Additional Parameter sets for LMS Hash-Based Signatures
draft-fluhrer-lms-more-parm-sets-04

Abstract

This note extends LMS (RFC 8554) by defining parameter sets by including additional hash functions. Hese include hash functions that result in signatures with significantly smaller than the signatures using the current parameter sets, and should have sufficient security.

This document is a product of the Crypto Forum Research Group (CFRG) in the IRTF.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Stateful hash based signatures have small private and public keys, are efficient to compute, and are believed to have excellent security. One disadvantage is that the signatures they produce tend to be somewhat large (possibly 1k - 4kbytes). What this draft explores are a set of parameter sets to the LMS (RFC8554) stateful hash based signature method that reduce the size of the signature significantly.

1.1. Disclaimer

This document is not intended as legal advice. Readers are advised to consult with their own legal advisers if they would like a legal interpretation of their rights.

The IETF policies and processes regarding intellectual property and patents are outlined in [RFC3979] and [RFC4879] and at https://datatracker.ietf.org/ipr/about.

2. Conventions Used In This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Additional Hash Function Definitions

3.1. 192 bit Hash Function based on SHA256

This document defines a SHA-2 based hash function with a 192 bit output. As such, we define SHA256-192 as a truncated version of SHA256 [FIPS180]. That is, it is the result of performing a SHA256 operation to a message, and then omitting the final 64 bits of the output. It is the same procedure used to define SHA224, except that we use the SHA256 IV (rather than using one dedicated to SHA256-192), and you truncate 64 bits, rather than 32.

The following test vector may illustrate this:

```
SHA256("abc")     = ba7816bf 8f01cfea 414140de 5dae2223
                    b00361a3 96177a9c b410ff61 f20015ad
SHA256-192("abc") = ba7816bf 8f01cfea 414140de 5dae2223
                    b00361a3 96177a9c
```

We use the same IV as the untruncated SHA256, rather than defining a distinct one, so that we can use a standard SHA256 hash implementation without modification. In addition, the fact that you get partial knowledge of the SHA256 hash of a message by examining the SHA256-192 hash of the same message is not a concern for this application. Each message that is hashed is randomized. Any message being signed includes the C randomizer which varies per message; in addition, all hashes include the I identifier, which varies depending on the public key. Therefore, signing the same message by SHA256 and by SHA256-192 will not result in the same value being hashed, and so the latter hash value is not a prefix of the former one.

3.2. 256 bit Hash Function based on SHAKE256

This document defines a SHAKE-based hash function with a 256 bit output. As such, we define SHAKE256-256 as a hash where you submit the preimage to the SHAKE256 XOF, with the output being 256 bits, see FIPS 202 [FIPS202] for more detail.

3.3. 192 bit Hash Function based on SHAKE256

This document defines a SHAKE-based hash function with a 192 bit output. As such, we define SHAKE256-192 as a hash where you submit the preimage to the SHAKE256 XOF, with the output being 192 bits, see FIPS 202 [FIPS202] for more detail.

4. Additional LM-OTS Parameter Sets

Here is a table with the LM-OTS parameters defined that use the above hashes:

Parameter Set Name	H	n	w	p	ls	id
LMOTS_SHA256_N24_W1	SHA256-192	24	1	200	8	TBD1
LMOTS_SHA256_N24_W2	SHA256-192	24	2	101	6	TBD2
LMOTS_SHA256_N24_W4	SHA256-192	24	4	51	4	TBD3
LMOTS_SHA256_N24_W8	SHA256-192	24	8	26	0	TBD4
LMOTS_SHAKE_N32_W1	SHAKE256-256	32	1	265	7	TBD5
LMOTS_SHAKE_N32_W2	SHAKE256-256	32	2	133	6	TBD6
LMOTS_SHAKE_N32_W4	SHAKE256-256	32	4	67	4	TBD7
LMOTS_SHAKE_N32_W8	SHAKE256-256	32	8	34	0	TBD8
LMOTS_SHAKE_N24_W1	SHAKE256-192	24	1	200	8	TBD9
LMOTS_SHAKE_N24_W2	SHAKE256-192	24	2	101	6	TBD10
LMOTS_SHAKE_N24_W4	SHAKE256-192	24	4	51	4	TBD11
LMOTS_SHAKE_N24_W8	SHAKE256-192	24	8	26	0	TBD12

Table 1

The id is the IANA-defined identifier used to denote this specific parameter set, and which appears in both public keys and signatures.

