DNSOP Working Group
Internet-Draft
Intended status: Informational
Expires: 3 April 2026

A.M. Fregly
J. Harvey
B. Kaliski
D. Wessels
Verisign Labs

30 September 2025

Stateless Hash-Based Signatures in Merkle Tree Ladder Mode (SLH-DSA-MTL) for DNSSEC

draft-fregly-dnsop-slh-dsa-mtl-dnssec-05

Abstract

This document describes how to apply the Stateless Hash-Based Digital Signature Algorithm in Merkle Tree Ladder mode to the DNS Security Extensions. This combination is referred to as the SLH-DSA-MTL Signature scheme. This document describes how to specify SLH-DSA-MTL keys and signatures in DNSSEC. It uses both the SHA2 and SHAKE family of hash functions. This document also provides guidance for use of EDNS(0) in signature retrieval.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 April 2026.

Copyright Notice

Table of Contents

1. Introduction

The Domain Name System Security Extensions (DNSSEC), which are broadly defined in [RFC4033], [RFC4034] and [RFC4035], use cryptographic keys and digital signatures to provide data origin authentication and data integrity in the DNS. This document describes the application of Merkle Tree Ladder (MTL) Mode to the Stateless Hash-Based Digital Signature Algorithm (SLH-DSA) as the SLH-DSA-MTL signature scheme for DNSSEC. SLH-DSA is described in the FIPS 205 standard [FIPS205] and MTL mode is described in [I-D.harvey-cfrg-mtl-mode]. As described herein, a DNSKEY resource record (RR) for an SLH-DSA-MTL key contains a SLH-DSA key. The SLH-DSA key is used for verifying signatures on Merkle tree ladders (MTLs). An RRSIG resource record for an SLH-DSA-MTL Signature contains a Merkle proof (authentication path) that is verifiable using a MTL, and optionally also contains the signed MTL.

The anticipation of quantum computers that can break the current signature algorithms led to NIST selecting post-quantum cryptographic (PQC) algorithms for standardization and developing specifications for the algorithms as NIST standards. These new algorithms are expected to replace classical digital signature algorithms (e.g., RSA and ECDSA) in IETF standards and to be widely implemented and deployed after that. NIST’s proposed PQC algorithms have significantly larger signature sizes than RSA and ECDSA. The larger sizes may have a significant operational impact on DNSSEC. For example, the size of signed NSEC and NSEC3 responses may exceed UDP MTUs with this degrading the use of UDP as the prevalent DNSSEC transport. Larger signature sizes could also substantially increase memory requirements for in-memory zone databases used by authoritative name servers and for in-memory caches used by resolvers.

As described in [I-D.harvey-cfrg-mtl-mode], MTL mode is designed to reduce the size impact of PQC signature algorithms. For DNSSEC, the size impact reduction is achieved when signatures provided in RRSIG RRs are primarily comprised of "condensed signatures" (Merkle proofs / authentication paths) and are only occasionally comprised of "full signatures" that contain both a condensed signature and a signed MTL, where the signed ladder includes a signature using the underlying PQC signature algorithm. MTL mode reduces the memory requirements for PQC signatures as the signature data in the zone database or cache is primarily comprised of Merkle proofs and only occasionally of signed MTLs [CTRSAMTL].

SLH-DSA is a stateless hash-based PQC signature scheme selected by NIST for standardization [NISTSELECTIONS] in July 2022 and formally published as a standard in August 2024 [FIPS205]. This document

specifies SLH-DSA for the initial application of MTL mode to DNSSEC based on three considerations: (1) SLH-DSA is also based on Merkle trees, and thus already has internal functions for computing leaf nodes and internal nodes; and (2) SLH-DSA has relatively large signature sizes and computational costs, and therefore can benefit significantly from the reductions offered by MTL mode; and (3) hash-based techniques are well understood and offer a conservative choice for long-term security relative to newer NIST selected signature schemes based on lattice-based cryptography. SLH-DSA is based on SPHINCS+ [SPHINCSPLUS], one of the submissions to NIST's PQC evaluation project [I-D.harvey-cfrg-mtl-mode] describes the combination of MTL mode with SLH-DSA.

This initial version of the draft focuses on the code-points applicable to DNSKEY and RRSIG formulation and a proposed DNSSEC protocol change to support retrieval of MTL mode condensed signatures and MTL mode full signatures as described in Section 3, Section 9.4, and Section 9.5 of [I-D.harvey-cfrg-mtl-mode]. Later versions may describe DNSSEC protocol and/or operational changes related to zone signing, zone composition, zone updates, zone transfer, name server processing, resolver signature processing, and resolver caching.

2. Conventions Used in This Document

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

Double pipe characters, "||" are used in this document to indicate concatenation of the elements preceding and following the double pipe characters.

All numeric DNSKEY elements and RRSIG elements specified in this document are unsigned integers in network byte order (big endian order).

3. DNSKEY Resource Records

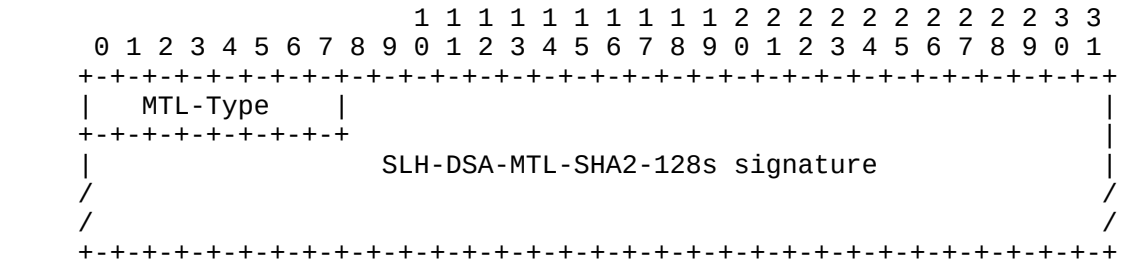
An SLHDSAMTLSHA2128S key consists of a 32-octet value, which is encoded into the Public Key field of a DNSKEY resource record as a simple bit string. SLHDSAMTLSHA2128S keys are generated as SLH-DSA keys using the SLH-DSA-SHA2-128s parameter set, as defined in 10.1 and 11 of [FIPS205].

An SLHDSAMTLSHAKE128S key consists of a 32-octet value, which is encoded into the Public Key field of a DNSKEY resource record as a simple bit string. SLHDSAMTLSHAKE128S keys are generated as SLH-DSA keys using the SLH-DSA-SHAKE-128s parameter set, as defined in 10.1 and 11 of [FIPS205].

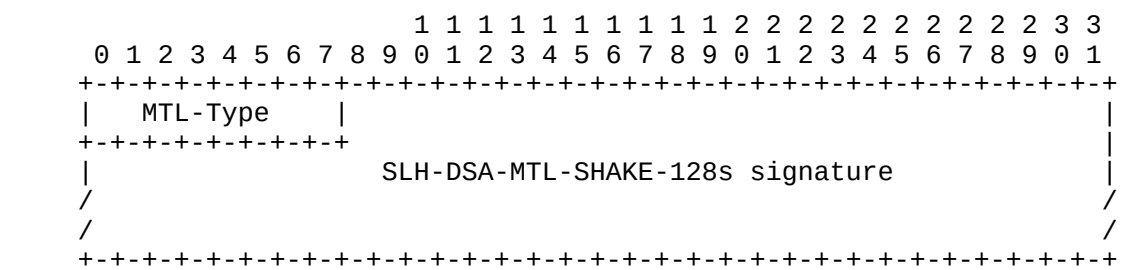
4. RRSIG Resource Records

MTL mode signatures are either full or condensed as described in [I-D.harvey-cfrg-mtl-mode]. SLHDSAMTLSHA2128S and SLHDSAMTLSHAKE128S signatures utilize a one-octet prefixed MTL-Type field to indicate whether the signature is condensed (0) or full (1).

An SLHDSAMTLSHA2128S signature consists of a variable-length value, which is encoded into the Signature field of an RRSIG resource record as a simple bit string as the concatenation of the MTL-Type and a SLH-DSA-MTL-SHA2-128s signature as described in [I-D.harvey-cfrg-mtl-mode]:



An SLHDSAMTLSHAKE128S signature consists of a variable-length value, which is encoded into the Signature field of an RRSIG resource record as a simple bit string as the concatenation of the MTL-Type and a SLH-DSA-MTL-SHAKE-128s signature as described in [I-D.harvey-cfrg-mtl-mode]:



The signature and verification algorithms for both SLH-DSA-MTL-SHA2-128s and SLH-DSA-MTL-SHAKE-128s are described in 9.1 and 9.2 of [I-D.harvey-cfrg-mtl-mode]. The signature and verification

algorithms for the underlying signature algorithms used for signing ladders in SLH-DSA-MTL-SHA2-128s and SLH-DSA-MTL-SHAKE-128s full signatures, SLH-DSA-SHA2-128s and SLH-DSA-SHAKE-128s respectively, are described in 10.2 and 10.3 of [FIPS205].

5. Algorithm Numbers for DS, DNSKEY, and RRSIG Resource Records

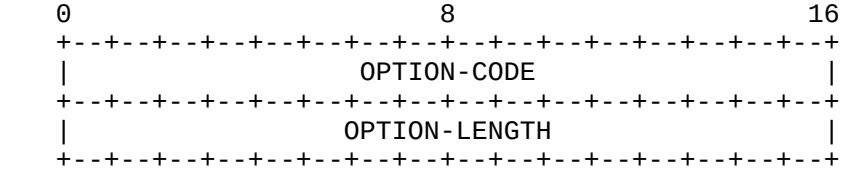
The algorithm number associated with the use of SLHDSAMTLSHA2128S in DS, DNSKEY, and RRSIG resource records is TBD. The algorithm number associated with the use of SLHDSAMTLSHAKE128S in DS, DNSKEY, and RRSIG resource records is TBD. This registration is fully defined in the IANA Considerations section.

6. The mtl-mode-full EDNS(0) Option

MTL mode signatures are either full or condensed. A MTL mode-aware client MAY request that signatures be returned in the full format by providing the mtl-mode-full EDNS(0) option in the OPT meta-RR of its query [RFC6891].

6.1. Option Format

The mtl-mode-full option is encoded as follows:



Where:

OPTION-CODE The EDNS0 option code assigned to mtl-mode-full, TBD.
OPTION-LENGTH Always zero.

6.2. Use By Responders

When a query includes the mtl-mode-full option, the response requirement depends on the number of RRSIG records in the response that were produced in MTL mode:

* If exactly one RRSIG record in the response was produced in MTL mode, then that RRSIG record MUST be returned in the full signature format.

* If more than one RRSIG record in the response was produced in MTL mode, then enough of these RRSIG records MUST be returned in the full signature format to ensure that every other RRSIG in the response that was produced in MTL mode can be verified.

When the mtl-mode-full option is not included, every signature in the response that was produced in MTL mode MUST be returned in the condensed signature format.

As described in 9.2 of [I-D.harvey-cfrg-mtl-mode], when a verifier receives a condensed signature, the verifier determines whether any of the MTLs it has previously verified includes a rung that is compatible with the authentication path in the condensed signature. If not, then the verifier requests a new signed ladder. Accordingly, a resolver SHOULD first query a name server without the mtl-mode-full option, and then, if needed, re-issue the query with the mtl-mode-full option. Since responses to queries with the mtl-mode-full option are expected to be large, it is RECOMMENDED that queries with the mtl-mode-full option be issued over transports (e.g., TCP, TLS, QUIC) that support large responses without truncation and/or fragmentation.

7. Implementation Considerations

Signing RRSets in batches rather than as individual messages can leverage MTL mode to reduce the number of public/private key signing operations performed with the underlying signature algorithm. This results in reducing the average computational overhead per message signed. This practice can also reduce the load on a hardware security module. Batches also benefit the verifier by reducing the number of full signatures required for validation because multiple RRSIGs can be verified by the ladder covering the batch. The appropriate batch size will depend on the properties of the zone and the requirements of the zone operator. Batch size needs to be considered carefully to ensure that new signatures are available in a timely manner while still gaining the benefits of batch signing [MTL-ENDURANCE].

8. Examples

Examples with detailed processing descriptions are found in Appendix A

9. IANA Considerations

This document updates the IANA registry for DNSSEC "Domain Name System Security (DNSSEC) Algorithm Numbers" located at https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml (https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml). The following entries are requested to be added to the registry subject to the Number update:

SLH-DSA-MTL-SHA2-128s

Number	TBD
Description	SLH-DSA-MTL-SHA2-128s
Mnemonic	SLHDSAMTLSHA2128S
Zone Signing	Y
Trans. Sec.	*
Reference	This specification

SLH-DSA-MTL-SHAKE-128s

Number	TBD
Description	SLH-DSA-MTL-SHAKE-128s
Mnemonic	SLHDSAMTLSHAKE128S
Zone Signing	Y
Trans. Sec.	*
Reference	This specification

* There has been no determination of standardization of the use of these algorithms with Transaction Security.

10. Implementation Status

NOTE: Please remove this section and the reference to RFC 7942 prior to publication as an RFC.

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".

For testing purposes, SLH-DSA-MTL-DNSSEC has been implemented in the following DNS open-source applications:

* LDNS for key generation, zone signing, and zone verification with MTL mode: https://github.com/Verisign/mtl-mode-ldns (https://github.com/Verisign/mtl-mode-ldns)
* NSD authoritative name server with MTL mode: https://github.com/NLnetLabs/nsd/pull/397 (https://github.com/NLnetLabs/nsd/pull/397)
* Unbound recursive resolver with MTL mode: https://github.com/Verisign/mtl-mode-unbound (https://github.com/Verisign/mtl-mode-unbound)

These implementations depend on the reference implementation of MTL mode which is available in C. The MTL library can be found at https://github.com/Verisign/MTL (https://github.com/Verisign/MTL).

