Workgroup: DNSOP Working Group
Published: 2 March 2022
Intended Status: Standards Track
Expires: 3 September 2022
Authors: A.M. Fregly
Verisign Labs
R. van Rijswijk-Deij
DACS group, EEMCS, University of Twente

Stateful Hash-based Signatures For DNSSEC

Abstract

This document describes how to use stateful hash-based signature schemes (SHBSS) with the DNS Security Extensions (DNSSEC). The schemes include the Hierarchical Signature System (HSS) variant of Leighton-Micali Hash-Based Signatures (HSS/LMS), the eXtended Merkle Signature Scheme (XMSS), and XMSS Multi-Tree (XMSS^MT). In addition, DNSKEY and RRSIG record formats for the signature algorithms are defined and new algorithm identifiers are described.

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 September 2022.

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. Popular digital signature algorithms currently required or recommended for DNSSEC [RFC8624] include RSA [RFC5702], the Elliptic Curve Digital Signature Algorithm (ECDSA) [RFC6605] and the Edwards-Curve Digital Security Algorithm (EdDSA) [RFC8080].

This document describes how to use the following signature algorithms with DNSSEC:

- Hierarchical Signature System/Leighton-Micali Hash-Based Signatures (HSS/LMS) [RFC8554]

- eXtended Merkle Signature Scheme (XMSS) [RFC8391]

- XMSS Multi-Tree (XMSS^MT) [RFC8391]

In addition, algorithm identifiers, DNSKEY record formats and RRSIG record formats are defined. The public key, signature and hash lengths specified here-in are based on algorithm parameters that specify 32-byte hashes. This length provides a desired minimum 128-bit security level. Use of algorithm parameters specifying hashes of lengths other than 32 will result in public key and signature lengths that differ from the lengths specified here-in.

This document covers multi-tree variants of both LMS (HSS/LMS) and XMSS (XMSS^MT). The single tree XMSS variant is also covered as XMSS^MT does not provide for a single tree.

HSS/LMS, XMSS and XMSS^MT are variants of Merkle Tree Signatures [MERKLE1979]. The multi-tree structures of HSS/LMS and XMSS^MT are optimized for incremental one-time key pair generation, where the desired number of one-time key pairs and signatures would make pre-generation of all one-time key pairs impractical using a single tree structure. HSS/LMS, XMSS and XMSS^MT also provide other optimizations such as those enabled by use of the Winternitz one-time signature scheme [MERKLE1989][HUELSING13].

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 in definitions of public keys 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

3.1. Algorithm Identifiers

Algorithms used in DNSSEC are identified by mnemonic identifiers and corresponding numeric identifiers that are registered in [DNSSECREGISTRY]. Per [RFC4034], the Algorithm field of DNSKEY RRs is a numeric identifier that identifies the public key's cryptographic algorithm and determines the format of the Public Key field. The mnemonic algorithm identifiers proposed here-in for HSS/LMS, XMSS and XMSS^MT are "HSSLMS", "XMSS", and "XMSSMT" respectively. These identifiers are proposed for adoption into [DNSSECREGISTRY] in the IANA Consideration section Section 7 of this document. It is anticipated that corresponding numeric identifiers will be assigned by IANA when the algorithms are registered into [DNSSECREGISTRY]. Placeholder numeric identifiers are defined here-in for use in examples Section 6.

3.2. HSS/LMS DNSKEY Resource Records

The DNSKEY HSS/LMS Public Key field consists of a 60-octet value of an HSS/LMS public key generated using one of the parameter sets associated with the HSS/LMS algorithm and identified in the IANA "Leighton-Micali Signatures (LMS)" registry of [LMSREGISTRY]. The public key of the HSS/LMS scheme consists of a concatenation of a 4-octet value L specifying the number of tree levels, followed by the elements of the public key for the top-level tree in the HSS/LMS tree hierarchy. The DNSKEY HSS/LMS Public Key field can be described as:

```
L || LMS-PUBLIC-KEY
```

where LMS_PUBLIC_KEY can be described as:

```
LMS-Type || LMS-OTS-Type || I || T[1])
```

where the elements are defined as follows:

*L: An unsigned 4-octet value specifying the number of tree levels in the HSS/LMS tree hierarchy

*LMS-Type: A 4-octet identifier for the public key hashing algorithm and Merkle tree parameters where the identifier MUST match a Numeric Identifier for an algorithm Name listed in the "Leighton-Micali Signatures (LMS)" registry of [LMSREGISTRY].

*LMS-OTS-Type: A 4-octet identifier for the one-time signature (OTS) hashing algorithm and parameters where the identifier MUST match a Numeric Identifier for an algorithm Name listed in the "LMS-OTS Signatures" registry of [LMSREGISTRY].

*I: A 16-octet value that is the private key identifier; according to Section 5.3 of [RFC8554], I is the value used for all computations for the same LMS tree, which in the case of the public key is the top-level Merkle Tree.

*T[1]: The hash that is the root node for the top level Merkle Tree and which serves as the public key. The length of the hash is defined based on the algorithm identified with the LMS-Type field and will be at least 32 octets

The values of the LMS-Type and LMS-OTS-Type elements MUST specify algorithms that create hashes with a length of at least 32 octets. Given the minimum hash length of 32 octets, an HSS/LMS public key field will have a minimum length of 60 octets.

3.3. XMSS and XMSS^MT DNSKEY Resource Records

The DNSKEY public key field of XMSS and XMSS^MT resource records consists of an XMSS or XMSS^MT public key generated using a parameter set listed in [XMSSREGISTRY]. The DNSKEY XMSS Public Key field and the XMSS^MT Public Key field can both be described as:

```
OID || T[1] || Seed
```

where the elements are defined as follows:

*OID: A 4-octet Object Identifier (OID) for the algorithm type where OID identifies a signature algorithm applicable to either the XMSS or XMSS^MT registry in [XMSSREGISTRY] and where the algorithm number field of the DNSKEY RR determine which registry is applicable. OID MUST match a Numeric Identifier for an Algorithm Name listed in the applicable registry in

[XMSSREGISTRY] as identified by the algorithm number field of the DNSKEY RR.

*T[1]: An n-octet root node hash from the top-level Merkle Tree where "n" is determined based on the algorithm type and MUST be at least 32.

*Seed: An n-octet seed where "n" is determined based on algorithm type and MUST be at least 32.

The total combined height of the tree levels and the number of tree levels in an XMSS^MT tree hierarchy are determined based on the "h/d" elements of the algorithm names where "h" is the total combined height and "d" is the number of tree levels in the tree hierarchy. All trees in an XMSS^MT tree hierarchy MUST be of height h/d per section 4.2. of [RFC8391].

The value of "n" for the length of the root node hash and the seed MUST be equal and at least 32. The set of allowable OIDs is therefore limited to algorithm types that use a value of 32 or more for "n". The length of the DNSKEY public key field can be determined based on the value of "n". The minimum length of the DNSKEY Public key field is 68 octets based on the minimum allowable value for "n" being 32.

4. RRSIG Resource Records

4.1. HSS/LMS RRSIG Resource Records

HSS/LMS signatures consist of a sequence of octet values whose length is dependent on the specific algorithm parameters used when creating the signature. Signatures are encoded into the Signature field of an RRSIG resource record as a simple octet string. HSS/LMS signatures are composed of elements, as described in 6.2. of [RFC8554], and MUST be generated with an algorithm consistent with 6.2. of [RFC8554].

HSS/LMS signature lengths are determined based on parsing an HSS/LMS signature and determining parameters associated with the LMS-Type and LMS-OTS-Type elements found in the signatures as illustrated in "Test Case 1 Signature" in Appendix F of [RFC8554]. The LMS-Type and LMS-OTS-Type elements of the Public Key Field of the DNSKEY RR MUST match the LMS-Type and LMS-OTS-Type for the top-level tree in the HSS/LMS tree hierarchy. Table 1 of [RFC8554] illustrates LMS-OTS signature lengths for various algorithm parameter sets. Table 3 of [RFC8554] illustrates HSS signature lengths for various algorithm parameter sets.

HSS/LMS signature verification is performed using an algorithm functionally equivalent to 6.3. of [RFC8554]. A signature is

verified if the final hash calculated by the verification algorithm matches the public key found in the DNSKEY RR applicable to the signature.

4.2. XMSS and XMSS^MT RRSIG Resource Records

XMSS and XMSS^MT signatures consist of a sequence of octet values whose length is dependent on the specific algorithm parameters used when creating the signature. Signatures are encoded into the Signature field of an RRSIG resource record as a simple octet string. XMSS signatures are composed of elements as described in 4.1.8. of [RFC8391] and MUST be generated using an algorithm consistent with 4.1.9. of [RFC8391]. XMSS^MT signatures are composed of elements as described in 4.2.3. of [RFC8391] and MUST be generated using an algorithm consistent with 4.2.4. of [RFC8391].

XMSS and XMSS^MT signature lengths are determined based on parameters associated with the XMSS or XMSS^MT algorithm identified in [XMSSREGISTRY] by the OID element of the public key field of the DNSKEY RR associated with the signature and with the Winternitz parameter implicitly set to 16 per 5.2. of [RFC8391]. Parameters corresponding to an OID are defined as elements of algorithm names corresponding to an OID as illustrated in Table 2 and Table 4 of [RFC8391]. Table 3 and Table 5 of [RFC8391] illustrate signature lengths for various algorithm parameter sets.

XMSS signature verification is performed using an algorithm functionally equivalent to 4.1.10. of [RFC8391]. XMSS^MT signature verification is performed using an algorithm functionally equivalent to 4.2.5. of [RFC8391]. A signature is verified if the final hash calculated by the verification algorithm matches the public key found in the DNSKEY RR applicable to the signature.

5. Implementation Considerations

5.1. Interoperability Across Implementations

HSS/LMS, XMSS and XMSS^MT implementations for DNSSEC MUST be implemented in conformance with [RFC8554] and [RFC8391]. Implementations subject to [NISTSP800171] MUST be implemented in conformance with [NISTSP800208]. The specifications of the algorithms herein (signature format and public key format) are intended to be "compatible" with the uses of the same algorithms in other protocols. Even though algorithm identifiers may vary by protocol (DNSSEC registry here, object identifier in CMS), the cryptographic inputs, operations and outputs are the same. Implementations therefore SHOULD be interoperable with algorithm implementations for other IETF-defined protocols such that the same cryptographic libraries may be used across protocols. In addition to

being specified here-in, HSS/LMS has been specified in [RFC8708] "Use of the HSS/LMS Hash-Based Signature Algorithm in the Cryptographic Message Syntax (CMS)" and in [RFC8778] "Use of the HSS/LMS Hash-Based Signature Algorithm with CBOR Object Signing and Encryption (COSE)". XMSS and XMSS^MT are not currently specified for use in any protocols defined by IETF RFCs.