The SHA256_N24, SHAKE_N32, SHAKE_N24 in the parameter set name denote the SHA256-192, SHAKE256-256 and SHAKE256-192 hash functions defined in Section 3.

Remember that the C message randomizer (which is included in the signature) is the size of the hash n, and so it shrinks from 32 bytes to 24 bytes for those the parameter sets that use either SHA256-192 or SHAKE256-192.

5. Additional LM Parameter Sets

Here is a table with the LM parameters defined that use SHA256-192, SHAKE256-256 and SHAKE256-192 hash functions:

Parameter Set Name	H	m	h	id
LMS_SHA256_M24_H5	SHA256-192	24	5	TBD13
LMS_SHA256_M24_H10	SHA256-192	24	10	TBD14
LMS_SHA256_M24_H15	SHA256-192	24	15	TBD15
LMS_SHA256_M24_H20	SHA256-192	24	20	TBD16
LMS_SHA256_M24_H25	SHA256-192	24	25	TBD17
LMS_SHAKE_M32_H5	SHAKE256-256	32	5	TBD18
LMS_SHAKE_M32_H10	SHAKE256-256	32	10	TBD19
LMS_SHAKE_M32_H15	SHAKE256-256	32	15	TBD20
LMS_SHAKE_M32_H20	SHAKE256-256	32	20	TBD21
LMS_SHAKE_M32_H25	SHAKE256-256	32	25	TBD22
LMS_SHAKE_M24_H5	SHAKE256-192	24	5	TBD23
LMS_SHAKE_M24_H10	SHAKE256-192	24	10	TBD24
LMS_SHAKE_M24_H15	SHAKE256-192	24	15	TBD25
LMS_SHAKE_M24_H20	SHAKE256-192	24	20	TBD26
LMS_SHAKE_M24_H25	SHAKE256-192	24	25	TBD27

Table 2

The id is the IANA-defined identifier used to denote this specific parameter set, and which appears in both public keys and signatures.

The SHA256_M24, SHAKE_M32, SHAKE_M24 in the parameter set name denote the SHA256-192, SHAKE256-256 and SHAKE256-192 hash functions defined in Section 3.

6. Comparisons of 192 bit and 256 bit parameter sets

Switching to a 192 bit hash affects the signature size, the computation time, and the security strength.

The major reason for considering these truncated parameter sets is that they cause the signatures to shrink considerably.

Here is a table that gives the space used by both the 256 bit parameter sets and the 192 bit parameter sets, for a range of plausible Winternitz parameters and tree heights

ParmSet	Winternitz	256 bit hash	192 bit hash
15	4	2672	1624
15	8	1616	1024
20	4	2832	1744
20	8	1776	1144
15/10	4	5236	3172
15/10	8	3124	1972
15/15	4	5396	3292
15/15	8	3284	2092
20/10	4	5396	3292
20/10	8	3284	2092
20/15	4	5556	3412
20/15	8	3444	2212

Table 3

ParmSet: this is the height of the Merkle tree(s); parameter sets listed as a single integer have L=1, and consist a single Merkle tree

of that height; parameter sets with L=2 are listed as x/y, with x being the height of the top level Merkle tree, and y being the bottom level.

Winternitz: this is the Winternitz parameter used (for the tests that use multiple trees, this applies to all of them).

256 bit hash: the size in bytes of a signature, assuming that a 256 bit hash is used in the signature (either SHA256 or SHAKE256-256).

192 bit hash: the size in bytes of a signature, assuming that a 192 bit hash is used in the signature (either SHA256-192 or SHAKE256-192).

An examination of the signature sizes show that the 192 bit parameters consistently give a 35% - 40% reduction in the size of the signature in comparison with the 256 bit parameters.

In addition, for SHA256-192, there is a smaller (circa 20%) reduction in the amount of computation required for a signature operation with a 192 bit hash. The SHAKE256-192 signatures may have either a faster or slower computation, depending on the implementation speed of SHAKE versus SHA256 hashes.

The SHAKE256-256 based parameter sets give no space advantage (or disadvantage) over the existing SHA256-based parameter sets; any performance delta would depend solely on the implementation and whether they can generate SHAKE hashes faster than SHA256 ones.