11. Security Considerations

The security considerations of [FIPS205] and [I-D.harvey-cfrg-mtl-mode] are inherited in the usage of SLH-DSA-MTL in DNSSEC.

SLH-DSA-MTL-SHA2-128s and SLH-DSA-MTL-SHAKE-128s are intended to operate at around the 128-bit security level against classical attacks and the 64-bit level against quantum attacks, consistent with NIST’s security level I.

A private key used for a DNSSEC zone MUST NOT be used for any other purpose than for that zone. Otherwise, cross-protocol or cross-application attacks are possible.

12. Acknowledgements

The authors would like to acknowledge the following individuals for their contributions to the development of this document: Scott Hollenbeck, Swapneel Sheth. This I-D has drawn from helpful examples of document structure and specification text from various DNSSEC algorithm RFCs. The authors express their gratitude to the authors of those RFCs for their contributions.

13. References

13.1. Normative References

[FIPS205] National Institute of Standards and Technology (NIST), "Stateless Hash-Based Digital Signature Standard", FIPS PUB 205, DOI 10.6028/NIST.FIPS.205, 13 August 2024, <https://doi.org/10.6028/NIST.FIPS.205>.

[I-D.harvey-cfrg-mtl-mode] Harvey, J., Kaliski, B., Fregly, A., and S. Sheth, "Merkle Tree Ladder (MTL) Mode Signatures", Work in Progress, Internet-Draft, draft-harvey-cfrg-mtl-mode-07, 9 September 2025, <https://datatracker.ietf.org/doc/html/draft-harvey-cfrg-mtl-mode-07>.

[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>.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

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

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

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

[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>.

13.2. Informative References

[CTRSAMTL] Kaliski, B., Fregly, A.M., Harvey, J., and S. Sheth, "Merkle Tree Ladder Mode: Reducing the Size Impact of NIST PQC Signature Algorithms in Practice", 2023.

[MTL-ENDURANCE] Tran, M. and T. Chung, "Randomized Evaluation of SLH-DSA-MTL's Impact on Reducing PQ-DNSSEC Signature Sizes", 18 March 2025, <https://github.com/IQTF/pq-dnssec-materials/raw/refs/heads/main/IETF122/Tran_Randomized_evaluation_of_SLH-DSA-MTL's_impact_on_reducing_PQ-DNSSEC_signature_sizes.pdf>.

[NISTSELECTIONS] National Institute of Standards and Technology (NIST), "Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process", June 2022, <https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf>.

[SPHINCSPLUS] Bernstein, D., Huelsing, A., Koelbl, S., Niederhagen, R., Rijneveld, J., and P. Schwabe, "The SPHINCS+ Signature Framework", Cryptology ePrint Archive, Report 2019/1086, 2019, <https://eprint.iacr.org/2019/1086.pdf>.

Appendix A. MTL Mode for DNSSEC Example

This appendix gives an example. The appendix also provides a step-by-step overview of how to verify an example condensed signature and an example full signature from the signed zone file. See [I-D.harvey-cfrg-mtl-mode] for additional details on the cryptographic operations.

In the following, byte strings are written in hexadecimal. For readability, a space or line break is inserted after each group of four bytes (eight hexadecimal characters). For example, the six-byte string with decimal byte values 1, 2, 4, 8, 16, 32 is written

01020408 1020

The function toByte(x,n) converts the integer x to a n-byte string, most significant byte first. (The function is defined in [FIPS205].) For example, toByte(16,4) produces the four-byte string

00000010

toByte assumes that 0 <= x <= 2^{8y}-1. This assumption holds in all calls to toByte within this appendix.

NOTE: For purposes of illustration we assigned the numeric DNSSEC algorithm identifier 50 for SLH-DSA-MTL-SHA2-128s. We plan to change to an experimental identifier in a future version of this draft, and before publishing any code for MTL mode for DNSSEC.

A.1. Initial Signed Zone File

The example zone file below includes several RRsets associated with the example.com zone. The SOA RRset has a full signature, while the A, AAAA, CNAME, MX, NS, NSEC3 and TXT RRsets each has a condensed signature. The DNSKEY RRset is unsigned. In practice, the DNSKEY RRset would be signed with a key signing key. We omitted this step for simplicity in this version of the draft. We plan to add sign the DNSKEY RRset with a MTL mode key signing key in the next version of the draft.

Any number of the signed RRsets in the zone file could have a full signature. We associated the full signature with the SOA record because the SOA record is updated whenever the zone changes. The condensed signatures on the other RRsets are all relative to the signed ladder in the full signature in the SOA RRSIG record. The corresponding full signature on an RRset can be formed by concatenating the condensed signature on the RRset with the signed ladder in the SOA RRSIG record’s full signature -- see Section 9.4 of [I-D.harvey-cfrg-mtl-mode]. As a result, a name server that loads this zone file can form a full signature on any of the RRsets when requested per Section 6 above, without access to the signer’s private key material.

The full signature is abridged in the example below. The complete value is given in Appendix A.6.

NOTE: The TXT record represented in the zone file below has been broken into two lines to fit in this Internet-Draft. Verifying the signature on the TXT record requires that the text (including spaces) match the source record which is a single line that reads: "This zone is an example input for SLH-DSA-MTL zone signing" with single spaces between each of the words.

```
example.com. 3600 IN SOA ns.example.com. admin.example.com. (
                            1719858941 7200 3600 1209600 3600 )
example.com. 3600 IN RRSIG SOA 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            AWOXFesN5grvg1Vk/TE3ZNEAAEkgbrJ3DnyxAAA
                            AAgAAAAAAAAAHAANsVqmmBNLfHo2J8nnZz+kcir
                            50wSllXgmtilZzYqNXNtPjWTkxvxviqKtdIWEZh
                            hIAAEkgbrJ3DnyxAAIAAAAAAAAAB0wqgHBF0FWf
                            pS3J9JgTrXoAAAAIAAAACI ) # ... abridged
example.com. 3600 IN A 192.0.2.1
example.com. 3600 IN RRSIG A 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            APnCCOkVSqjw6zKSPz40U6AAAEkgbrJ3DnyxAAA
                            AAAAAAAAAAAAHAAOGVodklRgciVyAG660gDJAS/
                            blgaqTfYU04u9LWETNe9PjWTkxvxviqKtdIWEZh
                            hI= )
example.com. 3600 IN NS ns1.example.net.
example.com. 3600 IN NS ns2.example.net.
example.com. 3600 IN RRSIG NS 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            APVLlIBjy13ydSa9FxADHF4AAEkgbrJ3DnyxAAA
                            AAQAAAAAAAAAHAAN5pQH0FHJTRUCYkOBtwexgS/
                            blgaqTfYU04u9LWETNe9PjWTkxvxviqKtdIWEZh
                            hI= )
example.com. 3600 IN MX 10 mail.example.net.
example.com. 3600 IN RRSIG MX 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            ALnLaReRJQiI5Zo1LcM/ajEAAEkgbrJ3DnyxAAA
                            AAwAAAAAAAAAHAAPO+30qRFTOs9aFxBzbQTVJir
                            50wSllXgmtilZzYqNXNtPjWTkxvxviqKtdIWEZh
                            hI= )
example.com. 3600 IN TXT "This zone is an example input for
                            SLH-DSA-MTL zone signing"
example.com. 3600 IN RRSIG TXT 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            ADo++BxJN5KgDczdjzW9yyoAAEkgbrJ3DnyxAAA
                            ABAAAAAAAAAAHAANIBHbegIOSEdvxj8FpuwUhzg
                            KJmdG75STS6V/0/RqEvdINr1pRx28N2ClBwmX0j
                            wI= )
example.com. 3600 IN AAAA 2001:db8::1
example.com. 3600 IN RRSIG AAAA 50 2 3600 20250701183541 (
```

```
                                20240701183541 53939 example.com.
                                AIiR3ec5YTYyufoN4/m6mfcAAEkgbrJ3DnyxAAA
                                ABQAAAAAAAAAHAAMCqwQKN/jTi7+3gCImVZr9zg
                                KJmdG75STS6V/0/RqEvdINr1pRx28N2ClBwmX0j
                                wI= )
     example.com. 3600 IN DNSKEY 256 3 50 (
                                PawPGCKuykH6QOtfh6b8HoJZw4xMM+3QKvsTgo
                                T/5/8= ;{id = 53939 (zsk), size = 0b} )
     9vq38lj9qs6s1aruer131mbtsfnvek2p.example.com. 3600 IN NSEC3 1 0 (
                                1 -  0lverorlcjoa2lji5rik0otij3lgoj3l
                                A NS SOA MX TXT AAAA RRSIG DNSKEY )
     9vq38lj9qs6s1aruer131mbtsfnvek2p.example.com. 3600 IN RRSIG (
                                NSEC3 50 3 3600 20250701183541
                                20240701183541 53939 example.com.
                                AFLTit749Nqqdkh+etQwoDkAAEkgbrJ3DnyxAAA
                                ABgAAAAAAAAAHAAMDtIHLhQIPR4YdqvKF++jwvr
                                4HJ28uILKC7IXrGCYpWNINr1pRx28N2ClBwmX0j
                                wI= )
     www.example.com. 3600 IN CNAME example.com.
     www.example.com. 3600 IN RRSIG CNAME 50 3 3600 (
                                20250701183541 20240701183541 53939
                                example.com.
                                ABaMIKiaAl8rpjCN1unR9zgAAEkgbrJ3DnyxAAA
                                ABwAAAAAAAAAHAAODZdDLIaNHOsGFK2ydA637vr
                                4HJ28uILKC7IXrGCYpWNINr1pRx28N2ClBwmX0j
                                wI= )
     0lverorlcjoa2lji5rik0otij3lgoj3l.example.com. 3600 IN NSEC3 1 0 (
                                1 -  9vq38lj9qs6s1aruer131mbtsfnvek2p
                                CNAME RRSIG )
     0lverorlcjoa2lji5rik0otij3lgoj3l.example.com. 3600 IN RRSIG (
                                NSEC3 50 3 3600 20250701183541
                                20240701183541 53939 example.com.
                                AD3B1TW3oNgurikkoA+mxSgAAEkgbrJ3DnyxAAA
                                ACAAAAAgAAAAIAAA= )
```

A.2. Obtaining the Public Key

As usual in DNSSEC, the verifier obtains the public key for verifying signatures from the DNSKEY RRset (which in this example includes only one record):

```
  example.com. 3600 IN DNSKEY 256 3 50 (
                                PawPGCKuykH6QOtfh6b8HoJZw4xMM+3QKvsTgo
                                T/5/8= ; key id = 53939 )
```

Following Section 2.2 of [RFC4034], the RDATA portion of this record (the fields to the right of "DNSKEY") includes the following fields:

* Flag: 256 (zone key bit = 1)
* Protocol: 3 (fixed value)
* Algorithm: 50 (SLH-DSA-MTL-SHA2-128s)
* Public Key: PawPGCKuykH6QOtfh6b8HoJZw4xMM+3QKvsTgoT/5/8= [44 characters in Base64]

The key tag for this public key, as shown in the comments, is 53939 (decimal). (The key tag is computed from the public key following Appendix B of [RFC4034].) The Base64 value of the Public Key field corresponds to the following byte string:

```
3DAC0F18 22AECA41 FA40EB5F 87A6FC1E
8259C38C 4C33EDD0 2AFB1382 84FFE7FF [32 bytes]
```

The verifier parses the byte string following [FIPS205] to obtain the public key components:

```
3DAC0F18 22AECA41 FA40EB5F 87A6FC1E - PK.seed [16 bytes]
8259C38C 4C33EDD0 2AFB1382 84FFE7FF - PK.root [16 bytes]
```

A.3. Verifying a Condensed Signature

This section illustrates how the example A RRSIG record can be verified. Other RRSIG records with condensed signatures can be verified similarly. The example A RRSIG record is:

```
example.com. 3600 IN RRSIG A 50 2 3600 20250701183541 (
                            20240701183541 53939 example.com.
                            APnCCOkVSqjw6zKSPz40U6AAAEkgbrJ3DnyxAAA
                            AAAAAAAAAAAAHAAOGVodklRgciVyAG660gDJAS/
                            blgaqTfYU04u9LWETNe9PjWTkxvxviqKtdIWEZh
                            hI= )
```

Following Section 3.2 of [RFC4034], the RDATA portion of this record includes the following fields:

* Type Covered: A
* Algorithm: 50 (SLH-DSA-MTL-SHA2-128s)
* Labels: 2
* Original TTL: 3600 seconds
* Signature Expiration: 1 July 2025 18:35:41 UTC
* Signature Inception: 1 July 2024 18:35:41 UTC
* Key Tag: 53939
* Signer’s Name: "example.com."
* Signature: APnCCOkVSqjw6zKSPz40U6AAAEkgbrJ3DnyxAAAAAAAAAAAAAAAAHAAO GVodklRgciVyAG660gDJAS/blgaqTfYU04u9LWETNe9PjWTkxvxviqKtdIWEZhhI= [120 characters in Base64]

The Base64 value of the Signature field corresponds to the following byte string:

```
00F9C208 E9154AA8 F0EB3292 3F3E3453 A0000049 206EB277
0E7CB100 00000000 00000000 00000700 03865687 6495181C
895C801B AEB48032 404BF6E5 81AA937D 8534E2EF 4B5844CD
7BD3E359 3931BF1B E2A8AB5D 21611986 12 [89 bytes]
```

Per Section 4 of this document, the initial 00 byte of the byte string indicates that the signature is in condensed format. The remaining 88 bytes are the condensed signature.

A.3.1. Parsing the Condensed Signature

The verifier parses the condensed signature to obtain the randomizer, the series identifier, the authentication path and other information following Section 9.5 of [I-D.harvey-cfrg-mtl-mode].