Algorithm numbers in DNSKEY records are specific to DNSSEC and are registered in [DNSSECREGISTRY].

Both XMSS and XMSS^MT are covered here-in due to several factors. XMSS^MT does not have a single tree level variant and therefore XMSS is included for implementations that may wish to use a single Merkle tree. Providing for both variants is also important because limiting choice to the parameter sets for XMSS^MT would result in less optimal tree and proof sizes for some implementations. This contrasts with covering just HSS/LMS. LMS is not covered because HSS/LMS provides for a single level Merkle tree with minimal overhead compared to a single level LMS tree. This overhead consists of an extra four bytes in both RRSIG RRs and DNSKEY RRs.

5.2. Trade-offs of Hierarchical Trees

The hierarchical tree structures of HSS/LMS and XMSS^MT allow subordinate trees to be generated when needed. This provides flexibility as the entire multi-tree does not need to be generated at one time. This also supports distributed generation of subordinate trees and distributed signing operations using distributed subordinate trees. Distributing tree generation and signing provides for fault tolerance and scalability that may be desirable based on zone characteristics and operational requirements.

Signatures from multi-level hierarchical tree structures are larger than signatures from single tree structures capable of generating the same number of signatures. This is primarily due to OTS signatures from each level in the hierarchy being included in the signature whereas a signature for a single tree with a height equal to the cumulative height of a tree hierarchy will only have a single OTS signature included in the signature.

5.3. Public Keys

Public keys MUST be created with a hash algorithm providing at least a 128-bit security level. Algorithms for HSS/LMS MUST be selected from the "Leighton-Micali Signatures (LMS)" registry [LMSREGISTRY]. Algorithms for XMSS and XMSS^MT MUST be selected from either the "XMSS Signatures" or "XMSS^MT Signatures" registries found in [XMSSREGISTRY].

5.4. Operational Considerations

The larger signature sizes of HSS/LMS, XMSS and XMSS^MT will result in DNS responses that are too big to fit within typical UDP MTUs, due to the size of the included RRSIG RRs. While EDNS(0) makes it possible to transport larger responses that exceed MTUs, implementers often try to avoid the resulting fragmentation [FUJIWARA], [FLAGDAY2020]. In the absence of EDNS(0), or in case the size of a response exceeds the maximum accepted size, as indicated in EDNS(0) parameters, DNS responses over UDP will be truncated and the requester will be asked to query again over TCP as described in 4. of [RFC7766].

The time needed for key generation and zone signing with stateful hash-based signature algorithms may vary significantly based on the parameter sets chosen for the algorithm per 6.4. of [RFC8554] and 5.4.1 of [RFC8391] with this having operational impact.

Methods for addressing the above-described operational considerations are not within the scope of this document and should be addressed in additional documents.

6. Examples

6.1. Examples Scope

The examples illustrate HSS/LMS and XMSS^MT public keys and signatures applicable to the example MX RR illustrated below. The example public keys may be used to verify the example signatures. The example signatures were generated with the example OTS private keys.

6.2. Example Scenario

The signature examples below are applicable to the following example MX record:

```
IN MX 10 mail.example.com.
```

To facilitate creation of the examples, dummy IANA DNSSEC algorithm identifiers are used. There is no intention that these algorithm identifiers have any applicability other than for their use in the examples, as future algorithm identifiers assigned by IANA may be different than those used in the examples. Any overlap of these identifiers with existing or planned IANA algorithm identifiers is unintentional and not meant to supersede the existing or planned identifier assignments. The algorithm identifiers used here-in have the following values:

```
XMSSMT:     20
HSSLMS:     21
XMSS:       22
```

A ZSK for a zone is used to sign RRsets within the zone, producing RRSIG RRs containing ZSK signatures on the RRsets. The RRSIG RRs link the RRsets into the DNSSEC verification chain and provide for verifying RRsets based on verification of the ZSK signatures. Further details on DNSSEC verification can be found in [RFC4033], [RFC4034] and [RFC4035].

The OTS keys used in creating the example signatures are provided as examples. They may be used for generating OTS signatures on the example MX RR that should match the example OTS signatures.

6.3. HSS/LMS Examples

6.3.1. HSS/LMS ZSK

DNSKEY RR for ZSK in Presentation Format:

```
example.com. IN DNSKEY 256 3 21
AAAAAgAAAAYAAAAE2OF4ZYOt5pa4NfYMpHE2P9V4aGCUzUrLNE/Z5aS1h7XgOYnx
J6pVAieUcpdVUH9c
```

OTS Private Key Used in Signature Example in base64:

```
AAAAAAAAAABkZP///////zblWjBP79a0R/y6rG8QB1+DOhelP7+5PzHdF3+J41/8
2OF4ZYOt5pa4NfYMpHE2Pw==
```

6.3.2. HSS/LMS ZSK Public Key Field Elements in Hexadecimal

The Public Key field of the example HSS/LMS ZSK is broken out in hexadecimal below to provide a more readable illustration of the elements of an HSS/LMS Public Key Field.

L: 00000002

LMS-Type: 00000006 (for the LMS_SHA256_M32_H10 entry in [LMSREGISTRY] which identifies the LMS algorithm will use: SHA256 for hashing; 32 bytes per node in the Merkle trees, and a height of 10 for the top-level tree)

LMS-OTS-Type: 00000004 (for the LMOTS_SHA256_N32_W8 entry in [LMSREGISTRY] which identifies the OTS algorithm will use: SHA256 for hashes; 32 byte individual key values; a Winternitz parameter value of 8)

I: d8e1786583ade696b835f60ca471363f

T[1] (root node):
d578686094cd4acb344fd9e5a4b587b5e03989f127aa55022794729755507f5c

6.3.3. HSS/LMS RRSIG on the example MX RR in Presentation Format

This example RRSIG corresponds to the example HSS/LMS public key. It has an OTS signature on an MX RRset comprised of the example MX RR. The OTS signature on the MX RRset was generated using the example HSS/LMS ZSK OTS private key illustrated above. In wire format, the signature is 2908 octets in length.

```
in.example.com. 3600 IN RRSIG MX 21 3 3600 20211203193730
20211105193730 63074 example.com.
AAAAAQAAAAAAAAAEynHceojgM19r1eYeRHEHbNxS+b0IXn0+5jobp9HcR4nixqAAfloR
9OghTdO4M0Sjs8QQwqp4s/zi9kuVHzREtfuO6eynK8aQDTeCBb1yf9b/EuVwmuyB8fkE
MvN0TuPtxgUg4e1wtZgafdpfTnIubnlwarEk6CqYzOFY5D3hHfI8KdbqPmRj0vBsgoqG
GWlhLOLerdDDlRkpUCuaO1auPT9j5Jq+nPGIQddsR4kxfVKfKVZ8klZDQxs9+cNYKMmU
anBdkAxQDzBY3uSV990lsETlEKoKb1GpLYojWucPtp6tyIZAabLhsGfXW+fsTnwJwcB+
eU17SkXd7BP5xmWVYLLIJt5Bkvfht3WVWWnHQXv6vMuznhKBdn5kPTlnUyCZLLbswZMa
cu5GcbjugtBqxA2tRpeYuw9XmaJPgkyz7zCJlRFaEl4xScv9gDg98g1Z/659Lh5docKd
hnndnbBheGc6QFpBB+L3WcpvK3VfIjjsijBDphwtWCJZYPc6JjldyGCiIPH78gTbbw4m
vXqjTVHZEehpBPSo0wt7HN2/fuhh50MCmydidAhhlZOXLv4dUkfvIp5Hn0NAEEmBuccW
NJAd7ASUKuh9pPO5yyJ7lokLe/sRAvmfuypaqe7AMCroS+nW31jOLTRhCMc/WJyTqQCo
aqE8yGW47GQyBhU0JNv6wVm0ssE1INinaFpeNaGuGyQv7qJEjTWLuNiOBEpcDgb80HSx
+0dUM66DTa78ODti6d2H9JvmYZK4WEhtg9vj6iQgmy6eK909WqJocx93OsRp8M4HDgAA
LwpZEr+MMEY1d6SsftpOQES9bo8NncO/IgkSEP3ULvmvhTwAMmcVe9rF69WQOimCd2zK
mDUPw3zsJe/coOC99URV/qGGG51j5jqQw0ng6wfFZ6mxfmdNvPmWxYAZYcLlibnavHyG
iFzpU/PxqHbYa4qELWcP1aLYiSkGK54ODdF5HRGRrFP6zLdk7MHWBHf7UiMRKw3JUHUn
2fnV8Cl7t97m8hiPQIyf5RaJotjj1ePt61st5HPVg66xraA66symmrqB0NlA3wn3TlBH
EYzr7ohJBlh8C6tGw4/dtTDaukSFjE6Tz0PqyQmUg1vafbOZNtmeKI3wM6hx6gHZKBYW
HHSOxojZr2mA/H0/1dzkjJ6l5B4w8306RKWIon73KukOGKfB1Yak9Gkls73KgOYcZWbx
MoJy9YVV2hFOcNr/dTZiu69DHcLCDFwSGP0LjDZvxe3xDoYRziOwTic2vs4WB+URDuFb
5kS8ABD9E4hNLRbPPHVsnaT5TpWAPwWJv3w3J5ul6Kxce5qqSHFwnpLWRfsiXK3K7Kee
TVdUP2zPosJ5KqcFVhzAGo+bGc+HtMiVZ59VQAEKIbn6RJCie/PP1M4X9lB/VAfW1tnU
IPWDpDT+6DTem5j71xPxLdcc49QUPhTw6l+SaVTkTJhAP1SzG20ZlxMIm8WNDGIdbTQQ
LUHSExdsIw8IiAAAAAZJp9+1/ReMohNXPX7CnjWSgv0KUuL9OX80N9rLGnfN38DZ11tL
EpmZWcKuYT7i9wHmEVIvaI4ZUY2K7m74LRGX/pN+0RWuklWE4dEqGLNI7gguyCvsHZ6S
LpWa+iUZ9onVO/IHkgz8nybMOjK6z9z3LrmNWTq7N/FnOTD1hwsSXKe2F/NafdFLI5d9
0o+AUkSfFEHLKpIHipaJ3XYl0sALf7TEUBgz9HNj7sjhqSaSHsqWxxqDLDJc1sxgkuYt
OX/ubpT/yJZpQcKJ0UBDse/HRGvpWMMIAjVky1Rou5bXaAsvvf2RZ8Z7NXEYurstbhPJ
WL0NmFLdzlTRIfcm408pLN/nntC4q9t9h+8IhmRV/oY8/GXvhJ7Ar8buw99WU/Pcu0+Q
ROvjraUa2Qy8nJiiJCNJJTcXrVf1Grs8u9jDRwAAAAYAAAAE5/vxPT8SR7fK19qoIwws
YM0VgBX2HhNXMUj/X5UYQftzqW9rzFMUg/4Kj7w+GZ9jAAAADAAAAARcEDuVUkrpU9wt
Bxp9Nxp+gVGekNtsTggRxyLG9nydkcQpjr/rtFPWYH1RD/0zDumSmx+C6+7dRSNGeVgn
F4XKyADuM6O/mCJ2tN1aXoYjFp29SdpHKqzta0wvTekmdTPNTB8Lc+3JTD8V9T1ADx8k
ecTiP1rrDc52cRTgPnWDSt2NhdESZ1c43nk7Bsg1DDuR9AZPMmmMTOW6aMyiDQt9Eb2t
tWtiBXO3gzegmjXktbXmkSGJN8K/o0tf50CSPeX4lpYK12r0ZQ8OM/jfA5fjMpNHlQHf
14aP/qt1Sa+RD/ekRtz5JxlrCgeE7n8fpP2iW/A7EXQf8bBxDZFgdMgh6tMf8/vpxPTg
ORFCl+f83bZ7Y+i8Q/FxU+lNFijjHIrVb28lXCWocivdLuI21XyOH5xwY7rMOTwNgUBz
SDe0nZ9zIqpJI48SWEqDT08ID7iBJUk7S5lpNrYm3jzetlziuLrQxnmfPmL13f5xdWhJ
J0QwcQB7/9IoslN9Obkzvff8PWRoGSgLBYczu0dPFOWw/MvFlVKue/cVBR/Jm7A09unA
V/i7iksb++CdpcpOpB8CDpdfYR1TcWUGODhn/OOe4VRQ1ujo11cry+cM8yFciazDffZQ
N54mhTner/mH0ko/ED95Do3Gz897DmpMFmdxypP9UVSNuMS1m7RxlPUJF5gHU2swzi4U
DMyQl+8q5r1PNx1mBXo+ylVnRLa4aRWb3us1PJUe3VZjQH4b03XF2EOcok1RVBrHZIQi
7wHk3QOS2CL2C0yXXkr/uvQ0qqREXiXzQY2vBUiXFa0qS2Z8JEh6iVyQno1r09CDOdTZ
eVhJor5pGAT74pDzhJSJ9lDhBbqI5wBUk/Y7s+NbF8X0UgSpBToyUvDo+i/BuulFrgM+
nfmSYSqItLVfpWtRwzoQxaHYAbC/OEdOsUS8xxIUi2LB/Gs7hUKsLoG2YqRp+jqSISm5
+VeocjvU6rbQPUB/BC0a2L5Xyvd3zdcDe09ZSD6svHEpqetp4AHdUiqBqxBVKVDfvInp
FXkeZ3s3F2fPgwAkAidadDH9OpDeCDEJyMUhAmi1n3hGaUqF5Pob4RuGJ235wGpQSJnp
```