7. IANA Considerations

[TO BE REMOVED: The entries from Section 4, namely LMOTS_SHA256_N24_W1 through LMOTS_SHAKE_N24_W8 , should be inserted into https://www.iana.org/assignments/leighton-micali-signatures/leighton-micali-signatures.xhtml#lm-ots-signatures]

[TO BE REMOVED: The entries from Section 5, namely LMS_SHA256_M24_H5 through LMS_SHAKE_M24_H25 should be inserted into https://www.iana.org/assignments/leighton-micali-signatures/leighton-micali-signatures.xhtml#leighton-micali-signatures-1]

Until IANA assigns the codepoints, we will (for testing purposes only) use the following private use code points to do any necessary interoperability testing. Such an implementation must change to the IANA-assigned code points when they become available.

Parameter Set Name	Temporary Codepoint

LMOTS_SHA256_N24_W1	0xE0000001
LMOTS_SHA256_N24_W2	0xE0000002
LMOTS_SHA256_N24_W4	0xE0000003
LMOTS_SHA256_N24_W8	0xE0000004
LMOTS_SHAKE_N32_W1	0xE0000005
LMOTS_SHAKE_N32_W2	0xE0000006
LMOTS_SHAKE_N32_W4	0xE0000007
LMOTS_SHAKE_N32_W8	0xE0000008
LMOTS_SHAKE_N24_W1	0xE0000009
LMOTS_SHAKE_N24_W2	0xE000000A
LMOTS_SHAKE_N24_W4	0xE000000B
LMOTS_SHAKE_N24_W8	0xE000000C
LMS_SHA256_M24_H5	0xE0000001
LMS_SHA256_M24_H10	0xE0000002
LMS_SHA256_M24_H15	0xE0000003
LMS_SHA256_M24_H20	0xE0000004
LMS_SHA256_M24_H25	0xE0000005
LMS_SHAKE_M32_H5	0xE0000006
LMS_SHAKE_M32_H10	0xE0000007
LMS_SHAKE_M32_H15	0xE0000008
LMS_SHAKE_M32_H20	0xE0000009
LMS_SHAKE_M32_H25	0xE000000A
LMS_SHAKE_M24_H5	0xE000000B
LMS_SHAKE_M24_H10	0xE000000C

LMS_SHAKE_M24_H15	0xE000000D
LMS_SHAKE_M24_H20	0xE000000E
LMS_SHAKE_M24_H25	0xE000000F

Table 4

8. Security Considerations

The strength of a signature that uses the SHA256-192, SHAKE256-256 and SHAKE256-192 hash functions is based on the difficultly in finding preimages or second preimages to those hash functions.

The case of SHAKE256-256 is essentially the same as the existing SHA256 based signatures; the difficultly of finding preimages is essentially the same, and so they have (barring unexpected cryptographical advances) essentially the same level of security.

The case of SHA256-192 and SHAKE256-192 requires closer analysis.

For a classical (nonquantum) computer, they have no known attack better than performing hashes of a large number of distinct preimages; as a successful attack has a high probability of requiring nearly 2**192 hash computations (for either SHA256-192 or SHAKE256-192). These can be taken as the expected work effort, and would appear to be completely infeasible in practice.

For a Quantum Computer, they could in theory use a Grover's algorithm to reduce the expected complexity required to circa 2**96 hash computations (for N=24). On the other hand, to implement Grover's algorithm with this number of hash computations would require performing circa 2**96 hash computations in succession, which will take more time than is likely to be acceptable to any attacker. To speed this up, the attacker would need to run a number of instances of Grover's algorithm in parallel. This would necessarily increase the total work effort required, and to an extent that makes it likely to be infeasible.

Hence, we expect that LMS based on these hash functions is secure against both classical and quantum computers, even though, in both cases, the expected work effort is less (for the N=24 case) than against either SHA256 or SHAKE256-256.

8.1. Note on the version of SHAKE

FIPS 202 defines both SHAKE128 and SHAKE256. This specification selects SHAKE256, even though it is, for large messages, less efficient. The reason is that SHAKE128 has a low upper bound on the difficulty of finding preimages (due to the invertibility of its internal permutation), which would limit the strength of LMS (whose strength is based on the difficulty of finding preimages). Hence, we specify the use of SHAKE256, which has a considerably stronger preimage resistance.

9. References

9.1. Normative References

[FIPS180] National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS 180-4, March 2012.

[FIPS202] National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", FIPS 202, August 2015.

[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>.

[RFC3979] Bradner, S., Ed., "Intellectual Property Rights in IETF Technology", RFC 3979, DOI 10.17487/RFC3979, March 2005, <https://www.rfc-editor.org/info/rfc3979>.

[RFC4879] Narten, T., "Clarification of the Third Party Disclosure Procedure in RFC 3979", RFC 4879, DOI 10.17487/RFC4879, April 2007, <https://www.rfc-editor.org/info/rfc4879>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <https://www.rfc-editor.org/info/rfc5226>.