For the example A RRSIG record, the parsing produces these fields:

Randomizer

```
F9C208E9 154A8F0 EB32923F 3E3453A0 - randomizer R_mtl [16 bytes]
```

Authentication Path

```
0000 - flags (must be 0 per [I-D.harvey-cfrg-mtl-mode]) [2 bytes]
49206EB2 770E7CB1 - series identifier SID [8 bytes]
00000000 - leaf index: i = 0 [4 bytes]
00000000 - rung left index: 0 [4 bytes]
00000007 - rung right index: 7 [4 bytes]
0003 - sibling hash count: 2 [2 bytes]
    Sibling node hash values
86568764 95181C89 5C801BAE B4803240 - V[1:1] [16 bytes]
4BF6E581 AA937D85 34E2EF4B 5844CD7B - V[2:3] [16 bytes]
D3E35939 31BF1BE2 A8AB5D21 61198612 - V[4:7] [16 bytes]
```

The authentication path for this signature connects the leaf node hash value V[0:0] to the ladder rung V[0:7] (see Appendix A.4.1). The sibling node hash values are denoted V[1:1], V[2:3] and V[4:7]. (In an implementation, a verifier may receive an authentication path with a different number of hash values and/or different actual values than the signer intended. The authentication path verification operation, e.g., Section 8.8 of [I-D.harvey-cfrg-mtl-mode], would check that both the number and values are correct.)

A.3.2. Forming the MTL Mode Message Input

The verifier forms the message input M[i] to the MTL mode verification operation following DNSSEC conventions specified in Section 3.1.8.1 of [RFC4034]: it is the concatenation of the wire format of the RDATA portion of the associated RRSIG record excluding the Signature field, and the wire format of the associated RRset. The value produced by this step for the example A RRset and its associated RRSIG record is:

```
M[0] = 00013202 00000E10 68642A7D 6682F6FD D2B30765 78616D70
       6C650363 6F6D0007 6578616D 706C6503 636F6D00 00010001
       00000E10 0004C000 0201 [58 bytes]
```

NOTE: For cryptography implementers not familiar with DNSSEC, the message bytes of M[0] can be parsed as follows:

RDATA portion of RRSIG (excluding Signature field) wire format

```
0001 - Type Covered: 1 (A) [2 bytes]
32 - Algorithm: 50 (SLH-DSA-MTL-SHA2-128s) [1 byte]
02 - Labels: 2 [1 byte]
00000E10 - Original TTL:  3600 seconds [4 bytes]
68642A7D - Sig Expiration: 1 July 2025 18:35:41 UTC [4 bytes]
6682F6FD - Sig Inception: 1 July 2024 18:35:41 UTC  [4 bytes]
D2B3 - Key Tag: 53939 [2 bytes]
07657861 6D706C65 03636F6D 00 - Signer’s Name:
                                "example.com." [variable]
```

RRset wire format

```
07657861 6D706C65 03636F6D 00 - Owner Name:
                                "example.com." [variable]
0001 - Type:  1 (A) [2 bytes]
0001 - Class:  1 (IN) [2 bytes]
00000E10 - Time to Live:  3600 seconds [4 bytes]
0004 - length in bytes of RDATA portion: 4 [2 bytes]
C0000201 - RDATA portion: Host Address (192.0.2.1) [4 bytes]
```

A.3.3. Computing the Leaf Node Hash Value

The verifier computes the leaf node hash value V[i] from the message M[i], the per-message randomizer R_mtl[i] and certain other information following Sections 5.1 and 8.2.1 of [I-D.harvey-cfrg-mtl-mode]. The process has two steps:

* Hash the message M[i] together with the randomizer R_mtl[i], the public key seed PK.seed, the public key root PK.root, and an address field to obtain a data value d[i] following Section 5.1 of [I-D.harvey-cfrg-mtl-mode]
* Hash the data value d[i] together the public key seed PK.seed and a compressed address field to obtain a leaf node hash value V[i] following Section 8.2.1 of [I-D.harvey-cfrg-mtl-mode]

For SLH-DSA-MTL-SHA2-128s, the steps simplify to the following operations:

```
ADRS[i] = toByte(0,8) || SID || toByte(16,4) || toByte(0,8) ||
          toByte (i,4)
d[i] = MGF1-SHA2-256 (R_mtl[i] || PK.seed || SHA2-256 (R_mtl[i] ||
          PK.seed || PK.root || toByte(128,1) || toByte (0,1) ||
          ADRS[i] || M[i]), 16)
ADRS^c[i] = toByte(0,1) || SID || toByte(17,1) || toByte(0,8) ||
          toByte (i,4)
V[i] = SHA2-256 (PK.seed || toByte(0,48) || ADRS^c[i] || d[i])
          truncated to the first 16 bytes
```

The leaf node hash value V[i] is alternatively denoted V[i:i] when input to the internal node hash value operations in the next section.

For the example record, the values involved are:

```
SID = 49206EB2 770E7CB1 [8 bytes]
ADRS[0] = 00000000 00000000 49206EB2 770E7CB1
          00000010 00000000 00000000 00000000 [32 bytes]
R_mtl[0] = F9C208E9 154AA8F0 EB32923F 3E3453A0 [16 bytes]
PK.seed = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E [16 bytes]
PK.root = 8259C38C 4C33EDD0 2AFB1382 84FFE7FF [16 bytes]
M[0] = 00013202 00000E10 68642A7D 6682F6FD D2B30765 78616D70
       6C650363 6F6D0007 6578616D 706C6503 636F6D00 00010001
       00000E10 0004C000 0201 [58 bytes]
SHA-256 input within MGF1-SHA2-256 call = F9C208E9 154AA8F0
       EB32923F 3E3453A0 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E
       8259C38C 4C33EDD0 2AFB1382 84FFE7FF 80000000 00000000
       00004920 6EB2770E 7CB10000 00100000 00000000 00000000
       00000001 32020000 0E106864 2A7D6682 F6FDD2B3 07657861
       6D706C65 03636F6D 00076578 616D706C 6503636F 6D000001
       00010000 0E100004 C0000201  [140 bytes]
SHA-256 full output within MGF1-SHA-256 call = 020D9241 F02420F6
       5855C6AA DAA82B18 F9F4E13E 78BF6C63 7ABA745A 593B5DB4
       [32 bytes]
MGF1-SHA-256 input = F9C208E9 154AA8F0 EB32923F 3E3453A0 3DAC0F18
       22AECA41 FA40EB5F 87A6FC1E 020D9241 F02420F6 5855C6AA
       DAA82B18 F9F4E13E 78BF6C63 7ABA745A 593B5DB4 [64 bytes]
d[0] = MGF1-SHA2-256 output = 3564B082 F8E79D9D 31B8BA7C B05E9EB7
       [16 bytes]
ADRS^c[0] = 0049206E B2770E7C B1110000 00000000 00000000 0000
       [22 bytes]
SHA2-256 input for V[0] = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E
       00000000 00000000 00000000 00000000 00000000 00000000
       00000000 00000000 00000000 00000000 00000000 00000000
       0049206E B2770E7C B1110000 00000000 00000000 00003564
       B082F8E7 9D9D31B8 BA7CB05E 9EB7 [102 bytes]
V[0:0] = V[0] = SHA2-256 output truncated to 16 bytes = 79A501F4
       14725345 409890E0 6DC1EC60 [16 bytes]
```

Note. The simplified operations given above for SLH-DSA-MTL-SHA2-128s can be derived from [I-D.harvey-cfrg-mtl-mode] as follows:

1. The address value ADRS[i] has type MTL_MSG = 16 per Section 4.6 of [I-D.harvey-cfrg-mtl-mode].
2. The randomized hash function H_mtl_msg is the MGF1-SHA2-256 / SHA-256 combination defined in Section 10.2.1 of [I-D.harvey-cfrg-mtl-mode]. The output of the call to SHA2-256 within this combination is not truncated; it remains 32 bytes when input to MGF1-SHA2-256.
3. The message input to H_mtl_msg is a message domain separator of type MTL_MSG_SEP = 128 with an empty context string, followed by ADRS[i], followed by the actual message M[i], per Section 5.1 of [I-D.harvey-cfrg-mtl-mode].

4. The compressed address value ADRS^c[i] has type MTL_DATA = 17 per Section 4.6 of [I-D.harvey-cfrg-mtl-mode].
5. The leaf node hash function F is SHA2-256 where PK.seed is padded with zeroes on the right to 64 bytes per Section 10.2.3 of [I-D.harvey-cfrg-mtl-mode].

A.3.4. Checking the Authentication Path

The verifier checks the authentication path from the leaf node hash value V[i:i] to a ladder rung following Section 8.8 of [I-D.harvey-cfrg-mtl-mode]. The ladder rung is obtained separately, either by requesting a full signature on the same RRset as described in Section 6 of this document (see also Appendix A.1), or from a full signature previously requested (and remembered) for a different RRset.

The authentication path checking process involves one or more iterations of this step:

* Hash a left node hash value and a right node hash value together with the public key seed PK.seed and an address field to obtain an internal node hash value

For SLH-DSA-MTL-SHA2-128s, the step simplifies to one or more operations of the following form:

* ADRS^c[L:R] = toByte(0,1) || SID || toByte(18,1) || toByte(0,4) || toByte (L,4)|| toByte (R,4)
* V[L:R] = SHA2-256 (PK.seed || toByte(0,48) || ADRS^c[L:R] || V[L:M-1] || V[M:R]) truncated to the first 16 bytes

Here, V[L:R] is the internal node hash value being computed and V[L:M-1] and V[M:R] are its child left and right node hash values. Following [I-D.harvey-cfrg-mtl-mode], M is the unique integer between L+1 and R that is divisible by the largest power of two.

For the example record, the process involves two iterations (following the Merkle node set structure in Appendix A.5.2 from leaf to rung):

* Hash V[0:0] and V[1:1] together with PK.seed and an address field to obtain V0:1 (L = 0, R = 1, M = 1)
* Hash V[0:1] and V[2:3] together with PK.seed and an address field to obtain V0:3 (L = 0, R = 3, M = 2)
* Hash V[0:3] and V[4:7] together with PK.seed and an address field to obtain V0:7 (L = 0, R = 7, M = 4)

V[0:0] = V[0] was computed in Appendix A.3.3, while V[0:1], V[2:3] and V[4:7] were obtained from the authentication path in Appendix A.3.1.

The values involved are:

```
SID = 49206EB2 770E7CB1 [8 bytes]
ADRS^c[0:1] = 0049206E B2770E7C B1120000 00000000 00000000 0001
              [22 bytes]
PK.seed = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E [16 bytes]
V[0:0] = 79A501F4 14725345 409890E0 6DC1EC60 [16 bytes]
V[1:1] = 86568764 95181C89 5C801BAE B4803240 [16 bytes]
SHA2-256 input = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E 00000000
              00000000 00000000 00000000 00000000 00000000 00000000
              00000000 00000000 00000000 00000000 00000000 0049206E
              B2770E7C B1120000 00000000 00000000 000179A5 01F41472
              53454098 90E06DC1 EC608656 87649518 1C895C80 1BAEB480
              3240 [118 bytes]
V[0:1] = SHA-256 output truncated to 16 bytes = 8ABE74C1 29655E09
              AD8A5673 62A35736 [16 bytes]
ADRS^c[2:3] = 0049206E B2770E7C B1120000 00000000 00000002 0003
              [22 bytes]
V[2:3] = 4BF6E581 AA937D85 34E2EF4B 5844CD7B [16 bytes]
SHA2-256 input = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E 00000000
              00000000 00000000 00000000 00000000 00000000 00000000
              00000000 00000000 00000000 00000000 00000000 0049206E
              B2770E7C B1120000 00000000 00000000 00038ABE 74C12965
              5E09AD8A 567362A3 57364BF6 E581AA93 7D8534E2 EF4B5844
              CD7B [118 bytes]
V[0:3] = SHA2-256 output truncated to 16 bytes = D20DAF5A 51C76F0D
              D82941C2 65F48F02 [16 bytes]
ADRS^c[4:7] = 0049206E B2770E7C B1120000 00000000 00000004 0007
              [22 bytes]
V[4:7] = D3E35939 31BF1BE2 A8AB5D21 61198612 [16 bytes]
SHA2-256 input = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E 00000000
              00000000 00000000 00000000 00000000 00000000 00000000
              00000000 00000000 00000000 00000000 00000000 0049206E
              B2770E7C B1120000 00000000 00000000 0007D20D AF5A51C7
              6F0DD829 41C265F4 8F02D3E3 593931BF 1BE2A8AB 5D216119
              8612 [118 bytes]
V[0:7] = SHA2-256 output truncated to 16 bytes = 4C2A8070 45D0559F
              A52DC9F4 9813AD7A [16 bytes]
```

Figure 1

The internal node hash value V[0:7] matches the corresponding rung in the ladder (see Appendix A.4.1), so the authentication path is verified.

Note. The simplified operations given above for SLH-DSA-MTL-SHA2-128s can be derived from [I-D.harvey-cfrg-mtl-mode]as follows:

1. The compressed address value ADRS^c[i] has type MTL_TREE = 18 as defined in Section 4.6 of [I-D.harvey-cfrg-mtl-mode].
2. The leaf node hash function H is SHA-256 where PK.seed is padded with zeroes on the right to 64 bytes, as defined in Section 10.2.3 of [I-D.harvey-cfrg-mtl-mode].