N/VilDYDualtN6O1bm2QswMlugfU/4V4fqkc2+htkU8ZiNVDUthirfgBf2+rnqtzFMS5
/ciCHpUE2hVvQ7LV2zUDXo/K+6iQRHg+nycaQ9B6UnpeFQ8QS5u7T8VrasooTrZPSLTP
LafiTaJWdKFwmLFDXYQSas9gmDMRej1OkLwMGWtL+UbsjrLCmds9W23TBqcteyaKx1va
O1eNTCEsqipOaC553SnUiwp77ImW9MxIxVPQGb6h31X7P/8Kq5Xg/ayFKRvXkI+I4ZmL
6Qk/SzkI3CvdssgyQAouFSCRS64Jlf5cZXVXdxy6HP4TLfWvEPoO3S3jElqvDBC6yPYW
DxHEAbP1NzPl7+Zqohzvi4JpH0qsndSHuHfndr7eoIwnY5BzkTrfAAAABnKlwPzZ1s6r
S4Xq/PWp6+55bQ3SRAwbYlG5J4hXCZiYsYSznaor6R6PDYPYqoRFaFwFkS8f1QIqYUql
Vk50xqjuVod+6DafgrSosIDseLjKllw3kXcUCLVT1M0QbeGxzWGGwLR4UeImrDn4uShA
Ng5SvotYccWZbgak0GV8FUvzP37THwNqy3Ag52Bac3BqSvjjzZ2lpJRy/5yT4ap5b+Qs
0TCMwsHZCY7M/8iVtENd/gq2/Tn5JggvBjwH775qtTfujTWZjUS9JZsHqjITWuxG8VsV
3KUHbovGyr18RjSzgJ6UtStWhlTB8idCaooGGA0RjXNK41+N9gpU3wAOcVpXvy8zhcwF
cMWvoPvm3ldN/EwSheFs4+t5sFQWZXeybfsVYLC/uNL8qRoNLOBBeuzTCtJuMw9bPAPJ
biUrajoc

6.3.4. HSS/LMS RRSIG on the example MX RR with Elements of Signature Broken Out in Hexadecimal

The signature field of the example HSS/LMS RRSIG RR is broken out in hexadecimal below to provide a more readable illustration of the elements of an HSS/LMS signature.

Nspk (number of signed public keys in the key hierarchy = number of tree levels minus one, as the top-level tree's public key is not signed): 00000001

Index (index of the OTS key from the bottom level tree that was used in creating the bottom level OTS signature): 00000000

LMS-OTS-Type: 00000004 (LMOTS_SHA256_N32_W8 - uses SHA256 for hashes with 32-bit hash values for key elements and generated using a Winternitz parameter of 8)

Random Seed (the random seed is used in signature generation to make it harder to perform attacks on the signatures):
ca71dc7a88e0335f6bd5e61e4471076cdc52f9bd085e7d3ee63a1ba7d1dc4789

OTS Signature (1 row for each OTS signature element plus the checksum):
e2c6a0007e5a11f4e8214dd3b83344a3b3c410c2aa78b3fce2f64b951f3444b5
fb8ee9eca72bc6900d378205bd727fd6ff12e5709aec81f1f90432f3744ee3ed
c60520e1ed70b5981a7dda5f4e722e6e79706ab124e82a98cce158e43de11df2
3c29d6ea3e6463d2f06c828a861969612ce2deadd0c3951929502b9a3b56ae3d
3f63e49abe9cf18841d76c4789317d529f29567c925643431b3df9c35828c994
6a705d900c500f3058dee495f7dd25b044e510aa0a6f51a92d8a235ae70fb69e
adc8864069b2e1b067d75be7ec4e7c09c1c07e794d7b4a45ddec13f9c6659560
b2c826de4192f7e1b775955969c7417bfabccbb39e1281767e643d3967532099
2cb6ecc1931a72ee4671b8ee82d06ac40dad469798bb0f5799a24f824cb3ef30
8995115a125e3149cbfd80383df20d59ffae7d2e1e5da1c29d8679dd9db06178
673a405a4107e2f759ca6f2b755f2238ec8a3043a61c2d58225960f73a26395d
c860a220f1fbf204db6f0e26bd7aa34d51d911e86904f4a8d30b7b1cddbf7ee8
61e743029b27627408619593972efe1d5247ef229e479f4340104981b9c71634
901dec04942ae87da4f3b9cb227b96890b7bfb1102f99fbb2a5aa9eec0302ae8
4be9d6df58ce2d346108c73f589c93a900a86aa13cc865b8ec643206153424db
fac159b4b2c13520d8a7685a5e35a1ae1b242feea2448d358bb8d88e044a5c0e
06fcd074b1fb475433ae834daefc383b62e9dd87f49be66192b858486d83dbe3
ea24209b2e9e2bdd3d5aa268731f773ac469f0ce070e00002f0a5912bf8c3046
3577a4ac7eda4e4044bd6e8f0d9dc3bf22091210fdd42ef9af853c003267157b
dac5ebd5903a2982776cca98350fc37cec25efdca0e0bdf54455fea1861b9d63
e63a90c349e0eb07c567a9b17e674dbcf996c5801961c2e589b9dabc7c86885c
e953f3f1a876d86b8a842d670fd5a2d88929062b9e0e0dd1791d1191ac53facc
b764ecc1d60477fb5223112b0dc9507527d9f9d5f0297bb7dee6f2188f408c9f
e51689a2d8e3d5e3edeb5b2de473d583aeb1ada03aeacca69aba81d0d940df09
f74e5047118cebee884906587c0bab46c38fddb530daba44858c4e93cf43eac9
0994835bda7db39936d99e288df033a871ea01d92816161c748ec688d9af6980
fc7d3fd5dce48c9ea5e41e30f37d3a44a588a27ef72ae90e18a7c1d586a4f469
25b3bdca80e61c6566f1328272f58555da114e70daff753662bbaf431dc2c20c
5c1218fd0b8c366fc5edf10e8611ce23b04e2736bece1607e5110ee15be644bc
0010fd13884d2d16cf3c756c9da4f94e95803f0589bf7c37279ba5e8ac5c7b9a
aa4871709e92d645fb225cadcaeca79e4d57543f6ccfa2c2792aa705561cc01a

8f9b19cf87b4c895679f5540010a21b9fa4490a27bf3cfd4ce17f6507f5407d6
d6d9d420f583a434fee834de9b98fbd713f12dd71ce3d4143e14f0ea5f926954
e44c98403f54b31b6d199713089bc58d0c621d6d34102d41d213176c230f0888

