851, July 2004, <https://www.rfc-editor.org/info/rfc3851>.

[RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", RFC 4880, DOI 10.17487/RFC4880, November 2007, <https://www.rfc-editor.org/info/rfc4880>.

[RFC5936] Lewis, E. and A. Hoenes, Ed., "DNS Zone Transfer Protocol (AXFR)", RFC 5936, DOI 10.17487/RFC5936, June 2010, <https://www.rfc-editor.org/info/rfc5936>.

[RFC7706] Kumari, W. and P. Hoffman, "Decreasing Access Time to Root Servers by Running One on Loopback", RFC 7706, DOI 10.17487/RFC7706, November 2015, <https://www.rfc-editor.org/info/rfc7706>.

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

[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>.

[RootServers]
Root Server Operators, "Root Server Technical Operations", July 2018, <https://www.root-servers.org/>.

Authors' Addresses

Duane Wessels
Verisign
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200
Email: dwessels@verisign.com
URI: http://verisign.com

Piet Barber
Verisign
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200
Email: pbarber@verisign.com
URI: http://verisign.com

Matt Weinberg
Verisign
12061 Bluemont Way
Reston, VA 20190

Phone: +1 703 948-3200
Email: mweinberg@verisign.com
URI: http://verisign.com

Warren Kumari
Google
1600 Amphitheatre Parkway
Mountain View, CA 94043

Email: warren@kumari.net

Wes Hardaker
USC/ISI
P.O. Box 382
Davis, CA 95617

Email: ietf@hardakers.net