A.4. Verifying a Full Signature

The RRSIG record for the example SOA RRset includes a full signature. The abridged Base64 value of the signature field of the RRSIG record is:

```
AWOXFesN5grvg1Vk/TE3ZNEAAEkgbrJ3DnyxAAAAAgAAAAAAAAAHAANsVqmmBNLfHo2
J8nnZz+kcir50wSllXgmtilZzYqNXNtPjWTkxvxviqKtdIWEZhhIAAEkgbrJ3DnyxAA
IAAAAAAAAAB0wqgHBF0FWfpS3J9JgTrXoAAAAIAAAACIqAre8NNFy48Tcs96QkJKAAA
B6w3N7mZva9FQDM ...
```

This value corresponds to the following abridged byte string:

```
01639715 EB0DE60A EF835564 FD313764 D1000049 206EB277
0E7CB100 00000200 00000000 00000700 036C56A9 A604D2DF
1E8D89F2 79D9CFE9 1C8ABE74 C129655E 09AD8A56 7362A357
36D3E359 3931BF1B E2A8AB5D 21611986 12000049 206EB277
0E7CB100 02000000 00000000 074C2A80 7045D055 9FA52DC9
F49813AD 7A000000 08000000 088A80AD EF0D345C B8F1372C
F7A42424 A000001E B0DCDEE6 66F6BD15 00CC ...
```

The complete Base64 value and byte string are given in Appendix A.6. Per Section 4 of this document, the initial 01 byte of this string indicates that the signature is in full format. The remaining 7856 bytes are the full signature.

A.4.1. Parsing the Full Signature

The verifier parses the full signature to obtain the randomizer, the series identifier, the authentication path, the ladder, the underlying signature on the ladder and other information following Section 9.4 of [I-D.harvey-cfrg-mtl-mode].

For the example record, the parsing produces these fields:

Randomizer

```
639715EB 0DE60AEF 835564FD 313764D1 - randomizer R_mtl [16 bytes]
```

Authentication Path

```
0000 - flags (must be 0 per [I-D.harvey-cfrg-mtl-mode]) [2 bytes]
49206EB2 770E7CB1 - series identifier SID [8 bytes]
00000002 - leaf index: i = 2 [4 bytes]
00000000 - rung left index: 0 [4 bytes]
00000007 - rung right index: 7 [4 bytes]
0003 - sibling hash count: 2 [2 bytes]
      Sibling node hash values
6C56A9A6 04D2DF1E 8D89F279 D9CFE91C - V[3:3] [16 bytes]
8ABE74C1 29655E09 AD8A5673 62A35736 - V[0:1] [16 bytes]
D3E35939 31BF1BE2 A8AB5D21 61198612 - V[4:7] [16 bytes]
```

Ladder

```
0000 - flags (must be 0 per [I-D.harvey-cfrg-mtl-mode]) [2 bytes]
49206EB2 770E7CB1 - series identifier SID [8 bytes]
0002 - rung count: 2 [4 bytes]
00000000 - rung left index: 0 [4 bytes]
00000007 - rung right index: 7 [4 bytes]
4C2A8070 45D0559F A52DC9F4 9813AD7A - rung hash V[0:7] [16 bytes]
00000008 - rung left index: 8 [4 bytes]
00000008 - rung right index: 8 [4 bytes]
8A80ADEF 0D345CB8 F1372CF7 A42424A0 - rung hash V[8:8] [16 bytes]
```

Signature on ladder

```
00001EB0 - length in bytes of underlying signature: 7856 [4 bytes]
DCDEE666 F6BD1500 CC ...  - underlying signature
```

The authentication path for this signature connects the leaf node hash value V[2:2] to the ladder rung V[0:7]. The sibling node hash values are therefore assumed to be V[3:3], V[0:1] and V[4:7]. (See Appendix A.3.1)

The rungs included in the ladder are V[0:7] and V[8:8].

The values produced by this step are:

```
i =  2
R_mtl[2] =  639715EB 0DE60AEF 835564FD 313764D1 [16 bytes]
SID =  49206EB2 770E7CB1 [8 bytes]
V[3:3] =  6C56A9A6 04D2DF1E 8D89F279 D9CFE91C [16 bytes]
V[0:1] =  8ABE74C1 29655E09 AD8A5673 62A35736 [16 bytes]
V[4:7] =  D3E35939 31BF1BE2 A8AB5D21 61198612 [16 bytes]
ladder =  00004920 6EB2770E 7CB10002 00000000 00000007 4C2A8070
   45D0559F A52DC9F4 9813AD7A 00000008 00000008 8A80ADEF 0D345CB8
   F1372CF7 A42424A0 [60 bytes]
```

V[0:7] = 4C2A8070 45D0559F A52DC9F4 9813AD7A [16 bytes]
V[8:8] = 8A80ADEF 0D345CB8 F1372CF7 A42424A0 [16 bytes]
underlying signature on ladder (abridged) = DCDEE666 F6BD1500 CC ...
[7856 bytes]

A.4.2. Verifying the Underlying Signature

The verifier verifies the underlying signature on the ladder following Section 9.2 of [I-D.harvey-cfrg-mtl-mode].

For SLH-DSA-MTL-SHA2-128s, the steps simplify to the following operation:

* Verify the underlying signature on the byte string toByte(129,1) || toByte(0,1) || ladder using SLH-DSA-SHA2-128s's internal verification operation.

The details of SLH-DSA-SHA2-128s are not included here.

For the example record, the values involved are:

* message input = 81000000 49206EB2 770E7CB1 00020000 00000000 00074C2A 807045D0 559FA52D C9F49813 AD7A0000 00080000 00088A80 ADEF0D34 5CB8F137 2CF7A424 24A0 [62 bytes]
* underlying signature (abridged) = DCDEE666 F6BD1500 CC ... [7856 bytes]
* public key input = 3DAC0F18 22AECA41 FA40EB5F 87A6FC1E 8259C38C 4C33EDD0 2AFB1382 84FFE7FF [32 bytes]

Once the signature on the ladder is verified, the rungs of the ladder can be used to verify authentication paths, e.g., as in Appendix A.3.4.

Note. The simplified operation given above for SLH-DSA-MTL-SHA2-128s can be derived from [I-D.harvey-cfrg-mtl-mode] as follows:

1. The message input to SLH-DHA-SHA-128s's internal verification operation is the ladder prepended with a domain separator of type MTL_LADDER_SEP = 129 and a context string length of 0, following Section 9.2 of [I-D.harvey-cfrg-mtl-mode]. (The context string is empty.)

A.4.3. Forming the Message, Computing the Leaf Node Hash Value and Checking the Authentication Path

These steps are the same as in Sections A.2.2, A.2.3 and A.2.4 for condensed signatures.

A.5. How the Example Signed Zone File was Generated

A.5.1. Generating the Signed Zone File

We started with the following unsigned zone file:

```
example.com. IN SOA ns.example.com. admin.example.com. 1719172701 (
                            7200 3600 1209600 3600 )
example.com. IN A 192.0.2.1
example.com. IN AAAA 2001:db8::1
example.com. IN MX 10 mail.example.net.
example.com. IN TXT "This zone is an example input for SLH-DSA-MTL
                            zone signing"
www.example.com. IN CNAME example.com.
example.com. IN NS ns1.example.net.
example.com. IN NS ns2.example.net.
```

The zone file includes seven RRsets. We added two NSEC3 records to provide proof of the non-existence of other RRtypes for example.com and of www.example.com, and of other domain names in the zone, bringing the number of RRsets to be signed to nine. As mentioned in Appendix A.1, we did not sign the DNSKEY RRset.

We generated a new SLH-DSA-MTL-SHA2-128s public key / private key pair. The public key is the one in Appendix A.1.

We decided to sign all nine non-DNSKEY RRsets in a single message series. We also decided to order the messages within the series according to the canonical order of the domain names per [RFC4034] (example.com followed by www.example.com) and within a given domain name, by the numeric values of the RRtypes:

* A (1)
* NS (2)
* SOA (6)
* MX (15)
* TXT (16)
* AAAA (28)
* NSEC3 (50)

Implementations may group and order the messages differently.

For the single message series, we generated the series identifier SID = 49206EB2 770E7CB1.

For each RRset message M[i], we then performed the following steps:

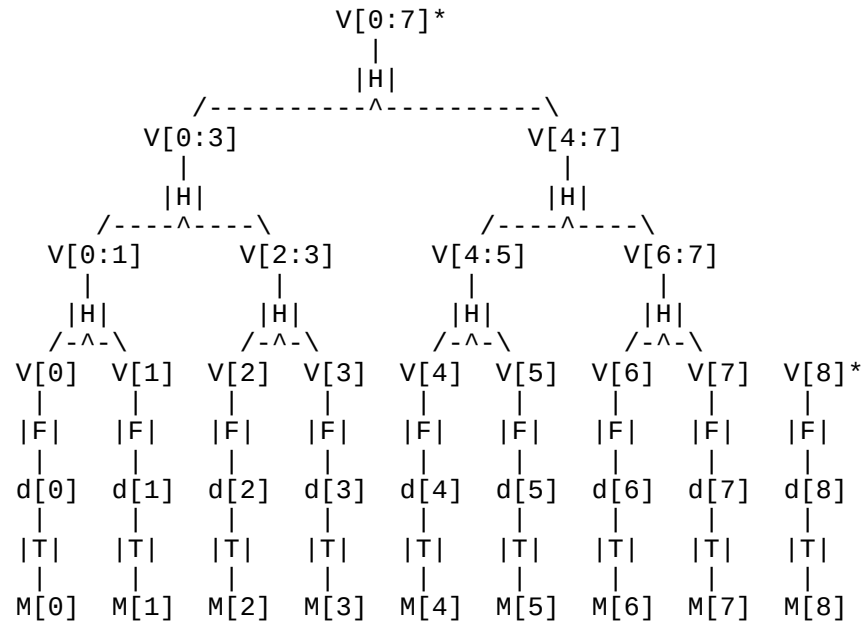
* We formed the messages M from the RRset and the RDATA portion of the anticipated RRSIG record excluding the Signature field following Appendix A.3.2.
* We computed a randomizer R_mtl[i] following Section 5.1 of [I-D.harvey-cfrg-mtl-mode]. (The details of this step are omitted for simplicity.)
* We computed a leaf node hash value V[i] from the message M[i], the randomizer R_mtl[i] and certain other inputs following Appendix A.3.3.

As we computed the leaf node hash values, we also computed internal node hash values in the Merkle node set following the same hashing steps as for checking authentication paths in Appendix A.3.4. We then formed a Merkle tree ladder from the internal node hash values following the binary rung strategy in [I-D.harvey-cfrg-mtl-mode] and signed the ladder with the SLH-DSA-MTL-SHA2-128s private key.

We next formed condensed signatures to be included in the RRSIG records associated with each of the messages being signed, other than the SOA record. We finally formed a full signature to be included in the RRSIG record associated with the SOA record.

A.5.2. Merkle Node Set Structure

The nine-message series that we signed produced a Merkle node set with the structure shown below. Following the binary rung strategy, the node set includes two binary trees: an eight-leaf tree with root hash value V[0:7] and a one-leaf tree with root hash value V[8:8]. In the diagram, an asterisk indicates that a node hash value is a rung in the Merkle tree ladder. The symbol T is shorthand for the H_msg_mtl function call. For simplicity, the randomizers and other inputs to the functions H, F and H_msg_mtl are not shown.