LMS-Type: 00000006 (LMS_SHA256_M32_H10 - Indicates the
LMS Tree uses 256-bit/32-byte hashes and has 10 levels in the
tree)

Authentication Path (1 row for each node - contains a sequence of
hashes within a path through the Merkle tree for the signature and
which is to be walked during signature verification):
49a7dfb5fd178ca213573d7ec29e359282fd0a52e2fd397f3437dacb1a77cddf
c0d9d75b4b12999959c2ae613ee2f701e611522f688e19518d8aee6ef82d1197
fe937ed115ae925584e1d12a18b348ee082ec82bec1d9e922e959afa2519f689
d53bf207920cfc9f26cc3a32bacfdcf72eb98d593abb37f1673930f5870b125c
a7b617f35a7dd14b23977dd28f8052449f1441cb2a92078a9689dd7625d2c00b
7fb4c4501833f47363eec8e1a926921eca96c71a832c325cd6cc6092e62d397f
ee6e94ffc8966941c289d14043b1efc7446be958c308023564cb5468bb96d768
0b2fbdfd9167c67b357118babb2d6e13c958bd0d9852ddce54d121f726e34f29
2cdfe79ed0b8abdb7d87ef08866455fe863cfc65ef849ec0afc6eec3df5653f3
dcbb4f9044ebe3ada51ad90cbc9c98a2242349253717ad57f51abb3cbbd8c347

LMS-Type: 00000006 (LMS_SHA256_M32_H10)

LMS-OTS-Type: 00000004 (LMOTS_SHA256_N32_W8)

Public Key Identifier: e7fbf13d3f1247b7cad7daa8230c2c60

Root Hash of lower-level tree (public key for the lower-level tree
which is signed with the OTS signature below):
cd158015f61e13573148ff5f951841fb73a96f6bcc531483fe0a8fbc3e199f63

Index: 0000000c

LMS-OTS-Type: 00000004 (LMOTS_SHA256_N32_W8 - LMS_OTS Type MUST be
the same for all instances within a signature)

Random Seed:
5c103b95524ae953dc2d071a7d371a7e81519e90db6c4e0811c722c6f67c9d91

OTS Signature (1 row for each OTS signature element plus the
checksum)
c4298ebfebb453d6607d510ffd330ee9929b1f82ebeedd4523467958271785ca
c800ee33a3bf982276b4dd5a5e8623169dbd49da472aaced6b4c2f4de9267533
cd4c1f0b73edc94c3f15f53d400f1f2479c4e23f5aeb0dce767114e03e75834a
dd8d85d112675738de793b06c8350c3b91f4064f32698c4ce5ba68cca20d0b7d
11bdadb56b620573b78337a09a35e4b5b5e691218937c2bfa34b5fe740923de5
f896960ad76af4650f0e33f8df0397e33293479501dfd7868ffeab7549af910f
f7a446dcf927196b0a0784ee7f1fa4fda25bf03b11741ff1b0710d916074c821
ead31ff3fbe9c4f4e039114297e7fcddb67b63e8bc43f17153e94d1628e31c8a

d56f6f255c25a8722bdd2ee236d57c8e1f9c7063bacc393c0d8140734837b49d
9f7322aa49238f12584a834f4f080fb88125493b4b996936b626de3cdeb65ce2
b8bad0c6799f3e62f5ddfe1756849274430710 07bffd228b2537d39b933bdf7
fc3d646819280b058733bb474f14e5b0fccbc59552ae7bf715051fc99bb034f6
e9c057f8bb8a4b1bfbe09da5ca4ea41f020e975f611d53716506383867fce39e
e15450d6e8e8d7572bcbe70cf3215c89acc37df650379e268539deaff987d24a
3f103f790e8dc6cfcf7b0e6a4c166771ca93fd51548db8c4b59bb47194f50917
9807536b30ce2e140ccc9097ef2ae6bd4f371d66057a3eca556744b6b869159b
deeb353c951edd5663407e1bd375c5d8439ca24d51541ac7648422ef01e4dd03
92d822f60b4c975e4affbaf434aaa4445e25f3418daf05489715ad2a4b667c24
487a895c909e8d6bd3d08339d4d9795849a2be691804fbe290f3849489f650e1
05ba88e7005493f63bb3e35b17c5f45204a9053a3252f0e8fa2fc1bae945ae03
3e9df992612a88b4b55fa56b51c33a10c5a1d801b0bf38474eb144bcc712148b
62c1fc6b3b8542ac2e81b662a469fa3a922129b9f957a8723bd4eab6d03d407f
042d1ad8be57caf777cdd7037b4f59483eacbc7129a9eb69e001dd522a81ab10
552950dfbc89e915791e677b371767cf83002402275a7431fd3a90de083109c8
c5210268b59f7846694a85e4fa1be11b86276df9c06a504899e937f562943603
b9a96d37a3b56e6d90b30325ba07d4ff85787ea91cdbe86d914f1988d54352d8
62adf8017f6fab9eab7314c4b9fdc8821e9504da156f43b2d5db35035e8fcafb
a89044783e9f271a43d07a527a5e150f104b9bbb4fc56b6aca284eb64f48b4cf
2da7e24da25674a17098b1435d84126acf609833117a3d4e90bc0c196b4bf946
ec8eb2c299db3d5b6dd306a72d7b268ac75bda3b578d4c212caa2a4e682e79dd
29d48b0a7bec8996f4cc48c553d019bea1df55fb3fff0aab95e0fdac85291bd7
908f88e1998be9093f4b3908dc2bddb2c832400a2e1520914bae0995fe5c6575
57771cba1cfe132df5af10fa0edd2de3125aaf0c10bac8f6160f11c401b3f537
33e5efe66aa21cef8b82691f4aac9dd487b877e776bedea08c27639073913adf

LMS-Type: 00000006 (LMS_SHA256_M32_H10)

Authentication Path (1 row for each node):
72a5c0fcd9d6ceab4b85eafcf5a9ebee796d0dd2440c1b6251b9278857099898
b184b39daa2be91e8f0d83d8aa8445685c05912f1fd5022a614aa5564e74c6a8
ee56877ee8369f82b4a8b080ec78b8ca965c3791771408b553d4cd106de1b1cd
6186c0b47851e226ac39f8b92840360e52be8b5871c5996e06a4d0657c154bf3
3f7ed31f036acb7020e7605a73706a4af8e3cd9da5a49472ff9c93e1aa796fe4
2cd1308cc2c1d9098eccffc895b4435dfe0ab6fd39f926082f063c07efbe6ab5
37ee8d35998d44bd259b07aa32135aec46f15b15dca5076e8bc6cabd7c4634b3
809e94b52b568654c1f227426a8a06180d118d734ae35f8df60a54df000e715a
57bf2f3385cc0570c5afa0fbe6de574dfc4c1285e16ce3eb79b054166577b26d
fb1560b0bfb8d2fca91a0d2ce0417aecd30ad26e330f5b3c03c96e252b6a3a1c

6.4. XMSS^MT Examples

6.4.1. XMSS^MT ZSK

OTS Private Key used for the example signature below in base64:

```
AAAAAQAAIJbk253wVxcPn3z6ztrGaoLgosugukEzj4EQPv1fRSrbBiM1C3cvkrS/g0uE
Oqx4eaOxnUWNK9AxS8Pavutf6mX4gI5f9LRt7OMvtZG9fqzYJ0I/6Oxbhwr5nwh7Mig2
mTzwEuZvQ+UZI9Jd8xYP4vbBzegvkIK5esy3pNjVBR8p
```

DNSKEY RR in Presentation Format:

```
example.com. IN DNSKEY 256 3 20
AAAAAfiAjl/0tG3s4y+1kb1+rNgnQj/o7FuHCvmfCHsyKDaZPPAS5m9D5Rkj0l3zFg/i
9sHN6C+Qgrl6zLek2NUFHyk=
```

6.4.2. XMSS^MT ZSK Public Key Field Elements in Hexadecimal

The Public Key field of the example XMSS^MT ZSK is broken out in hexadecimal below to provide a more readable illustration of the elements of an HSS/LMS Public Key Field.

OID: 00000001 (for the XMSSMT-SHA2_20/2_256 in [XMSSREGISTRY] which identifies the algorithm will use: SHA256 for hashing, use aggregate height of 20 with two levels of trees, each of height 10. Note that XMSS and XMSS^MT both assume a Winternitz parameter of 16 for OTS signatures)

T[1] (root node):
f8808e5ff4b46dece32fb591bd7eacd827423fe8ec5b870af99f087b32283699

Seed:
3cf012e66f43e51923d25df3160fe2f6c1cde82f9082b97accb7a4d8d5051f29

6.4.3. XMSS^MT RRSIG on the example MX record in Presentation Format

This example RRSIG corresponds to the example XMSS^MT public key. It has an OTS signature on an MX RRset comprised of the example MX RR. The OTS signature on the MX RRset was generated using the example XMSS^MT ZSK OTS private key illustrated above. In wire format the signature would be 4963 octets in length.

```
in.example.com. 3600 IN RRSIG MX 20 3 3600 20211203193726
20211105193726 44758 example.com.
AABLykUnzyJ2aWKh8e9smKgzAIlASWsKaPYjn9+sNU6y2e6oQUKgfvtljYtN/K8KBsgu
/crA+ONXlmiw3unYS47sN+J4/xIIWHOG+Ojn+ZWeGQdPQ7M5i+cM9yytSymTm0z1c1Oq
EC1GdYJQqnHQ1v/AErOVUFhDKIfh2UFh9bJuQD+zacpMYyIGUakfHBF/9wcw5OHWD9Hd
qDtyiGygYQEzxgj5ZY6tC+nfLQHcCgjqMx9xudTuvc1oyoDLWlCTiwqBzUuUh+5z5jqH
GafAGoV/Ymp1s3xbCGmwlSCHEdTeZD8BB9ZZtVVi20ame3cpvUC37NSegVrwLq/Gtdnk
MdRNUW4eSDEjoDQX+4Z3AtZ86bkG/asidXM581YovrjHTVxNeoLd5dzEW47Eqs/YK2Uf
O8D8gmlWiCO94zHOgoKoFkRTmThaNioKlcoLbWpjCWMqfmbhGko0Run0lb6u36Dn5rJ3
O6zAA6r3PxAMlZrB8cwAn5h8Md/DLvcePwqVMB/33OVatPsZbd9SDgnBTeKW7Gd7LQkb
WKS/XnvEIMgrrm9Isp0xUzJVdb/F5wwEWEZQVKdDuR6Jshtotbs2lBY8vwHXlOlK0cNS
jdjfiRhuQYNR60MvVRRtHsZWHw/N+FfbtVV4dIyP1RqwtfaX8vlVZ3eHsREdoPlx877p
DEsWQdd0Xj88uph1UJKynmVsOrjPMArFzweoeG0jDHbmroSevUKfGOlDIoG8L1xEAk2t
qkHRv63+2xhBnWd6fw/OuH85WQSi5tGZ2EnaCXdAh8UhPOg2J0rbwXFI0f/QITDIkGtl
CtqcGU6U3aXBCGekufwvPSWCY9KyVptNBFS5aWj0bUWLZaUuYapU1S/JWlUj1x8TwCaU
PNfVS4lBqbUNP8fsM5P9pfnKl7ZKD3q9RrKO1b+Zo8I9zIL7s4LqfKcD6WJJeYFI79QD
f7upKRmmtjvahQp56BqFFON8pWxuhJH8FAnKDS/e9/0DXTtD1sMgtT0mnbGBsYz+57hr
tvLV2wp6sh+AgdGKIHlvhaETHKE3DzqJKR2ZH/YNM/WIm/l+coplZgTLh2ZbW6tlYMfR
JxXYoHMJFjHT1uHB7g1XSWiHoFCosTIcweTfSq058LFJOvTcuQzXNXWY1FXHjF65jcPZ
s1G3Fgi5MX+LqXBDxZuFC/WMOJDrNbuuW9xZcB94uMB5IHNHpKS8d9svMQ+YucEF9t4b
B9nUS817AB7hwS/Ke1L4UhbX8QEU7P29uvAL+6/Us/f0A31et0G7MEyhpf5FDKPXnAR0
s0FtWMy+e6TM5GQ8BsIHnYwEvPpazmwaZJwHQi5waZgOn31j1BWJiJT0AD2k+e2lJ40g
nU6vbCqJC1NcBa99kciVBmot22VvJ0qfNpYGs6GhF9bEMm6/g3JXe59WC1WZCp/hhcSZ
V8s/Wrttoz4e3CIWof3C+C02oveofhHAc6ElSC4bRCfBniOz6WSL3UE/rAYvfl1h8x3B
8YyQ6PRVsyTAQQz8jXUDbNTNBe9/vgz5ZSH2IcSda/1TY2a4FKSiCMI9JK0Du5qn5yJu
M8321VBdxS4Swz/mh0RJQW3/Joqr4HWHG++zcxc4d9KM+JIyrx5Ik1eSOxMmJGg24YdA
k2UZ8+mgbOD5L4cyBgx3SMWzrlDK9+LjNiCYSXuJfQ104Rt325efDIpokY11xmJc2c+i
eU+ZUB+CCRpht6pXw/nLSnk8Hda2LXz3zX5KEhW5Alm1gIN/XpMGKl1uEBo4KhwaHYi9
TlsSsjRrMJgRxMPAfEcB64w9R3nsZGjHqKj1H01vhDWdNX6x+wbP24Pr/4t0mb1WdlK7
fzA4FPLkWzWhgEsGZyYAgVw15EaweCMf5Dk7wA1QKD6iAApZ/r0Ep1yOyLWhsfCKJxcH
DEMDIR/Rj62bUCutA5V02reA5HWUikCeSlCDNxMb02G9B2T/FFdWJMQrokmEnemaHy0G
tMemdYPj6rNgkS4fnqd9rR3Q2sqP2AUQ3C3REj43ezt+A9dMiwXVcez8BDmbEG5Mnn1E
moYEstVtJjdAuzkVdRzvDDlF2Z7DLSLTQXTpI5txgcUnhTuE3Et11cNThQYie4tOoNOT
WtHvL8BwU1hbTxT2RyPHB/VGquuxg1VSZvGJeOiYYyava9nVfpF2MrjsTvFONQA9hkuU
OhjBUILWu9830l+dTUUw47wlNAopPwtdkXyxRDtfDzx3WKHov55hWL3OZCy3xGQa1y2H
1c9MipK4XFqOBYsq4r2LZgk5cwrjeaaiyVvjDGQe9ceQFXAZ42w3FRDGfxtuQf7hh0qi
PQhOUaPbGtn1WgBuVAY1ifBg9zlEteUV8ndeLuvM/UHSIUiQLcGCNeg/su6zqLtzrjGM
otBOpv3YR6lUWq8AKumZXnLBNxJPsC4CpQEYyn3PYpYMXYc1HPGV/3bwbS3rWV+Nd96t
IXZ8ULv8CVoKGOdC/gmm2X53Ub/DQC+qcLLlJNTv18TaFZW0bhP/h7otmHwdeesFF51q
6Kr4zdR6orU9qSOblBQ/EAjhgi2LN3ZFuDwgIL4+tPUj2HwZEpnF0KC/hYewE1m6usPA
Ap/LMR72kcnkJ9ozOkFR1Et8Pbn2bgkq/91QY5fK2mt5urFPhjRmMCT19/IfJbN2JGgF
NSOmbyWEDhqCg/wvSPHOhWERq0Y1wh0SE5UaJsOpuKQfioUq6H7LMrBBWH7q/RFtWljq
Nvxwsz3LYvywyWokWFl+Cd07BhQROJQP8GeRzTxHLzORFfVBG70zYrcY+JMtwG4oyT6R
yUFzQtJNM+T0JkrkTpLK2JRDFKG5bC2vK7fvRsRpiQtmtvq4K/Umyw94HhGaQiq6c4kw
2pV+IpJZuwPD9FytpyRtOUl5bGBzMjf7rsXqLpV/dEbp2TYGWgjkaZFFMQPxjj+tcVoe
6XIkQDmfNfHAvm5Du/uYT+ZwbyukXkB40Gs4WBLMfKi3PqQeFWYbxBgtkdD7TsQKsP9+
bxfO/fsJNgvxg80y0LnTL+Fuogtrsavlx6z7tYnnxvOdVCtcdNCnt3xwQZgWMydJzP//
jIVBWdeSlBO0/Wc8924bZgH2A/BO3CP/Hc7HJcVQD2sIP5CMwnPoLuu8f+ahi7Te5l4K
```

```
thKGGfXEEcRYOeQ9+XcJ5MHlqIpcySMjlk5wn0dYylepQMh5AKYFc5D6ynOgJgMGkW+e
xCMF51ZRLXB3puaFutfV7gKWpqESvhNXE2fT/RHmxDms5Ag2yNwxQDZTTxjiZhXxE5aI
p3di5p09QGtPPfv1YjSlQeZ2RlTqXQhBdbW7CLz/8WCXWdpNCkI7Re1X6F6gP6ysRgwO
5s0EGWqTlgHDLXFtEYd3OHYiMqI/s2R/D8QgtKcpz+wCPINVDqdXjXLXzyIy7IlM31vE
A6tXu8SeNXFcysj3RNta3AvrVGwTLVtlRIIn7g6S+s3t9XGSLHqw5TfNQnJyfMdtn+1B
iI9Xu713J2E0HVBHawAeaewuGSGVBizUc8xqJXh8DkWgTRdijtBqKfbvY5ctqjZtW3IZ
HwSJuynbFHVmi8/czsaymj5cBV7g4aLzACc6XgMaU6A1fZ9EJ8onlFanXEJQxCh/ik3b
c1IQvVug7FVPjTiv3yp19yGBedU0SlHVciOeam0wPA3fG73Rov6bzHp80ihYgYNgs7ME
MUFiqzHL15IbEVFEcg6IzakISVleRememnJPAzcae8Z/HILXUmx/IokIvMOeQcYt3HTa
XQLX7TzY2o2AIX/1QHpLCKZqtzcU91eapVkF1u5mWD3brwGdMlk3Qa4DphqeOe+4QAb3
mwjuILalyBeKEC4iqcw5Y05FKzNeU7WU7ms8VCzcmrTh8gpzZcY7f4HXoWGLpYsm5ist
ehmFlKi0c1oeZi5zBKRfP6bP9cgZo4KkdLUow+BKSOhrK5lWcOAyLJrXet7g5CiKaWyt
TUV5fxnuNiJvCjottu6QTS1ySAJR02uwpubY1BwLs1D9bRBeCZiCS02nwuRHIox1PaOv
6KO4sdcW+/Cp5BTElyQfvV1ubgvywQzCBVHA1xktue3kGFBiB6ujez2xfVrU10e6VgKH
DmPjCyxQ/jvlJ7VjjwOOM9nVvwS0qDoDqQ0dfkZFlsJdA7YvW1mlV4XyKwOH5FDkRfZt
MKMAg5usdPE/bu5b21R8aU9dNaqahTmP3hj48vZ41gssvGOKsG8sPTgPK7lxVRTvQrg8
8CpIi1L04kfHlXdvwmiA92JKKW72BrKRqK06VktyQ1FEm3fl72LIKC0wZX4d95lWCbN0
w57CsuEQ50uDylToCB1BbTDlPoEEWF+ZuaZS/rB3DN4bQKFYBDgsuaJElJpbWTbiS1Wv
2NwsvPlhwLTXAQa2QP61ADnIuxMV6N9MgZs7rkHWPJhh52AFm3jRfuXOOB7xKCk+DPxE
SamOPeb+u1WPwDaMHJWy8iixpQvBhf4hkgzE/twZJ0Tsl0KXRPkypV7m1ClK1qei6p0X
GIb/5WcKZjghdmVyqXKIu36+QSfnM3FDdSmEt2tRf62cuIfQ7UzlPc2gDbrjL9vSec0y
KQSQnKgDgCxTxj01CtMyMRY8yZuLn/Typk0Nm3HafZhR7HAnsy/d0oENkOwKXqq1RiZx
OOemEKsngP8j6vcUkUE5BnZA8Aspmmciej+Pabwluuz+Gzs3nPNFvOOM4uRbirXuBw7T
W/tBtAoFXxpwEjxMsCvxDOE/RQpy6OufLqB/KtPuSXvwe7Y/W6JjvvflGnbs38le+Yps
NjkFEgXxipTi30qxOz//GWKw8WjaMyK94Em4UQXg2I7Hp/tDV9F7IWlOYvbCHQEotTNP
DmxNdueRXI83YgDYzqYW6dEJQSTBjYiBM97esSIjSvf7gkR1lT4xZOMoRogat9eTzKcv
c/kZ+8Q/qVGf0D60E64Nm/dxRNXzsOjACznpEx6iaCIm3nIu/mMpAciSr+8eLgb7RYB+
XUeMonM2Ozd4D7H3b7pTipv6Xoe9CUZ2NDK+s0yjfIcMmWI9TsEG+B0fpMJYyVP9FfAk
4xGjmw5damlzdZ1iFB/mTj0DQeLsWYBUM/vjcfGJ66M1SswfzopG9yiIJULkigMYSgJ2
D4EkInVKGB2hqO4YLNSgx4qNJEX/lGXdLWrYdSmjyPT41591+JoB9/SV1zX5xMli5Rvy
6cUN0nCWT0VRJoyP23vl2y/p7Gt3kXuxiBcGT+eSbHg5XYPkQguhcKdI7aYZGPo8dvJd
pbRYopgcqDp6Uuw8NWvEBZrAoWskZj7gL954XFNpvu968MeYUAU6LWoSwe1Q67VleC3D
YV8+1x/NEzy/pR6cCUImoEx++D4wRZt2mbHgrjfh6V7JDLmp02CMeb0URdfZNAY1q4VQ
1Cgo3W4ASky2drEcv/yh63GgDWR+Z0YmONcnAH0GcaSC/b5o4PlC/fo0OQldYjRho/CU
nlfPc5FbPUgdlwYeuo3YYMPeRjeAO9vwzlETb3eyvU7WOB8TVvfJzEIWAXV0823xDV0k
4g/zZxg87tgdjNvGl2isZqoK8UCCNOq7tuHcQCDMkMa0WD5um9MetAOuIN+z03yy+MEu
IgrWOWNIrBHo881YdrssxEjv7HSMz+PLF28nSsIPdlHgRZABr/Vz6XE6/S9mum4NrNnV
hbIO1Q9vCXF35jiE/srqg6P5N1SwQWM1O95i9cDynYa6c/VXF02JTnD+bkM/0pjclS7y
kg631wGHeg08BkxMomDoCci7/lrYdVJQdz0ehESFrEuDqn+/0YmWGvymWPD6scrttpil
5e4spXJRumMUGwgAnKXWWIeRcoKehPOvwQUdVFmxoiLBEanNcvvfPX2yFgiB8EfdCP6h
NFWJnJWOJQ6bN2spJFdCImoZpgmGQboH8O4gVPn0Gp5wEgjQCVJTNf9UTM22zrqo2DFD
kXerHy+BZlBI5w4gQnVKX5LJV8v+QRDgHRvdPZHrBuc/9PHiMC9/WEo8+Sxw4HgiHdML
vGzRYWDyLfZZJWUB2aQlmXJnAoir642qSe1Gqe42DEStZfgScIs9cgOJqZSXWyNRjBJU
jPOunhttIQQFDq2J8DXaXBPpoxG1KWh7elvJHwUUdk7Xer/VtF+A/N8KskzMxlI99fi7
jaCgsy4TXO4F8NcHAD7CWBLTtU1mNUF7vL5Dr7bTEXNiDtOkazfYM53ovzZsg79viGUF
xZZcIRDo8Uso3B0DhV6Rs0gQVM4CSRSr7cYeVHbCabVDRXZ/vrEcTZMQcQTdLh023D7l
ylLESSjN43I8JkOkc74fSlBWZ2zQVFRQ48ZtRFlGNF5bu622fYTNothUQ1eAJWCqIFyu
AMncNmF6k2aGsQaiqA1WGzsjfDAollBjRsWf5acfLX3eCFwBq5GQIyL+4xjrugD7tQyl
uI8LW/Kxj1GyiqyAs40Gdbvt7UrB5FNZaZcX6tmmVDdIdDpNLvYwm+PiGMMXmXGI8pFS
```