A.6. Full Signature

The full signature byte string is: 01639715 EB0DE60A EF835564
FD313764 D1000049 206EB277 0E7CB100 00000200 00000000 00000700
036C56A9 A604D2DF 1E8D89F2 79D9CFE9 1C8ABE74 C129655E 09AD8A56
7362A357 36D3E359 3931BF1B E2A8AB5D 21611986 12000049 206EB277
0E7CB100 02000000 00000000 074C2A80 7045D055 9FA52DC9 F49813AD
7A000000 08000000 088A80AD EF0D345C B8F1372C F7A42424 A000001E
B0DCDEE6 66F6BD15 00CC8B96 E8A56A67 C13C4325 08C3C29D FFD98566
37C4B60D 6874508C 078363C1 48CCE173 179DCCDB 06A08E5E 2FC65C12
0D39141F 7ABFAEF0 7CEFA113 595CA3EA 1B696C57 7984E4E4 25A35556
09C9F105 F11246E1 A2DD2286 006D0267 B13C72CB 765400E0 09F40283
211D7576 632836A7 F0240FE8 33FCC73D B067E874 DE4D53BE 3B74AE9B
2AFD2820 B60F0BB1 0373C958 7A627B38 9E7F9CE0 2241709C 2BE68B7A
4FC70EA4 2218743C 0023BADF 2264709F 41428E0E 1D7AD877 8330E058
899E7A3F 2415E5BF 811796EB 97A71BAA 8D21EAAD 6EE15C7D B79E2145
B0DE85BC 97B71121 E2F86C90 9A445950 651C973C E18B25CF 07D779BA
2A0C43C3 51A1B24C 3E6FAAD8 A5599D9E 12260856 F9C7F132 2F9BD297
1A5CF48C 58F87AF9 028C1CCF EBCBDA83 5DCB04C5 B794CCED 43D40963
EBD56F47 35D9B36F FFE2B144 85C2EA40 020FE108 455AF337 8FED7569
EF6278F4 B048C391 5F17DA4E EF01E77B D585007E F34C14D5 BE41ADA1
24AAAAB1 02D6662E 05D45915 8B81FBBD 0D07F2E4 9A32CAF6 4E6EB3D3
318BFBAF 3576547D B258D910 73074D28 1E3F1E9F 08F3C498 09691724
0CC6F09A D2BEA46F 67491B27 6A03F40D 600877CA 2709E914 94004FCD
8E500B3B C441456D 3D6CC46C 31B91968 E255BE46 21C6C75A BE21FA7A
99897481 8F0C57D4 A7B385ED 10A9559C CC88AE68 25899BCE 15AF960D

```
   37B3884A 1351FD87 92912F2C C3EBC824 B06BEAF2 41B5972F 05C5EC6B
   5C346D01 D935D75E 887A0590 57B85893 BA879ED2 1B135A47 048DB9BC
   7C10CC1C D596C675 C3C2DC29 2BF3A0CE A48DAA7F 6F23D0E4 1405FDE0
   F04B4BA8 F7E80AD9 435306CF 2B8AE120 D638B29C C3146515 CFE4855E
   91F604C4 C6DE4C41 54DEB6F7 2CCDCDDE 5E247B71 795ADB25 8CCEE545
   A9AC0013 B2C68822 3665DB45 A6328361 B19B6B22 27982D95 8FE46979
   A058665A DDBE64A8 8D7C4E12 3BCA8792 A89DC673 6D6AE4A9 81909EE3
   DBD5482B 44FD9552 48A220CD 629E5E42 1934C4D5 205BFE4A FB9ADB4D
   F330F08D F9E7BB35 BE4E782D 5F5DEAA4 15184240 3B26843E EE3C2B8F
   B2814E6F D9EA0088 6CB8E233 4E159D47 D8B77E4A DD315A21 3A828DAE
   526847E9 9C414BD9 BA9E3558 F11DB57B 9228BEBB 7560EEF4 50209260
   A3611C5C 347823D9 3B8280DE 242035AC 89C7FEAB 49123849 75CD41D5
   312B7FB3 C76CF446 989D32FF D2E23770 92D0920B 62BD8E4B 9593047C
   28667D81 A95043F1 09AE89D9 3ED6D676 63D5C337 BD41B176 2E5F1AA8
   6B2D4521 2C46731C 0420484A BB464C1C FD397C94 C791668D 715ADBCE
   AD488A4D ABC1FF2B 15188214 9B68613D F0ADA730 8C1C2F97 330FD4AE
   022110A7 E3197368 52AF0C64 FDFAB881 D19E7552 97A970A0 ECB899C9
   9BB30B9A 66D94951 F57CEE19 DD6229BB 1C195369 A48BABE6 0CD51B37
   8CDA53A7 4C5D5A74 1C244058 A888C0A0 6EE61DC2 CA0AC2B3 34B8F74C
   C3514EED A8E7614E E3265BBD 36B76C97 FE6A62C4 0A20E494 5AE7C9A2
   3F32D09D 05EA4A00 9F7CC2D7 CF083EFB D1FACCC7 77F5E29F 528DCA42
   0D4A5FB1 F3EEA9DD 436C6B17 21BF6901 3338FFD2 4921E433 6618595E
   3C85341A EEE90D14 F7752481 78785507 1BEEE94E 045D3584 E54434BD
   62D75796 0383148D F549233F 24CEE6B8 14722A28 BA377C9F FB0F212A
   F6EED10D C1E433A6 E04CB874 DCA5D93B B93AA536 FBBA8A10 9DDD3BFA
   B3C42EAB 4C4637E8 70E3430A DBAAC146 7FA39DA9 3416EFB7 C968CAEC
   BBF2BD68 6E8125CB E4DCB994 495A659B A3C4D705 2902FFEF 73AA11E3
   F2FD8CDF BEDD94E6 3CF14DC7 977F5632 00F74683 A15FAF81 3C95415C
   FF6D7A44 FD133E46 7B828F3D E9DEDD7A 6F8646F7 C16EF195 F6F16766
   B3272897 90821964 21CA7759 3340E37C 9E89F96B 779B22CE 4BB6DAA0
   94838285 AD0A82CF F5A2C042 6009E109 DA186D6F BE50FCC4 7C252209
   341E693F 66AC2B21 9CBBEFE9 9979E705 8689137E A66B36C1 C549AAF7
   E4294BCB 1EAE94C3 AA227A2F 5C3D1441 9D62C9DF 952C9E74 D7FB50CC
   35C59597 B59D7DA6 BE0120F3 640BEC53 BF5599A0 AC683DB9 BF33E3FC
   922B802F BB5F96EA DB85ECC6 73D4D9C7 A917BF3B FBEBC20F 98A84F9F
   56A35D71 9E0EA3F0 98AF596D 5C0E6497 23E25663 4012E8C9 C1CA4DA5
   F9D0D574 888DD7E0 41803553 8E2F8CEB 23B618C9 DBFBB80C 972C91DC
   2C04F1F3 830CA090 25CFD16E 588351A3 1A04DBC9 6FFD2736 F699102F
   17D4D1A1 D4C38E9A 491BD133 D7BD3741 54131B2F A8E9FC12 A0ED10F9
   A35FF3CC D062484F 4142F2F4 5B73B7A6 202423DE 0EBAFD36 9779C68B
   4F52D6AF 5A21E233 A508149E E0473DE4 E6262DBC 11D9B13A 06865A5F
   A5E133A0 2FCF35F2 BC85B4C5 9C394592 4B323679 694F20D9 586BB08E
   5890D315 FBBA452C 40EB0529 7AEB3D4A C3CACE6A CBF8CDBC 810A272B
   61A6A370 5FB32798 354377CF 318494DD CB6C62FF 748BA102 2802EEC7
   D27DCBD2 BF89F843 EF6308A6 66B2BB74 E7420C2F C92D8821 4C5CD699
   B2C4A529 74808206 829BEAC9 BC1A1E95 F6E70AA3 11BAA033 B272E146
   34E5635E 0D7C554D 1EEFFE70 16EFDDDC 91D8CDB0 442B1BA1 A4CF619A
   4DFB28B7 FA433A00 1BD1C697 6A12E287 20E23F20 5D0F0DE8 4322C609
```

```
   B22CAF84 0A915D04 CD54E3EF BF78AFDE 5DDF7F8A 152C5368 04E7CC15
   B6EDF024 6D6D3049 E2D5192F EA00E0B3 32ACF7B3 35A429A7 F6080188
   D11B308D 9B9225F9 F405D79D A67C6869 A4264EDC DEE0EA32 7BA4A45A
   F4DCBDA9 39BB2159 42A46073 8BEC87AC 1865AFBA 2AC02966 13646311
   4F347464 1E7A651F 4C9B518A 06FDF256 52149EC5 7BF08A3D FE2E9485
   E3767180 EF516987 5B46E09A 389C37DD 6C69955A CAED3F52 22E777FD
   E7AF9C6E 16CD99AD F1B0A412 2577F8AC CAA6B071 111E3F7A CD79C7AE
   68816C78 6D6EFC68 F5087486 A59A8D56 9A90E7AA 459F1ABC F19BDD48
   4669442E 136FA4E1 0E501AC0 74B06658 1CD63098 E5A3D5C0 8671BBA2
   0789F997 4BC03698 8ACCF9DB D8BBF1C1 174E2385 65CF888E 4A4615E5
   389DB438 3A1C8058 E84A8DC5 9CC64E67 FF25FD2C E3693089 457008F2
   EC44A0B7 B01E6375 6860E5B4 6BAD510D 16CF4BAA 45F1ECC2 0082417E
   83EC7100 35EA1272 4C31872C 5FC905BE C1AFA59E 388AECE2 7BBA86F3
   19EF476A F9B0CDF1 D66D9A2B 5DDEE0BC A2C89DE3 F1DF1200 CA555EFB
   CE4A8243 359260CC 8F07E64D 2C55AFAD 893A56D5 91AD733E 3D064911
   D296A439 98542A70 B4EB7451 7A48C19D ED48D9AA DF2112CD AB6E2418
   FC770B00 4F7BFBF7 5CB0EC5C DF4A96CD ADD784B2 F570AA4E A89EC31F
   7FB85B50 20502E73 8D83CF5E AC3D4A8E 1F00EB37 AD7C137E D1C515C0
   7414D130 A3E38C05 576E0A41 C822E097 333FE4B5 5846B298 EB648D9A
   9BD04EAF A8F2EC9F 1F23E45D 8642A62E 2C32CF04 8CD3458A 964FE023
   5D25A075 B13B0A45 4297A490 541EF4A9 955ABD69 5F05C8AD 13B428AF
   DC3A6769 8EBFF402 BB01A6F1 642AC919 9785FFC0 9C8FA99B 583A0A70
   13417434 17CB3F0F E4CAF114 26C2B770 4347819B 44718451 6BDE0569
   3EEC61C7 7ACC8FA2 8C601062 3E9A2C41 F395EDE7 478A8A80 E0F3ECD4
   91EFC6BF E3DAD79F 2A6245D3 5DB3E073 B23A36D6 951DB1C3 E30D83C2
   FE3AC978 F0B682D0 F72A55C7 FD6CBA75 22B27F59 D213D5E9 D5DF6808
   C473BFE4 E7F4B01B 0E75D4F1 0A44F362 9C50F1AB FF5E6D2B F8FB600D
   6C7F8222 D646921D 024179BB DE98863E 7743505E 45D1C4AF 6B52ED97
   A379BCCD 22D19BD6 11823196 DF7DAAD2 BDCADCCD DA750574 6483CD56
   CCB60118 90E485D5 F3482CD2 5897D973 490E3B5C 912C31F0 7D8093FF
   8A1E9EDF 18E6FF78 8A8CA096 AEB4F4BE 27B43767 7A23058D FE7F5AC6
   47B44D4B C3A35324 FB5B1028 9F6911F9 531BD0CD 98278997 93B647A5
   7BCBB7F9 32D88FBF E6073778 3FBA37B9 7C4E289D CE36B5A8 F2324819
   016B0A6C E91DB85F 26BB2D67 8AE2254B 18769DDE 158E90BB AB25A055
   327F0D7F 7326CE79 07F897BF 2C6FE1A6 79E03B89 334943C4 EF3334B3
   E5C3270B FBB9FBB3 7C5EAD90 0D3C7FA3 61543908 65A7C54D 1A17376F
   CC019B91 94C46BB8 DF13B466 19F970DA 48B7DC74 BAAE412F F93C8432
   C34AE108 0D549DAD 2EB70BBB 60E48552 1A2541EF A6891DBC 6402D1CB
   1B7C192A 5FBB4CB7 D34EB830 ACF707FE 5BCCD1FA 915AA561 80757555
   B7A3CFB2 A0A5DC16 2C19D9A0 1BA1AEA8 DDDFED09 DFECCBB7 9F05A45A
   CC5F5A64 F0F70590 9D16B72A BEE727F0 D25F8675 155AC2E0 DE3AAEA0
   83A860CC FA6312F6 88A5C6B9 0F6BAA8D 542D3810 E25D5355 7A566A30
   D08DB9B0 918BDCD0 6EF73B3D 40DD1B0C 405FB324 BCBF9A3E 865B0BD5
   4CAD2CF0 B12D9900 356CC65F FD5D2A5D 5773F233 235FCC6A 18835407
   AFD2E2AB 46C25653 CC59CF8B 29BAD15F 5DBCC294 BD319B9A 4FDAD597
   203EEC6E 7A78BDDA D94EB457 BDE39E7F 973FFC58 B5E41657 3A974478
   D9F76402 BB888A83 293AFF15 6B0AE2FB CD212F23 0CE4F020 A78E89C0
   FA7A79A0 033DD06F C6792668 4E59BEE7 3C4517AA 8B70AEB4 C5C2A5CE
```

```
   5AEB63BD 2324D51C 41D4D072 DE69C213 43AA2A2A 9A6E49D6 29123D94
   84F870F7 F545F702 45C0D7DF A0DB1448 56CFB7F8 5157788E 2088A620
   07D7FFDE 39EF97BB 351B168F 3C5A9815 E03F80FE 2CB444C4 CDF93F38
   79610DA4 F618E931 077C11E3 559BEDD0 3673010F 4BFFD9A0 A271282B
   54BCD69C 33961AB8 0F99AEDF C8574FA1 04937833 53947E68 7C3BF500
   DE1C2F0D 6AFCF913 55876125 DC76669D AD688442 C81BD2BB BFCE55E6
   6BC66DD1 99DCDB53 D98DC931 48C107EA 9188996F 19B3D6D5 F48011C0
   69BA18BA DD824FF9 C1090260 23296CE0 A99CA0B2 F7BFA276 5FCA1443
   DB988BB7 8243E968 1399E840 590CF0DB 712D20F8 15E55273 98248645
   545F5975 821D9082 48F3D991 7676ACCC 041A0E5F 38072C85 D57C284A
   09B72812 98EE094E A1865926 886EFC84 C2B0156B 41D17082 E6BA61E4
   D0F836E4 F5FB4153 AADA3862 40B62710 65B90DE0 C823165F 4EFF12EB
   C0748878 D944EBA7 56766B6C 1B5E6375 BD2A0BD3 72A700BF 2D73F653
   2DD48D74 80ABCE66 18CCDE47 469DD21F F09BA448 B03EAEF0 171FA440
   AC3C10F5 3C6E1D0B 05B2EDD3 55951391 33DEA6C6 69ACC262 AF421A05
   20784221 CF2F2430 D415CD7F 2C2B5E62 64C018AD 37E11179 08AC97E8
   BD0C2024 E5EFDEE0 21606DA2 D3E01E11 DA2BCD9F C58BC76E 9340F27E
   5496D62D 21C8E3F9 D772A350 E55BBBD6 A1AC33B1 8ACF0F74 299911DA
   76433F5D B0896E8C 0D586AEB 3D073A05 7F6B21F8 39130267 7B988C08
   71FEA48C BBAC8FA0 668F2581 4014075B DE16C33B 594958B9 F7FE917F
   777E9EE6 7ABFC7F6 7EB6726C 88D494FE F1B8B7CB 4C9E7F24 8F477936
   3E5E766E A85040FA F21E3B4C BCF4B713 97945EF0 BABFCCAC 4D055DA0
   54143119 3AF46B18 AD61B929 BE970F8F 34E9C365 BB1A719B 53D5295A
   DF294847 96FCC450 E581516E FC9618B9 9534D28D E43867C4 B34C2F46
   DC2828F5 83FA3748 2E5BBB08 FEAC1F46 B14DD9B5 F860A543 5532AFD1
   CCEA717B 76D75784 095FD77B A160E339 7AFBA079 6414EC32 53E7EF40
   9FB67387 7FEE7CBA 47B94462 5F2C8FCA 1E46793D 13007163 E7254E3A
   7FDB332A F00BDD17 DBC4305C 76EBB1DB E436A1D0 9D3E0A8E FAEEF2A5
   2A530438 AC3D897A 52508471 0676E847 449612A1 2102BD6B C5DE7308
   4E6CFCDD D4FBE91C DAE0AC82 A864DF98 D5D5DF95 22F1B20A 314C948C
   189BD1A3 3252C9EE 71A7123C 38A7E950 E5EDB15E FC3A7052 2DE21B44
   9207DD2A B5618C90 1E79FF1A 39719ED2 2BD0FDD8 B126176B 11D28EA6
   8CB7114A 6C9647A3 5D385605 D6A7F866 10803880 A8317B1E 6F6212B2
   4E3EED03 9D11D42C 33FF4E2A 6EE36127 D35ED2BF 13DC2871 C2ED51F2
   8D8EB885 E0811825 61E36D9A B908BE82 931C89A8 28F400FF C64DB7E5
   7EC47931 C05E2099 A71B99E3 4BA6B834 20031E9A 1AEAECC6 F223F08F
   9EFC91AF F8242A97 5B475191 19FF6A0E B8F97DC6 EFA5BC06 DA0ACD6A
   0472CD79 648ACA91 7D3792DE 2AEF08DE 33546BA3 97916102 011472E4
   05C4FC23 7A3C5755 FD9F7B99 5F13280D FDBA02AC B07C1F56 60E914DD
   82BEB2D1 D1F4AA2A 39DAE80E 9CD83BB4 D9D75201 3DAD14A6 32EAB8FF
   4C4B3C75 EF10C366 BD220C53 09CBB530 473036D0 8C9132CD 681365A3
   CF504226 06E6A470 3355555E 1A534F20 3CD84E8A B695E0F2 4ECCE754
   19EA9804 9C79CCA5 841FD95D 40F8AF44 26637318 65886EF8 B6F306EC
   AEA990A0 A605ADF8 CBB7D4B4 3C845601 A8E6DDAF 22AB2430 248FF24B
   6FD06D61 EAE1950A F970D01C 1C368E70 C78357DD F6F4163E D3F6FC97
   E58445C5 0828F650 DEA74894 A2F22764 603980AC EF1858CA D6A2671A
   112DB70A 99F0859F 211077EB 0CFEEA31 A20EBF06 FD33DC5E 4C0585DB
   C496B177 137CC30A 9A7AC5C5 7A250A74 6FE40F75 BBE2EDF9 7675F792
```

```
   C1298166 952AC141 5CFAFCFA ACB9F0DE 3DA0433A 717A79D0 49063787
   7B04D725 B27F8742 9CA9E3EB 9FD4D0C9 CD39FDB3 979E7EEC FA5F8FCD
   D244F3D5 E0C17D97 76C8E7D7 77B4ECA5 585D9173 50FED8DE F808F4D5
   A4B75978 93784DA7 D7448AC8 1083A7BD C7D12795 F3EF735B 666FB18D
   A3AEEA73 1F7ECBC2 9672B11C 08EE8F05 36D3ED19 383F95FB C967701F
   E3836D50 BF5CA299 C912A831 754FA7C5 AB5862CB 3D56AD0A 3FDB1D36
   2E74EF21 279BA17E 1893B4CB 0BE77E07 36DDF543 B3A9B645 FE54182A
   D699A60F 1BFA6C7B 558B73F4 512254D0 07DC8523 F19D8588 95CD1D85
   F65588F8 E957A020 1854BB35 DFCBF5CE 35D2B12D 834D6FF0 4EB3898B
   BE4423AA 73C05FD9 94F6F686 6F6EAC93 B3CBE4BE 5184CBB7 7CBCCE43
   2D19F398 BB9E131D 5B413416 EE0BEC76 4B5D1C12 55E73971 E1407029
   C5B79F7A 1DF7F568 C677A708 B79076A7 4F0EE2D1 D9B41E72 1343FFE2
   1FB3A194 8C42950F F7D54249 5D008BE5 EA218BE8 6282F771 167649AB
   8863D8E9 C79BCF9D 68E56D70 CD9F9195 6F42E0E3 22F83AE9 9958C4F5
   F0BCB0CA 88510ABD A8001D73 2E38D608 E0699376 AA8CE93D A13E35F1
   7958F787 BA234459 2B908A21 0926E109 05ED7625 131B8B54 22CDD1DD
   694267BF 4CCE00AB 160C323A C00D7FF7 9EBC9896 301FCD8E C327892A
   CF1AC575 3041E0B0 8F0D45EB E0514DB1 B8C413BE 6939600F A7DAF9C6
   CA366CEE A7052B12 0403EF97 C232036C 5B481347 28600824 91C73293
   476A6465 71BE2ABB 45090323 7D479288 9B171949 BC06E34C DB8D9415
   7AF13FEF 717D16F6 1107BB7C EF7AB81D 87808FD6 893A767B 9FFBE55A
   24D41DAA E1461808 C236DEFA CD3AF2C6 255833C0 AB831E71 B6E181EB
   B339F362 6581C65A 51E9A59E 4FF78205 32558309 B77B6AF1 E490FC96
   2D44B68C DAF1B375 4696081C 79FFB921 1C69B532 03F54A17 5DCE01E6
   DD6EF8B9 4F6F6496 4B26C4B4 381A1141 7D99C8A1 A97C21E1 5CD4A5EA
   A1E8F7F2 7DA5653D 71A47840 E38E131D 6FF375D1 7F7E562A B8A17B86
   EE884F72 E388D867 51A866F3 3F640F5B 23F0F933 6FCB3889 3966540A
   A34C9C2A 857BC10E 8FEBE0C8 2B053916 9D154241 D776601C DCBFCD3B
   20925131 AEA470AE 84179356 2DEAA5E5 9DC4564E C971FAC0 8CF4FE57
   20724602 DFFA90FD 61DF2906 968202A6 8B11459E D7037D47 60AA252D
   70489E35 8E374C35 4DD56459 2456D9E8 7AAE24EF F0E241AB E886B069
   3B4C9FB8 298A53B1 15752D21 7CE74962 92387297 C6A3984C 666D3B89
   22CDEBA0 328D4C3E 7659D2A6 4D5874AA EA6793C6 F5A36FFD E2153586
   283E78AF 014231FA 028F37B2 756CC252 391E7760 BA94A0FA 23DA8A74
   E1E3AE33 5477D10B FD390541 65C15C63 BE924473 7124E8A0 6ABD1794
   9BBBB51A B34D5CA7 FD19523E 72766460 C1E2112D BD1AC027 5C0771F2
   ED4656AC 84F00366 094AB7E0 B419B851 B5604D7B E313B12F 7DE0B817
   9140FE8F B7DB2698 66D47098 5ED0DB56 81070CE1 F203668F 44294D35
   A7F70C3D CB997AC6 4DFEA7AD 1961633B F0C104DF B7C3038F E987D6CA
   F7A1FCC4 D4B12634 BA6871D5 B1A179BD CB89F8DF 0DFCD732 7A0BE545
   075764E6 2B938D80 A2583C98 CB59CD60 8C8CA0DA FE576202 28E5A7DA
   D4DEB304 ABFA4EF5 DBC002BA C334DBDA 24710B15 0E464BD1 385095C9
   F6A74C8C F38400A9 1E2DE0C2 0A92ABB1 A81BD279 DCB70F9A 90540FA3
   00027F12 E1C04935 B2DCC120 FD43F0BC 2A867B9D FEA782CB 2216C923
   FD016048 353336B7 82C713F2 4F8D264E 1C81A3ED F04E1A01 848D628E
   A7D28185 EF477A0C B242482F B8827332 8B84829D 1594F69F 9E4AEC07
   2BBCBE8C ED4435D4 98476649 85B0C0DF 4E6AEFBE E9DA4FE8 75CFEBB4
   1A5E3B66 BC92B3A8 895F6AAD 6609B979 DA4D580D 12E4DD6C 4012DCAC
```

```
   7E544FEB DCAAA47F ADC5872A 0BC915F0 DB231E88 B1CA538F 12C1CF50
   565DC5D3 DFEEF30C 0B8DD90F BF1492F9 1DD05E5F 4C7F38B5 033CB733
   F46ED6E2 816A3978 87DA21F8 1C0B3B29 7336D5B0 A9B6D64D 23294AD6
   81923398 862AAF8E 8B142055 DE4F562C 9BD6DBAF 911C377B B2D54F4B
   61374C51 19EF24B2 08D4DF31 B5DD1719 BA878109 3DCE1EBB D0843BF5
   B8CE8C4D B9D3CCFD 65A92CE9 A3AC325B 42562D84 92A1E660 A0FBD98E
   0B68CF7D 112F03B9 23702515 486D1840 F95D74A6 558A84A8 ACB82CF2
   F5B4160C 93CD18BF 7B6D203B 9E924DC6 683EE5B7 241C7291 2E39944C
   BABE1D4E 56732D69 F2D5162E 4F4C40E0 8AE95468 D469BB4C 09EE3E4E
   4BD716B2 ADA0CE33 EDB5B5FB CC835682 4FAA59D5 FC591269 E60C7238
   FFDBDEDF BF65DE75 34946FC9 F302D4C3 4F919637 BFA486C9 3CFB8CE3
   5ADE3B08 B75134E3 26D2E942 3B0EAD4D 5BEF77D8 43D2AC7D 413F69AD
   3D8AA993 7D177AB3 78C0FBF3 D752A5A6 0967321E 9E9CE0E4 8DBF5AF8
   BC4BB894 CC69899C 3E081C7B 84FC615F 26FF0411 A6A6157A C5DFB5E5
   CB9F922E 3EAC98CA D3EFDE52 EE25B58C 572744E8 61C810DF 2442B1EE
   0F7F2494 98C40446 9E1E365F 9663796C BA41D144 A33DAECA 2447AF72
   43507822 E07B32A8 89B7D01D 483CF7C2 F601980F E0C4A286 ECA59BCF
   800610E6 A132C5E2 DC6EE5F3 37623C27 C74FE036 F2028282 8DD2D81D
   CD430EF7 494F45DC 21057D27 A99D313D A856931F 2ED78331 226A98A2
   AD969594 FBDD265A 4BBA49B0 7F6A6978 0AAA9CA1 429DFA0D 068E16FC
   7FC4858D 14062A28 223D0E52 D69A2A78 AE4AEF9E B26D3B4D E6F91764
   57AFF4A1 FBAB5F85 D212A141 5EEDB1ED FB7443C7 DF3CE903 0303D6A0
   85ECD438 9EF54C0F 3B9AB0F9 5651F33D C2F70301 488230CE 0B4BF17B
   36E43857 53DEEFE5 2429EC88 9DE7B900 9E12660B 9571FBB4 87BA58B2
   E116945F 6C8CDE81 16F8EB98 07D1645B CC8FE705 8FF4E32D A34633DC
   62CE2EE3 33897100 1B50BCDF 0BEBDD07 44F1B39C 8CA851C7 AF3762B7
   9E330748 E139A149 65A7ED81 158E6E0F 7BB43FD0 5DB15870 E8F2C587
   A82BE7D2 0E33BEB1 18D73AED F877629C 5581C64E 0B28EBDF 0CDA9869
   45127D3B 5E417939 2CD4CB4E 6E99D779 470AFF50 4BDAC047 CD25442D
   137C650A 32585460 A9ADC20C A0272097 87B23FC8 6CD16CA0 FAAB635F
   FA432423 49BD70B4 589B69AC 842BA054 37E761C0 BC917E17 91B28D66
   59B1ECF3 1567BA4C 107B9267 67C25FC9 3D3B794D CBAD20B2 BD221EFF
   E2D30A63 62B61582 BCDD81DA 3B3CDC16 B2D9BC57 2B693C3F D1E92149
   491EF851 F0AA920E 65713ED7 F1FF2194 81627B88 28E7CD2A 07CDB82A
   05135A20 239B8A37 7E85F36E 38258277 0DDD0503 3E9270D0 5F33AFC9
   B3867861 364E7250 BF7010C2 A9CDB27C CFC1509C A01D4721 DDF8998B
   8C218ABA 63742292 02C94A38 139940B8 E2107517 1D042DC5 51445CB3
   95773936 F7D0DFBC F2808D8B C3731364 D56DEE9E 3C7DA05E 6C0C2CF4
   334147CC 28A74F51 C9C6A9AA 0B7876AB 95B0ED1D E1C82619 ED76B2F9
   B5789DE0 4F3217B1 DD1CD4D2 651B2B1E 6C113033 871A3997 441B7A4F
   DF6528FC BE078E23 1A5AE209 EB2991CE 4DEFCA9A 455C336C 96C2A373
   07957E10 7E6FE375 A07245B9 506D5A19 D602E70F 1ACD4790 28717656
   11A2BAC3 462AA6F5 CD6028F1 1F5009AF C39B32B6 8F15D3FE FC73696F
   198B4DDC 1447CEEF 38ED09CB 25653CFE BDC0B661 8028654E 10F07880
   4A172C27 5F9B4D7D 219AD49C C4CC776E 09FB5F2E 87773972 49780325
   FB016652 426F0031 AF5DD9A6 83745E6D 9D4D51D7 50B0AB7B BD792AD8
   A20BA44D C16FBD13 E78DC937 BCE322EA 4A3CDD4F B08EE150 A05A881D
   70E3E635 AECD2E87 33F817B0 85F8BC5B 620C125D DE8DDAFE 7DA96455
```

```
39EE0A89 648A2D6B B6785A63 77EF328F 8EB84E4F 134A7817 A05B61BF
BEBE8D45 C0871334 988C5267 9AF91200 F0217DE3 94362587 0775A4BB
EC8C92D2 2FD94715 964BE5F2 E5584121 45B6328C 5B2D45C9 771476E0
2603460D 8FF5CC49 C8BAB05D 7EBDE9E4 974C05B8 3B31A424 283E2E7A
BF28C6F4 23B1AC40 D47725E0 8C3E916D 4A905A5C E8FCB86F 86B9CE7F
92C10232 8B513A92 D913E803 21F32AFA 653A7F2C F4DC09E8 200B5319
06B980B4 006FC7C2 1C483C81 B13717B0 34EFACC4 8C4A2E1D 216211D5
B65E2956 5831AA08 0D6870A7 8B4879B6 A4211147 CE7473DF 8C9BBE94
6BD857CE 0ED469AB 4E9CCA48 760850EF 365DCB53 B7C71373 84E74A87
80B27DF4 6EE5EE70 BCD62E4F 0B40F4E2 B2C2D972 EEDBC748 D63DFCE5
711B62CA 128A4911 DECCB486 264221A4 C7DE8162 8762C862 93124F65
770D98D4 6E8B4014 860A9A78 3739B6E7 B5308F28 BA78FEBA 586048CB
EB8C21A8 6F2D64E8 D0ED9F7A 42C73308 9E953252 BDA2AB38 8A41623F
3DD9D2B2 21BAAD73 6269D9AF BEA8F406 EC037013 D3F56C40 3B29A28E
965FF032 EEF2D88C 0AFBE3BC 58BDB500 871AE982 6F441926 C95CF4B5
6E01CC09 23786E6B E5C5381C A37690D7 660E8C35 89EC4DCE 5B289324
92C9E372 5010C3A6 1F78D462 0E2D1278 460DF3B1 99605AFF C80A4760
EE261951 D0A19D67 B9B598DE 71C61AA5 5D37A494 07264B83 42B41463
2BB8A988 8850DDEE D26041D8 D70D13D9 75ED4248 FEA01787 4FEE050A
32FD6F6C A298DBEB BD1953BC F7F2002E 4ABEC072 20B14A85 04C577BA
9B54864E F458A0C9 78D7329C 93129476 F383C1DD 196B0B37 74A2B557
DC85F5AB 8A180107 2938ADBB A1AD8B18 7DF6578D 1FA05478 B03E07B9
01C2AFC0 11C7B0AF 908E0B4C CA1E88DA 4E
```