zOKPTN7tfvcZ4yB5G1Z29BVCpm1NvVOvTtqM0oZtTFWOzi8sDNLGUZEz6LAyXRU/+tq8
x0R0SBCP8s+E80P7sWXNfr47Qk6hZfFNX/KioWRN2+hxjx+K523bGwUitDxa0IV2dDGm
9P5HKhipqyC/D0MGAXJebAAPFAMAAA4QYapx9mGFh/au1gdleGFtcGxlA2NvbQACaW4H
ZXhhbXBsZQNjb20AAA8AAQAADhAAFAAKBG1haWwHZXhhbXBsZQNjb20A

6.4.4. XMSS^MT RRSIG on the example MX Record with Elements of Signature Broken Out in Hexadecimal

The signature field of the example XMSS^MT RRSIG RR is broken out in hexadecimal below to provide a more readable illustration of the elements of an XMSS^MT signature.

Index: 00004b (index of the OTS key from the bottom level tree that was used in creating the bottom level OTS signature)

Random:
ca4527cf22766962a1f1ef6c98a833008940496b0a68f6239fdfac354eb2d9ee
(the random seed is used in signature generation to make it harder to perform attacks on the signatures)

Reduced signature for bottom layer tree:
OTS signature:
a84142a07efb658d8b4dfcaf0a06c82efdcac0f8e3579668b0dee9d84b8eec37
e278ff1208587386f8e8e7f9959e19074f43b3398be70cf72cad4b29939b4cf5
7353aa102d46758250aa71d0d6ffc012b3955058432887e1d94161f5b26e403f
b369ca4c63220651a91f1c117ff70730e4e1d60fd1dda83b72886ca0610133c6
08f9658ead0be9df2d01dc0a08ea331f71b9d4eebdcd68ca80cb5a50938b0a81
cd4b9487ee73e63a8719a7c01a857f626a75b37c5b0869b095208711d4de643f
0107d659b55562db46a67b7729bd40b7ecd49e815af02eafc6b5d9e431d44d51
6e1e483123a03417fb867702d67ce9b906fdab22757339f35628beb8c74d5c4d
7a82dde5dcc45b8ec4aacfd82b651f3bc0fc8269568823bde331ce8282a81644
5399385a362a0a95ca0b6d6a6309632a7e66e11a4a3446e9f495beaedfa0e7e6
b2773bacc003aaf73f100c959ac1f1cc009f987c31dfc32ef71e3f0a95301ff7
dce55ab4fb196ddf520e09c14de296ec677b2d091b58a4bf5e7bc420c82bae6f
48b29d3153325575bfc5e70c0458465054a743b91e89b21b68b5bb3694163cbf
01d794e94ad1c3528dd8df89186e418351eb432f55146d1ec6561f0fcdf857db
b55578748c8fd51ab0b5f697f2f955677787b1111da0f971f3bee90c4b1641d7
745e3f3cba98755092b29e656c3ab8cf300ac5cf07a8786d230c76e6ae849ebd
429f18e9432281bc2f5c44024dadaa41d1bfadfedb18419d677a7f0fceb87f39
5904a2e6d199d849da09774087c5213ce836274adbc17148d1ffd02130c8906b
650ada9c194e94dda5c10867a4b9fc2f3d258263d2b2569b4d0454b96968f46d
458b65a52e61aa54d52fc95a5523d71f13c026943cd7d54b8941a9b50d3fc7ec
3393fda5f9ca97b64a0f7abd46b28ed5bf99a3c23dcc82fbb382ea7ca703e962
49798148efd4037fbba92919a6b63bda850a79e81a8514e37ca56c6e8491fc14
09ca0d2fdef7fd035d3b43d6c320b53d269db181b18cfee7b86bb6f2d5db0a7a
b21f8081d18a20796f85a1131ca1370f3a89291d991ff60d33f5889bf97e728a
656604cb87665b5bab6560c7d12715d8a073091631d3d6e1c1ee0d57496887a0
50a8b1321cc1e4df4aad39f0b1493af4dcb90cd7357598d455c78c5eb98dc3d9
b351b71608b9317f8ba97043c59b850bf58c3890eb35bbae5bdc59701f78b8c0
79207347a4a4bc77db2f310f98b9c105f6de1b07d9d44bcd7b001ee1c12fca7b
52f85216d7f10114ecfdbdbaf00bfbafd4b3f7f4037d5eb741bb304ca1a5fe45
0ca3d79c0474b3416d58ccbe7ba4cce4643c06c2079d8c04bcfa5ace6c1a649c
07422e7069980e9f7d63d415898894f4003da4f9eda5278d209d4eaf6c2a890b
535c05af7d91c895066a2ddb656f274a9f369606b3a1a117d6c4326ebf837257
7b9f560b55990a9fe185c49957cb3f5abb6da33e1edc2216a1fdc2f82d36a2f7
a87e11c073a125482e1b4427c19e23b3e9648bdd413fac062f7e5d61f31dc1f1
8c90e8f455b324c0410cfc8d75036cd4cd05ef7fbe0cf96521f621c49d6bfd53
6366b814a4a208c23d24ad03bb9aa7e7226e33cdf6d5505dc52e12c33fe68744
49416dff268aabe075871befb373173877d28cf89232af1e489357923b132624
6836e18740936519f3e9a06ce0f92f8732060c7748c5b3ae50caf7e2e3362098

497b897d0d74e11b77db979f0c8a68918d75c6625cd9cfa2794f99501f82091a
61b7aa57c3f9cb4a793c1dd6b62d7cf7cd7e4a1215b90259b580837f5e93062a
5d6e101a382a1c1a1d88bd4e5b12b2346b309811c4c3c07c4701eb8c3d4779ec
6468c7a8a8f51f4d6f84359d357eb1fb06cfdb83ebff8b7499bd567652bb7f30
3814f2e45b35a1804b06672600815c35e446b078231fe4393bc00d50283ea200
0a59febd04a75c8ec8b5a1b1f08a2717070c4303211fd18fad9b502bad039574
dab780e475948a409e4a508337131bd361bd0764ff14575624c42ba249849de9
9a1f2d06b4c7a67583e3eab360912e1f9ea77dad1dd0daca8fd80510dc2dd112
3e377b3b7e03d74c8b05d571ecfc04399b106e4c9e7d449a8604b2d56d263740
bb3915751cef0c3945d99ec32d22d34174e9239b7181c527853b84dc4b75d5c3
538506227b8b4ea0d3935ad1ef2fc07053585b4f14f64723c707f546aaebb183
555266f18978e8986326af6bd9d57e917632b8ec4ef14e35003d864b943a18c1
5082d6bbdf37d25f9d4d4530e3bc25340a293f0b5d917cb1443b5f0f3c7758a1
e8bf9e6158bdce642cb7c4641ad72d87d5cf4c8a92b85c5a8e058b2ae2bd8b66
0939730ae379a6a2c95be30c641ef5c790157019e36c371510c67f1b6e41fee1
874aa23d084e51a3db1ad9f55a006e54063589f060f73944b5e515f2775e2eeb
ccfd41d22148902dc18235e83fb2eeb3a8bb73ae318ca2d04ea6fdd847a9545a
af002ae9995e72c137124fb02e02a50118ca7dcf62960c5d87351cf195ff76f0
6d2deb595f8d77dead21767c50bbfc095a0a18e742fe09a6d97e7751bfc3402f
aa70b2e524d4efd7c4da1595b46e13ff87ba2d987c1d79eb05179d6ae8aaf8cd
d47aa2b53da9239b94143f1008e1822d8b377645b83c2020be3eb4f523d87c19
1299c5d0a0bf8587b01359babac3c0029fcb311ef691c9e427da333a4151d44b
7c3db9f66e092affdd506397cada6b79bab14f8634663024f5f7f21f25b37624
68053523a66f25840e1a8283fc2f48f1ce856111ab4635c21d1213951a26c3a9
b8a41f8a852ae87ecb32b041587eeafd116d5a58ea36fc70b33dcb62fcb0c96a
2458597e09dd3b06141138940ff06791cd3c472f339115f5411bbd3362b718f8
932dc06e28c93e91c9417342d24d33e4f4264ae44e92cad8944314a1b96c2daf
2bb7ef46c469890b66b6fab82bf526cb0f781e119a422aba738930da957e2292
59bb03c3f45cada7246d3949796c60733237fbaec5ea2e957f7446e9d936065a

Authentication Path for Bottom Tree:
08e46991453103f18e3fad715a1ee9722440399f35f1c0be6e43bbfb984fe670
6f2ba45e4078d06b385812cc7ca8b73ea41e15661bc4182d91d0fb4ec40ab0ff
7e6f17cefdfb09360bf183cd32d0b9d32fe16ea20b6bb1abe5c7acfbb589e7c6
f39d542b5c74d0a7b77c70419816332749ccffff8c854159d7929413b4fd673c
f76e1b6601f603f04edc23ff1dcec725c5500f6b083f908cc273e82eebbc7fe6
a18bb4dee65e0ab6128619f5c411c45839e43df97709e4c1e5a88a5cc9232396
4e709f4758ca57a940c87900a6057390faca73a0260306916f9ec42305e75651
2d7077a6e685bad7d5ee0296a6a112be13571367d3fd11e6c439ace40836c8dc
314036534f18e26615f1139688a77762e69d3d406b4f3dfbf56234a541e67646
54ea5d084175b5bb08bcfff1609759da4d0a423b45ed57e85ea03facac460c0e

Reduced signature for top tree:
OTS signature:
e6cd04196a939601c32d716d11877738762232a23fb3647f0fc420b4a729cfec
023c83550ea7578d72d7cf2232ec894cdf5bc403ab57bbc49e35715ccac8f744
db5adc0beb546c132d5b65448227ee0e92facdedf571922c7ab0e537cd427272
7cc76d9fed41888f57bbbd772761341d50476b001e69ec2e192195062cd473cc
6a25787c0e45a04d17628ed06a29f6ef63972daa366d5b72191f0489bb29db14

75668bcfdccec6b29a3e5c055ee0e1a2f300273a5e031a53a0357d9f4427ca27
9456a75c4250c4287f8a4ddb735210bd5ba0ec554f8d38afdf2a75f7218179d5
344a51d572239e6a6d303c0ddf1bbdd1a2fe9bcc7a7cd22858818360b3b30431
4162ab31cbd7921b115144720e88cda90849595e45e99e9a724f03371a7bc67f
1c82d7526c7f228908bcc39e41c62ddc74da5d02d7ed3cd8da8d80217ff5407a
4b08a66ab73714f7579aa55905d6ee66583ddbaf019d32593741ae03a61a9e39
efb84006f79b08ee20b6a5c8178a102e22a9cc39634e452b335e53b594ee6b3c
542cdc9ab4e1f20a7365c63b7f81d7a1618ba58b26e62b2d7a198594a8b4735a
1e662e7304a45f3fa6cff5c819a382a474b528c3e04a48e86b2b995670e0322c
9ad77adee0e4288a696cad4d45797f19ee36226f0a3a2db6ee904d2d72480251
d36bb0a6e6d8d41c0bb350fd6d105e0998824b4da7c2e447228c753da3afe8a3
b8b1d716fbf0a9e414c497241fbd5d6e6e0bf2c10cc20551c0d7192db9ede418
506207aba37b3db17d5ad4d747ba5602870e63e30b2c50fe3be527b5638f038e
33d9d5bf04b4a83a03a90d1d7e464596c25d03b62f5b59a55785f22b0387e450
e445f66d30a300839bac74f13f6eee5bdb547c694f5d35aa9a85398fde18f8f2
f678d60b2cbc638ab06f2c3d380f2bb9715514ef42b83cf02a488b52f4e247c7
95776fc26880f7624a296ef606b291a8ad3a564b724351449b77e5ef62c8282d
30657e1df7995609b374c39ec2b2e110e74b83ca54e8081d416d30e53e810458
5f99b9a652feb0770cde1b40a15804382cb9a244949a5b5936e24b55afd8dc2c
bcf961c0b4d70106b640feb50039c8bb1315e8df4c819b3bae41d63c9861e760
059b78d17ee5ce381ef128293e0cfc4449a98e3de6febb558fc0368c1c95b2f2
28b1a50bc185fe21920cc4fedc192744ec97429744f932a55ee6d4294ad6a7a2
ea9d171886ffe5670a663821766572a97288bb7ebe4127e7337143752984b76b
517fad9cb887d0ed4ce53dcda00dbae32fdbd279cd322904909ca803802c53c6
3d350ad33231163cc99b8b9ff4f2a64d0d9b71da7d9851ec7027b32fddd2810d
90ec0a5eaab546267138e7a610ab2780ff23eaf714914139067640f00b299a67
227a3f8f69bc25baecfe1b3b379cf345bce38ce2e45b8ab5ee070ed35bfb41b4
0a055f1a70123c4cb02bf10ce13f450a72e8eb9f2ea07f2ad3ee497bf07bb63f
5ba263bef7e51a76ecdfc95ef98a6c3639051205f18a94e2df4ab13b3fff1962
b0f168da3322bde049b85105e0d88ec7a7fb4357d17b21694e62f6c21d0128b5
334f0e6c4d76e7915c8f376200d8cea616e9d1094124c18d888133dedeb12223
4af7fb824475953e3164e32846881ab7d793cca72f73f919fbc43fa9519fd03e
b413ae0d9bf77144d5f3b0e8c00b39e9131ea2682226de722efe632901c892af
ef1e2e06fb45807e5d478ca273363b37780fb1f76fba538a9bfa5e87bd094676
3432beb34ca37c870c99623d4ec106f81d1fa4c258c953fd15f024e311a39b0e
5d6a6973759d62141fe64e3d0341e2ec59805433fbe371f189eba3354acc1fce
8a46f728882542e48a03184a02760f812422754a181da1a8ee182cd4a0c78a8d
2445ff9465dd2d6ad87529a3c8f4f8d79f75f89a01f7f495d735f9c4c962e51b
f2e9c50dd270964f4551268c8fdb7be5db2fe9ec6b77917bb18817064fe7926c
78395d83e4420ba170a748eda61918fa3c76f25da5b458a2981ca83a7a52ec3c
356bc4059ac0a16b24663ee02fde785c5369beef7af0c79850053a2d6a12c1ed
50ebb565782dc3615f3ed71fcd133cbfa51e9c094226a04c7ef83e30459b7699
b1e0ae37e1e95ec90cb9a9d3608c79bd1445d7d9340635ab8550d42828dd6e00
4a4cb676b11cbffca1eb71a00d647e67462638d727007d0671a482fdbe68e0f9
42fdfa3439095d623461a3f0949e57cf73915b3d481d97061eba8dd860c3de46
37803bdbf0ce51136f77b2bd4ed6381f1356f7c9cc4216017574f36df10d5d24
e20ff367183ceed81d8cdbc69768ac66aa0af1408234eabbb6e1dc4020cc90c6
b4583e6e9bd31eb403ae20dfb3d37cb2f8c12e220ad6396348ac11e8f3cd5876
bb2cc448efec748ccfe3cb176f274ac20f7651e0459001aff573e9713afd2f66

ba6e0dacd9d585b20ed50f6f097177e63884fecaea83a3f93754b04163353bde
62f5c0f29d86ba73f557174d894e70fe6e433fd298dc952ef2920eb7d701877a
0d3c064c4ca260e809c8bbfe5ad8755250773d1e844485ac4b83aa7fbfd18996
1afca658f0fab1caedb698a5e5ee2ca57251ba63141b08009ca5d65887917282
9e84f3afc1051d5459b1a222c111a9cd72fbdf3d7db2160881f047dd08fea134
55899c958e250e9b376b29245742226a19a6098641ba07f0ee2054f9f41a9e70
1208d009525335ff544ccdb6cebaa8d831439177ab1f2f81665048e70e204275
4a5f92c957cbfe4110e01d1bdd3d91eb06e73ff4f1e2302f7f584a3cf92c70e0
78221dd30bbc6cd16160f22df659256501d9a4259972670288abeb8daa49ed46
a9ee360c44ad65f812708b3d720389a994975b23518c12548cf3ae9e1b6d2104
050ead89f035da5c13e9a311b529687b7a5bc91f0514764ed77abfd5b45f80fc
df0ab24cccc6523df5f8bb8da0a0b32e135cee05f0d707003ec25812d3b54d66
35417bbcbe43afb6d31173620ed3a46b37d8339de8bf366c83bf6f886505c596