The Base64 encoding of the full signature is:

```
AWOXFesN5grvg1Vk/TE3ZNE
AAEkgbrJ3DnyxAAAAAgAAAAAAAAHAANsVqmmBNLfHo2J8nnZz+kcir50wSllXgmtilZz
YqNXNtPjWTkxvxviqKtdIWEZhhIAAEkgbrJ3DnyxAAIAAAAAAAAAB0wqgHBF0FWfpS3J9
JgTrXoAAAAIAAAACIqAre8NNFy48Tcs96QkJKAAAB6w3N7mZva9FQDMi5bopWpnwTxDJQ
jDwp3/2YVmN8S2DWh0UIwHg2PBSMzhcxedzNsGoI5eL8ZcEg05FB96v67wfO+hE1lco+o
baWxXeYTk5CWjVVYJyfEF8RJG4aLdIoYAbQJnsTxyy3ZUAOAJ9AKDIR11dmMoNqfwJA/o
M/zHPbBn6HTeTVO+O3Sumyr9KCC2DwuxA3PJWHpiezief5zgIkFwnCvmi3pPxw6kIhh0P
AAjut8iZHCfQUKODh162HeDMOBYiZ56PyQV5b+BF5brl6cbqo0h6q1u4Vx9t54hRbDehb
yXtxEh4vhskJpEWVBlHJc84YslzwfXeboqDEPDUaGyTD5vqtilWZ2eEiYIVvnH8TIvm9K
XGlz0jFj4evkCjBzP68vag13LBMW3lMztQ9QJY+vVb0c12bNv/+KxRIXC6kACD+EIRVrz
N4/tdWnvYnj0sEjDkV8X2k7vAed71YUAfvNMFNW+Qa2hJKqqsQLWZi4F1FkVi4H7vQ0H8
uSaMsr2Tm6z0zGL+681dlR9sljZEHMHTSgePx6fCPPEmAlpFyQMxvCa0r6kb2dJGydqA/
QNYAh3yicJ6RSUAE/NjlALO8RBRW09bMRsMbkZaOJVvkYhxsdaviH6epmJdIGPDFfUp7O
F7RCpVZzMiK5oJYmbzhWvlg03s4hKE1H9h5KRLyzD68gksGvq8kG1ly8FxexrXDRtAdk1
116IegWQV7hYk7qHntIbE1pHBI25vHwQzBzVlsZ1w8LcKSvzoM6kjap/byPQ5BQF/eDwS
0uo9+gK2UNTBs8riuEg1jiynMMUZRXP5IVekfYExMbeTEFU3rb3LM3N3l4ke3F5WtsljM
7lRamsABOyxogiNmXbRaYyg2Gxm2siJ5gtlY/kaXmgWGZa3b5kqI18ThI7yoeSqJ3Gc21
q5KmBkJ7j29VIK0T9lVJIoiDNYp5eQhk0xNUgW/5K+5rbTfMw8I3557s1vk54LV9d6qQV
GEJAOyaEPu48K4+ygU5v2eoAiGy44jNOFZ1H2Ld+St0xWiE6go2uUmhH6ZxBS9m6njVY8
R21e5Iovrt1YO70UCCSYKNhHFw0eCPZO4KA3iQgNayJx/6rSRI4SXXNQdUxK3+zx2z0Rp
idMv/S4jdwktCSC2K9jkuVkwR8KGZ9galQQ/EJronZPtbWdmPVwze9QbF2Ll8aqGstRSE
sRnMcBCBISrtGTBz9OXyUx5FmjXFa286tSIpNq8H/KxUYghSbaGE98K2nMIwcL5czD9Su
AiEQp+MZc2hSrwxk/fq4gdGedVKXqXCg7LiZyZuzC5pm2UlR9XzuGd1iKbscGVNppIur5
gzVGzeM2lOnTF1adBwkQFioiMCgbuYdwsoKwrM0uPdMw1FO7ajnYU7jJlu9Nrdsl/5qYs
```