Authentication Path for Top Tree:
5c2110e8f14b28dc1d03855e91b3481054ce024914abedc61e5476c269b54345
767fbeb11c4d93107104dd2e1d36dc3ee5ca52c44928cde3723c2643a473be1f
4a5056676cd0545450e3c66d445946345e5bbbadb67d84cda2d8544357802560
aa205cae00c9dc36617a936686b106a2a80d561b3b237c302896506346c59fe5
a71f2d7dde085c01ab91902322fee318ebba00fbb50ca5b88f0b5bf2b18f51b2
8aac80b38d0675bbeded4ac1e45359699717ead9a6543748743a4d2ef6309be3
e218c317997188f29152cce28f4cdeed7ef719e320791b5676f41542a66d4dbd
53af4eda8cd2866d4c558ece2f2c0cd2c6519133e8b0325d153ffadabcc74474
48108ff2cf84f343fbb165cd7ebe3b424ea165f14d5ff2a2a1644ddbe8718f1f
8ae76ddb1b0522b43c5ad085767431a6f4fe472a18a9ab20bf0f430601725e6c

7. IANA Considerations

This document updates the IANA registry "Domain Name System Security (DNSSEC) Algorithm Numbers". The following entries are requested to be added to the registry:

HSS/LMS

Number	TBD
Description	HSS/LMS
Mnemonic	HSSLMS
Zone Signing	Y
Trans. Sec.	*
Reference	This specification

XMSS

Number	TBD
Description	XMSS
Mnemonic	XMSS
Zone Signing	Y
Trans. Sec.	*
Reference	This specification

XMSS^MT

Number	TBD
Description	XMSS^MT.
Mnemonic	XMSSMT.
Zone Signing	Y
Trans. Sec.	*
Reference	This specification

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

[LMSREGISTRY] and [XMSSREGISTRY] are IANA registries in which additional algorithm identifiers may be registered in the future for use with HSS/LMS, XMSS and XMSS^MT.

This document refers to allowable HSS/LMS algorithm signature identifiers. These identifiers are found in the IANA "Leighton Micali Signatures (LMS)" registry [LMSREGISTRY].

HSS/LMS refers to allowable LMS OTS algorithm signature identifiers. These identifiers are found in the IANA "LM-OTS Signatures" registry [LMSREGISTRY].

This document refers to allowable XMSS and XMSS^MT algorithm signature identifiers. These identifiers are found in the IANA "XMSS Signatures" and "XMSS^MT Signatures" registries [XMSSREGISTRY]. Separate algorithm identifiers for XMSS and XMSS^MT are needed so that the appropriate registry in [XMSSREGISTRY] is identified for matching OIDs to algorithm parameter sets.

XMSS and XMSS^MT use a set of allowable XMSS OTS signature algorithms. Identifiers for allowable algorithms (WOTS+ signatures) are found in the IANA "WOTS+ Signatures" registry [XMSSREGISTRY].

8. Implementation Status

NOTE: Please remove this section and the reference to RFC 794 prior to publication as an RFC.

There are currently no implementations available for public examination

9. Security Considerations

9.1. Guidelines for Secure Use of Hash-Based Signature Algorithms

HSS/LMS, XMSS and XMSS^MT as defined here MUST use 256-bit or larger hash algorithms. 256-bit hash sizes are consistent with a desired minimum 128-bit security level for HSS/LMS, XMSS and XMSS^MT per [EATON] [SPHINCSPLUS].

The HSS/LMS public key MUST be generated in conformance with Section 4 of [NISTSP800208].

The XMSS public key for a single level tree structure or a XMSS^MT hierarchical tree structure generated by a single computing device ("cryptographic module") MUST be generated in conformance with processing steps described in section 5 of [RFC8391].

Implementations MUST use the same hash algorithm for hashes in all nodes within a Merkle Tree or Merkle Tree hierarchy.

OTS private keys MUST NOT be used more than once for signing. This requirement is needed as re-use of OTS private keys may provide an attacker with key information that enables forging of signatures. A detailed discussion of the vulnerabilities re-use of OTS private keys enables is described in [OOPS].

The requirement that OTS private keys may only be used once creates the need for state management to assure OTS keys are not re-used. This can be a challenge as state needs to be maintained in case of system failure and also across distributed implementations. This challenge is increased for DNSSEC implementations that perform dynamic signing.

Seeds and random numbers used in key generation and for hashing SHOULD be generated in accordance with 6.1 and 6.2 of [NISTSP800208].

9.2. NIST SP 800-208 Compliance

The requirements in this section apply to implementations subject to [NISTSP800208].

OTS private keys for hash-based algorithms MUST NOT be exported from a cryptographic module. In adherence to this operational constraint, OTS private keys for individual sub-trees in a hierarchical Merkle Tree structure including the top-level tree MUST be generated, stored and used within a single cryptographic module and MUST NOT be exported.

In [NISTSP800208] NIST provides guidance and requirements for multi-tree hash-based algorithms implemented across distributed "cryptographic modules". Per 7.2. of [NISTSP800208], these implementations require minor changes to XMSS^MT, and no changes to HSS/LMS. XMSS^MT Implementations which distribute trees across cryptographic modules and which are subject to [NISTSP800208] therefore MUST use the algorithms specified in 7.2. of [NISTSP800208]. NIST's approach is to have the cryptographic modules for each tree level simply implement a single-tree signature scheme, with the signer combining the outputs of the different cryptographic modules into a multi-level signature, without further cryptographic processing.

Seeds and random numbers used in key generation and for hashing MUST be generated in accordance with 6.1 and 6.2 of [NISTSP800208].

10. Acknowledgements

The authors would like to acknowledge the following individuals for their contributions to the development of this document: Dave Blacka, Jim Goodman, James Gould, Joseph Harvey, Scott Hollenbeck, Russ Housley, Burt Kaliski, Swapneel Sheth, Sean Turner, Duane Wessels

11. Normative References

[DNSSECREGISTRY]

IANA, "Domain Name System Security (DNSSEC) Algorithm Numbers", April 2020, <https://www.iana.org/assignments/dns-sec-alg-numbers/dns-sec-alg-numbers.xhtml>.

[LMSREGISTRY] IANA, "Leighton-Micali Signature (LMS)", June 2021, <https://www.iana.org/assignments/leighton-micali-signatures/leighton-micali-signatures.xhtml>.

[NISTSP800171] National Institute of Standards and Technology (NIST), "SP 800-171 Protecting Controlled Unclassified Information in Nonfederal Systems and Organizations", February 2020, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-171r2.pdf>.

[NISTSP800208] National Institute of Standards and Technology (NIST), "SP 800-208 Recommendation for Stateful Hash-Based Signature Schemes", October 2020, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-208.pdf>.

[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>.

[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>.

[RFC8391] Huelsing, A., Butin, D., Gazdag, S., Rijneveld, J., and A. Mohaisen, "XMSS: eXtended Merkle Signature Scheme",

RFC 8391, DOI 10.17487/RFC8391, May 2018, <https://www.rfc-editor.org/info/rfc8391>.

[RFC8554] McGrew, D., Curcio, M., and S. Fluhrer, "Leighton-Micali Hash-Based Signatures", RFC 8554, DOI 10.17487/RFC8554, April 2019, <https://www.rfc-editor.org/info/rfc8554>.

[XMSSREGISTRY] IANA, "XMSS: Extended Hash-Based Signatures", May 2020, <https://www.iana.org/assignments/xmss-extended-hash-based-signatures/xmss-extended-hash-based-signatures.xhtml>.

12. Informative References

[EATON] Eaton, E., "Leighton-Micali Hash-Based Signatures in the Quantum Random-Oracle Model", Cryptology ePrint Archive, Report 2017/607, 2017, <https://ia.cr/2017/607>.

[FLAGDAY2020] Surý, O., "DNS Flag Day 2020", September 2020, <https://www.isc.org/blogs/dns-flag-day-2020-2>.

[FUJIWARA] Fujiwara, K. and P. Vixie, "Fragmentation Avoidance in DNS", Work in Progress, Internet-Draft, draft-ietf-dnsop-avoid-fragmentation-05, June 2021, <https://datatracker.ietf.org/doc/draft-ietf-dnsop-avoid-fragmentation/>.

[HUELSING13] Huelsing, A., "W-OTS+ - Shorter Signatures for Hash-Based Signature Schemes", Lecture Notes in Computer Science, Volume 7918, Progress in Cryptology - AFRICACRYPT, January 2018, <https://doi.org/10.1007/978-3-642-38553-7_10>.

[MERKLE1979] Merkle, R., "Secrecy, Authentication, and Public Key Systems", Technical Report No. 1979-1, June 1979, <https://dl.acm.org/doi/10.5555/909000>.

[MERKLE1989] Merkle, R., "A Certified Digital Signature", Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, 1989, <https://doi.org/10.1007/0-387-34805-0_21>.

[OOPS] Bruinderink, L. G. and A. Huelsing, ""Oops, I Did It Again" - Security of One-Time Signatures Under Two-Message Attacks", SAC 2017. SAC 2017. Lecture Notes in Computer Science, vol 10719. Springer, Cham., August 2017, <https://eprint.iacr.org/2016/1042.pdf>.

[RFC5702] Jansen, J., "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, DOI

10.17487/RFC5702, October 2009, <https://www.rfc-editor.org/info/rfc5702>.

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

[RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP - Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <https://www.rfc-editor.org/info/rfc7766>.

[RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <https://www.rfc-editor.org/info/rfc8080>.

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

[RFC8708] Housley, R., "Use of the HSS/LMS Hash-Based Signature Algorithm in the Cryptographic Message Syntax (CMS)", RFC 8708, DOI 10.17487/RFC8708, February 2020, <https://www.rfc-editor.org/info/rfc8708>.

[RFC8778] Housley, R., "Use of the HSS/LMS Hash-Based Signature Algorithm with CBOR Object Signing and Encryption (COSE)", RFC 8778, DOI 10.17487/RFC8778, April 2020, <https://www.rfc-editor.org/info/rfc8778>.

[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. Change Log

Authors' Addresses

Andrew Fregly
Verisign Labs
12061 Bluemont Way
Reston, VA 20190
United States of America

Email: afregly@verisign.com
URI: http://www.verisignlabs.com/

Roland Martijn van Rijswijk-Deij
DACS group, EEMCS, University of Twente
DRIENERLOLAAN 5
7522 NB ENSCHEDE
Netherlands

Email: r.m.vanrijswijk@utwente.nl
URI: https://people.utwente.nl/r.m.vanrijswijk