QKIOSUWufJoj8y0J0F6koAn3zC188IPvvR+szHd/Xin1KNykINSl+x8+6p3UNsaxchv2k
BMzj/0kkh5DNmGFlePIU0Gu7pDRT3dSSBeHhVBxvu6U4EXTWE5UQ0vWLXV5YDgxSN9Ukj
PyTO5rgUcioojd8n/sPISr27tENweQzpuBMuHTcpdk7uTqlNvu6ihCd3Tv6s8Quq0xGN
+hw40MK26rBRn+jnak0Fu+3yWjK7LvyvWhugSXL5Ny5lElaZZujxNcFKQL/73OqEePy/Y
zfvt2U5jzxTceXf1YyAPdGg6Ffr4E8lUFc/216RP0TPkZ7go896d7dem+GRvfBbvGV9vF
nZrMnKJeQghlkIcp3WTNA43yeiflrd5sizku22qCUg4KFrQqCz/WiwEJgCeEJ2hhtb75Q
/MR8JSIJNB5pP2asKyGcu+/pmXnnBYaJE36mazbBxUmq9+QpS8serpTDqiJ6L1w9FEGdY
snflSyedNf7UMw1xZWXtZ19pr4BIPNkC+xTv1WZoKxoPbm/M+P8kiuAL7tflurbhezGc9
TZx6kXvzv768IPmKhPn1ajXXGeDqPwmK9ZbVwOZJcj4lZjQBLoycHKTaX50NV0iI3X4EG
ANVOOL4zrI7YYydv7uAyXLJHcLATx84MMoJAlz9FuWINRoxoE28lv/Sc29pkQLxfU0aHU
w46aSRvRM9e9N0FUExsvqOn8EqDtEPmjX/PM0GJIT0FC8vRbc7emICQj3g66/TaXecaLT
1LWr1oh4jOlCBSe4Ec95OYmLbwR2bE6BoZaX6XhM6AvzzXyvIW0xZw5RZJLMjZ5aU8g2V
hrsI5YkNMV+7pFLEDrBSl66z1Kw8rOasv4zbyBCicrYaajcF+zJ5g1Q3fPMYSU3ctsYv9
0i6ECKALux9J9y9K/ifhD72MIpmayu3TnQgwvyS2IIUxc1pmyxKUpdICCBoKb6sm8Gh6V
9ucKoxG6oDOycuFGNOVjXg18VU0e7/5wFu/d3JHYzbBEKxuhpM9hmk37KLf6QzoAG9HGl
2oS4ocg4j8gXQ8N6EMixgmyLK+ECpFdBM1U4++/eK/eXd9/ihUsU2gE58wVtu3wJG1tME
ni1Rkv6gDgszKs97M1pCmn9ggBiNEbMI2bkiX59AXXnaZ8aGmkJk7c3uDqMnukpFr03L2
pObshWUKkYHOL7IesGGWvuirAKWYTZGMRTzR0ZB56ZR9Mm1GKBv3yVlIUnsV78Io9/i6U
heN2cYDvUWmHW0bgmjicN91saZVayu0/UiLnd/3nr5xuFs2ZrfGwpBIld/isyqawcREeP
3rNeceuaIFseG1u/Gj1CHSGpZqNVpqQ56pFnxq88ZvdSEZpRC4Tb6ThDlAawHSwZlgc1j
CY5aPVwIZxu6IHifmXS8A2mIrM+dvYu/HBF04jhWXPiI5KRhXlOJ20ODocgFjoSo3FnMZ
OZ/8l/SzjaTCJRXAI8uxEoLewHmN1aGDltGutUQ0Wz0uqRfHswgCCQX6D7HEANeoSckwx
hyxfyQW+wa+lnjiK7OJ7uobzGe9HavmwzfHWbZorXd7gvKLInePx3xIAylVe+85KgkM1k
mDMjwfmTSxVr62JOlbVka1zPj0GSRHSlqQ5mFQqcLTrdFF6SMGd7UjZqt8hEs2rbiQY/H
cLAE97+/dcsOxc30qWza3XhLL1cKpOqJ7DH3+4W1AgUC5zjYPPXqw9So4fAOs3rXwTftH
FFcB0FNEwo+OMBVduCkHIIuCXMz/ktVhGspjrZI2am9BOr6jy7J8fI+RdhkKmLiwyzwSM
00WKlk/gI10loHWxOwpFQpekkFQe9KmVWr1pXwXIrRO0KK/cOmdpjr/0ArsBpvFkKskZl
4X/wJyPqZtYOgpwE0F0NBfLPw/kyvEUJsK3cENHgZtEcYRRa94FaT7sYcd6zI+ijGAQYj
6aLEHzle3nR4qKgODz7NSR78a/49rXnypiRdNds+Bzsjo21pUdscPjDYPC/jrJePC2gtD
3KlXH/Wy6dSKyf1nSE9Xp1d9oCMRzv+Tn9LAbDnXU8QpE82KcUPGr/15tK/j7YA1sf4Ii
1kaSHQJBebvemIY+d0NQXkXRxK9rUu2Xo3m8zSLRm9YRgjGW332q0r3K3M3adQV0ZIPNV
sy2ARiQ5IXV80gs0liX2XNJDjtckSwx8H2Ak/+KHp7fGOb/eIqMoJautPS+J7Q3Z3ojBY
3+f1rGR7RNS8OjUyT7WxAon2kR+VMb0M2YJ4mXk7ZHpXvLt/ky2I+/5gc3eD+6N7l8Tii
dzja1qPIySBkBawps6R24Xya7LWeK4iVLGHad3hWOkLurJaBVMn8Nf3MmznkH+Je/LG/h
pnngO4kzSUPE7zM0s+XDJwv7ufuzfF6tkA08f6NhVDkIZafFTRoXN2/MAZuRlMRruN8Tt
GYZ+XDaSLfcdLquQS/5PIQyw0rhCA1Una0utwu7YOSFUholQe+miR28ZALRyxt8GSpfu0
y30064MKz3B/5bzNH6kVqlYYB1dVW3o8+yoKXcFiwZ2aAboa6o3d/tCd/sy7efBaRazF9
aZPD3BZCdFrcqvucn8NJfhnUVWsLg3jquoIOoYMz6YxL2iKXGuQ9rqo1ULTgQ4l1TVXpW
ajDQjbmwkYvc0G73Oz1A3RsMQF+zJLy/mj6GWwvVTK0s8LEtmQA1bMZf/V0qXVdz8jMjX
8xqGINUB6/S4qtGwlZTzFnPiym60V9dvMKUvTGbmk/a1ZcgPuxueni92tlOtFe9455/lz
/8WLXkFlc6l0R42fdkAruIioMpOv8Vawri+80hLyMM5PAgp46JwPp6eaADPdBvxnkmaE5
Zvuc8RReqi3CutMXCpc5a62O9IyTVHEHU0HLeacITQ6oqKppuSdYpEj2UhPhw9/VF9wJF
wNffoNsUSFbPt/hRV3iOIIimIAfX/94575e7NRsWjzxamBXgP4D+LLRExM35Pzh5YQ2k9
hjpMQd8EeNVm+3QNnMBD0v/2aCicSgrVLzWnDOWGrgPma7fyFdPoQSTeDNTlH5ofDv1AN
4cLw1q/PkTVYdhJdx2Zp2taIRCyBvSu7/OVeZrxm3RmdzbU9mNyTFIwQfqkYiZbxmz1tX
0gBHAaboYut2CT/nBCQJgIyls4KmcoLL3v6J2X8oUQ9uYi7eCQ+loE5noQFkM8NtxLSD4
FeVSc5gkhkVUX1l1gh2Qgkjz2ZF2dqzMBBoOXzgHLIXVfChKCbcoEpjuCU6hhlkmiG78h
MKwFWtB0XCC5rph5ND4NuT1+0FTqto4YkC2JxBluQ3gyCMWX07/EuvAdIh42UTrp1Z2a2

wbXmN1vSoL03KnAL8tc/ZTLdSNdICrzmYYzN5HRp3SH/CbpEiwPq7wFx+kQKw8EPU8bh0
LBbLt01WVE5Ez3qbGaazCYq9CGgUgeEIhzy8kMNQVzX8sK15iZMAYrTfhEXkIrJfovQwg
JOXv3uAhYG2i0+AeEdorzZ/Fi8duk0DyflSW1i0hyOP513KjUOVbu9ahrDOxis8PdCmZE
dp2Qz9dsIlujA1Yaus9BzoFf2sh+DkTAmd7mIwIcf6kjLusj6BmjyWBQBQHW94WwztZSV
i59/6Rf3d+nuZ6v8f2frZybIjUlP7xuLfLTJ5/JI9HeTY+XnZuqFBA+vIeO0y89LcTl5R
e8Lq/zKxNBV2gVBQxGTr0axitYbkpvpcPjzTpw2W7GnGbU9UpWt8pSEeW/MRQ5YFRbvyW
GLmVNNKN5DhnxLNML0bcKCj1g/o3SC5buwj+rB9GsU3ZtfhgpUNVMq/RzOpxe3bXV4QJX
9d7oWDjOXr7oHlkFOwyU+fvQJ+2c4d/7ny6R7lEYl8sj8oeRnk9EwBxY+clTjp/2zMq8A
vdF9vEMFx267Hb5Dah0J0+Co767vKlKlMEOKw9iXpSUIRxBnboR0SWEqEhAr1rxd5zCE5
s/N3U++kc2uCsgqhk35jV1d+VIvGyCjFMlIwYm9GjMlLJ7nGnEjw4p+lQ5e2xXvw6cFIt
4htEkgfdKrVhjJAeef8aOXGe0ivQ/dixJhdrEdKOpoy3EUpslkejXThWBdan+GYQgDiAq
DF7Hm9iErJOPu0DnRHULDP/Tipu42En017SvxPcKHHC7VHyjY64heCBGCVh422auQi+gp
Mciago9AD/xk235X7EeTHAXiCZpxuZ40umuDQgAx6aGursxvIj8I+e/JGv+CQql1tHUZE
Z/2oOuPl9xu+lvAbaCs1qBHLNeWSKypF9N5LeKu8I3jNUa6OXkWECARRy5AXE/CN6PFdV
/Z97mV8TKA39ugKssHwfVmDpFN2CvrLR0fSqKjna6A6c2Du02ddSAT2tFKYy6rj/TEs8d
e8Qw2a9IgxTCcu1MEcwNtCMkTLNaBNlo89QQiYG5qRwM1VVXhpTTyA82E6KtpXg8k7M51
QZ6pgEnHnMpYQf2V1A+K9EJmNzGGWIbvi28wbsrqmQoKYFrfjLt9S0PIRWAajm3a8iqyQ
wJI/yS2/QbWHq4ZUK+XDQHBw2jnDHg1fd9vQWPtP2/JflhEXFCCj2UN6nSJSi8idkYDmA
rO8YWMrWomcaES23CpnwhZ8hEHfrDP7qMaIOvwb9M9xeTAWF28SWsXcTfMMKmnrFxXolC
nRv5A91u+Lt+XZ195LBKYFmlSrBQVz6/PqsufDePaBDOnF6edBJBjeHewTXJbJ/h0Kcqe
Prn9TQyc05/bOXnn7s+l+PzdJE89XgwX2Xdsjn13e07KVYXZFzUP7Y3vgI9NWkt1l4k3h
Np9dEisgQg6e9x9EnlfPvc1tmb7GNo67qcx9+y8KWcrEcCO6PBTbT7Rk4P5X7yWdwH+OD
bVC/XKKZyRKoMXVPp8WrWGLLPVatCj/bHTYudO8hJ5uhfhiTtMsL534HNt31Q7OptkX+V
Bgq1pmmDxv6bHtVi3P0USJU0AfchSPxnYWIlc0dhfZViPjpV6AgGFS7Nd/L9c410rEtg0
1v8E6ziYu+RCOqc8Bf2ZT29oZvbqyTs8vkvlGEy7d8vM5DLRnzmLueEx1bQTQW7gvsdkt
dHBJV5zlx4UBwKcW3n3od9/VoxnenCLeQdqdPDuLR2bQechND/+Ifs6GUjEKVD/fVQkld
AIvl6iGL6GKC93EWdkmriGPY6cebz51o5W1wzZ+RlW9C4OMi+DrpmVjE9fC8sMqIUQq9q
AAdcy441gjgaZN2qozpPaE+NfF5WPeHuiNEWSuQiiEJJuEJBe12JRMbi1QizdHdaUJnv0
zOAKsWDDI6wA1/9568mJYwH82OwyeJKs8axXUwQeCwjw1F6+BRTbG4xBO+aTlgD6fa+cb
KNmzupwUrEgQD75fCMgNsW0gTRyhgCCSRxzKTR2pkZXG+KrtFCQMjfUeSiJsXGUm8BuNM
242UFXrxP+9xfRb2EQe7fO96uB2HgI/WiTp2e5/75Vok1B2q4UYYCMI23vrNOvLGJVgzw
KuDHnG24YHrsznzYmWBxlpR6aWeT/eCBTJVgwm3e2rx5JD8li1Etoza8bN1RpYIHHn/uS
EcabUyA/VKF13OAebdbvi5T29klksmxLQ4GhFBfZnIoal8IeFc1KXqoej38n2lZT1xpHh
A444THW/zddF/flYquKF7hu6IT3LjiNhnUahm8z9kD1sj8Pkzb8s4iTlmVAqjTJwqhXvB
Do/r4MgrBTkWnRVCQdd2YBzcv807IJJRMa6kcK6EF5NWLeql5Z3EVk7JcfrAjPT+VyByR
gLf+pD9Yd8pBpaCAqaLEUWe1wN9R2CqJS1wSJ41jjdMNU3VZFkkVtnoeq4k7/DiQavohr
BpO0yfuCmKU7EVdS0hfOdJYpI4cpfGo5hMZm07iSLN66AyjUw+dlnSpk1YdKrqZ5PG9aN
v/eIVNYYoPnivAUIx+gKPN7J1bMJSOR53YLqUoPoj2op04eOuM1R30Qv9OQVBZcFcY76S
RHNxJOigar0XlJu7tRqzTVyn/RlSPnJ2ZGDB4hEtvRrAJ1wHcfLtRlashPADZglKt+C0G
bhRtWBNe+MTsS994LgXkUD+j7fbJphm1HCYXtDbVoEHDOHyA2aPRClNNaf3DD3LmXrGTf
6nrRlhYzvwwQTft8MDj+mH1sr3ofzE1LEmNLpocdWxoXm9y4n43w381zJ6C+VFB1dk5iu
TjYCiWDyYy1nNYIyMoNr+V2ICKOWn2tTeswSr+k7128ACusM029okcQsVDkZL0ThQlcn2
p0yM84QAqR4t4MIKkquxqBvSedy3D5qQVA+jAAJ/EuHASTWy3MEg/UPwvCqGe53+p4LLI
hbJI/0BYEg1Mza3gscT8k+NJk4cgaPt8E4aAYSNYo6n0oGF70d6DLJCSC+4gnMyi4SCnR
WU9p+eSuwHK7y+jO1ENdSYR2ZJhbDA305q777p2k/odc/rtBpeO2a8krOoiV9qrWYJuXn
aTVgNEuTdbEAS3Kx+VE/r3Kqkf63FhyoLyRXw2yMeiLHKU48Swc9QVl3F09/u8wwLjdkP
vxSS+R3QXl9Mfzi1Azy3M/Ru1uKBajl4h9oh+BwLOylzNtWwqbbWTSMpStaBkjOYhiqvj
osUIFXeT1Ysm9bbr5EcN3uy1U9LYTdMURnvJLII1N8xtd0XGbqHgQk9zh670IQ79bjOjE

2508z9Zaks6aOsMltCVi2EkqHmYKD72Y4LaM99ES8DuSNwJRVIbRhA+V10plWKhKisuCz
y9bQWDJPNGL97bSA7npJNxmg+5bckHHKRLjmUTLq+HU5Wcy1p8tUWLk9MQOCK6VRo1Gm7
TAnuPk5L1xayraDOM+21tfvMg1aCT6pZ1fxZEmnmDHI4/9ve379l3nU0lG/J8wLUw0+Rl
je/pIbJPPuM41reOwi3UTTjJtLpQjsOrU1b73fYQ9KsfUE/aa09iqmTfRd6s3jA+/PXUq
WmCWcyHp6c4OSNv1r4vEu4lMxpiZw+CBx7hPxhXyb/BBGmphV6xd+15cufki4+rJjK0+/
eUu4ltYxXJ0ToYcgQ3yRCse4PfySUmMQERp4eNl+WY3lsukHRRKM9rsokR69yQ1B4IuB7
MqiJt9AdSDz3wvYBmA/gxKKG7KWbz4AGEOahMsXi3G7l8zdiPCfHT+A28gKCgo3S2B3NQ
w73SU9F3CEFfSepnTE9qFaTHy7XgzEiapiirZaVlPvdJlpLukmwf2ppeAqqnKFCnfoNBo
4W/H/EhY0UBiooIj0OUtaaKniuSu+esm07Teb5F2RXr/Sh+6tfhdISoUFe7bHt+3RDx98
86QMDA9aghezUOJ71TA87mrD5VlHzPcL3AwFIgjDOC0vxezbkOFdT3u/lJCnsiJ3nuQCe
EmYLlXH7tIe6WLLhFpRfbIzegRb465gH0WRbzI/nBY/04y2jRjPcYs4u4zOJcQAbULzfC
+vdB0Txs5yMqFHHrzdit54zB0jhOaFJZaftgRWObg97tD/QXbFYcOjyxYeoK+fSDjO+sR
jXOu34d2KcVYHGTgso698M2phpRRJ9O15BeTks1MtObpnXeUcK/1BL2sBHzSVELRN8ZQo
yWFRgqa3CDKAnIJeHsj/IbNFsoPqrY1/6QyQjSb1wtFibaayEK6BUN+dhwLyRfheRso1m
WbHs8xVnukwQe5JnZ8JfyT07eU3LrSCyvSIe/+LTCmNithWCvN2B2js83Bay2bxXK2k8P
9HpIUlJHvhR8KqSDmVxPtfx/yGUgWJ7iCjnzSoHzbgqBRNaICObijd+hfNuOCWCdw3dBQ
M+knDQXzOvybOGeGE2TnJQv3AQwqnNsnzPwVCcoB1HId34mYuMIYq6Y3QikgLJSjgTmUC
44hB1Fx0ELcVRRFyzlXc5NvfQ37zygI2Lw3MTZNVt7p48faBebAws9DNBR8wop09Rycap
qgt4dquVsO0d4cgmGe12svm1eJ3gTzIXsd0c1NJlGysebBEwM4caOZdEG3pP32Uo/L4Hj
iMaWuIJ6ymRzk3vyppFXDNslsKjcweVfhB+b+N1oHJFuVBtWhnWAucPGs1HkChxdlYRor
rDRiqm9c1gKPEfUAmvw5syto8V0/78c2lvGYtN3BRHzu847QnLJWU8/r3AtmGAKGVOEPB
4gEoXLCdfm019IZrUnMTMd24J+18uh3c5ckl4AyX7AWZSQm8AMa9d2aaDdF5tnU1R11Cw
q3u9eSrYogukTcFvvRPnjck3vOMi6ko83U+wjuFQoFqIHXDj5jWuzS6HM/gXsIX4vFtiD
BJd3o3a/n2pZFU57gqJZIota7Z4WmN37zKPjrhOTxNKeBegW2G/vr6NRcCHEzSYjFJnmv
kSAPAhfeOUNiWHB3Wku+yMktIv2UcVlkvl8uVYQSFFtjKMWy1FyXcUduAmA0YNj/XMSci
6sF1+venkl0wFuDsxpCQoPi56vyjG9COxrEDUdyXgjD6RbUqQWlzo/LhvhrnOf5LBAjKL
UTqS2RPoAyHzKvplOn8s9NwJ6CALUxkGuYC0AG/HwhxIPIGxNxewNO+sxIxKLh0hYhHVt
l4pVlgxqggNaHCni0h5tqQhEUfOdHPfjJu+lGvYV84O1GmrTpzKSHYIUO82XctTt8cTc4
TnSoeAsn30buXucLzWLk8LQPTissLZcu7bx0jWPfzlcRtiyhKKSRHezLSGJkIhpMfegWK
HYshikxJPZXcNmNRui0AUhgqaeDc5tue1MI8ounj+ulhgSMvrjCGoby1k6NDtn3pCxzMI
npUyUr2iqziKQWI/PdnSsiG6rXNiadmvvqj0BuwDcBPT9WxAOymijpZf8DLu8tiMCvvjv
Fi9tQCHGumCb0QZJslc9LVuAcwJI3hua+XFOByjdpDXZg6MNYnsTc5bKJMkksnjclAQw6
YfeNRiDi0SeEYN87GZYFr/yApHYO4mGVHQoZ1nubWY3nHGGqVdN6SUByZLg0K0FGMruKm
IiFDd7tJgQdjXDRPZde1CSP6gF4dP7gUKMv1vbKKY2+u9GVO89/IALkq+wHIgsUqFBMV3
uptUhk70WKDJeNcynJMSlHbzg8HdGWsLN3SitVfchfWrihgBByk4rbuhrYsYffZXjR+gV
HiwPge5AcKvwBHHsK+QjgtMyh6I2k4=

Appendix B. Change Log

00: Initial draft of the document.
01: Update expiration of document
02: Add appendix with example of MTL Mode signatures in DNSSEC
03: Update draft to align with FIPS-205
04: Updated the implementation status section with links to known implementations
05: Added the implementation considerations section on batch signing

Authors' Addresses

A. Fregly
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: afregly@verisign.com
URI: https://www.verisignlabs.com/

J. Harvey
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: jsharvey@verisign.com
URI: https://www.verisignlabs.com/

B. Kaliski
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: bkaliski@verisign.com
URI: https://www.verisignlabs.com/

D. Wessels
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America
Email: dwessels@verisign.com
URI: https://www.verisignlabs.com/