[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>.

9.2. Informative References

[Grover96]
Grover, L., "A fast quantum mechanical algorithm for database search", 28th ACM Symposium on the Theory of Computing p. 212, 1996.

Appendix A. Test Cases

This section provides three test cases that can be used to verify or debug an implementation, one for each hash function. This data is formatted with the name of the elements on the left, and the value of the elements on the right, in hexadecimal. The concatenation of all of the values within a public key or signature produces that public key or signature, and values that do not fit within a single line are listed across successive lines.

Test Case 1 Private Key for SHA256-192

```
--------------------------------------------
(note: procedure in Appendix A of RFC8554 is used)
SEED        000102030405060708090a0b0c0d0e0f
            1011121314151617
I           202122232425262728292a2b2c2d2e2f
--------------------------------------------
--------------------------------------------
```

Test Case 1 Public Key for SHA256-192

```
--------------------------------------------
HSS public key
levels      00000001
--------------------------------------------
LMS type    e0000001                         # LMS_SHA256_N24_H5
LMOTS type  e0000004                         # LMOTS_SHA256_N24_W8
I           202122232425262728292a2b2c2d2e2f
K           2c571450aed99cfb4f4ac285da148827
            96618314508b12d2
--------------------------------------------
--------------------------------------------
```

Test Case 1 Message for SHA256-192

```
--------------------------------------------
Message     54657374206d65737361676520666f72  |Test message for|
            205348413235362d3139320a          | SHA256-192.|
--------------------------------------------
```

Test Case 1 Signature for SHA256-192

```
--------------------------------------------
HSS signature
Nspk        00000000
sig[0]:
--------------------------------------------
LMS signature
q           00000005
--------------------------------------------
LMOTS signature
LMOTS type  e0000004                         # LMOTS_SHA256_N24_W8
C           0b5040a18c1b5cabcbc85b047402ec62
            94a30dd8da8fc3da
y[0]        e13b9f0875f09361dc77fcc4481ea463
            c073716249719193
y[1]        614b835b4694c059f12d3aedd34f3db9
            3f3580fb88743b8b
y[2]        3d0648c0537b7a50e433d7ea9d6672ff
            fc5f42770feab4f9
y[3]        8eb3f3b23fd2061e4d0b38f832860ae7
            6673ad1a1a52a900
y[4]        5dcf1bfb56fe16ff723627612f9a48f7
            90f3c47a67f870b8
y[5]        1e919d99919c8db48168838cece0abfb
            683da48b9209868b
y[6]        e8ec10c63d8bf80d36498dfc205dc45d
            0dd870572d6d8f1d
y[7]        90177cf5137b8bbf7bcb67a46f86f26c
            fa5a44cbcaa4e18d
y[8]        a099a98b0b3f96d5ac8ac375d8da2a7c
            248004ba11d7ac77
y[9]        5b9218359cddab4cf8ccc6d54cb7e1b3
            5a36ddc9265c0870
y[10]       63d2fc6742a7177876476a324b03295b
            fed99f2eaf1f3897
y[11]       0583c1b2b616aad0f31cd7a4b1bb0a51
            e477e94a01bbb4d6
y[12]       f8866e2528a159df3d6ce244d2b6518d
            1f0212285a3c2d4a
y[13]       927054a1e1620b5b02aab0c8c10ed48a
            e518ea73cba81fcf
y[14]       ff88bff461dac51e7ab4ca75f47a6259
            d24820b9995792d1
y[15]       39f61ae2a8186ae4e3c9bfe0af2cc717
            f424f41aa67f03fa
y[16]       edb0665115f2067a46843a4cbbd297d5
            e83bc1aafc18d1d0
```

```
   y[17]        3b3d894e8595a6526073f02ab0f08b99
                fd9eb208b59ff631
   y[18]        7e5545e6f9ad5f9c183abd043d5acd6e
                b2dd4da3f02dbc31
   y[19]        67b468720a4b8b92ddfe7960998bb7a0
                ecf2a26a37598299
   y[20]        413f7b2aecd39a30cec527b4d9710c44
                73639022451f50d0
   y[21]        1c0457125da0fa4429c07dad859c846c
                bbd93ab5b91b01bc
   y[22]        770b089cfede6f651e86dd7c15989c8b
                5321dea9ca608c71
   y[23]        fd862323072b827cee7a7e28e4e2b999
                647233c3456944bb
   y[24]        7aef9187c96b3f5b79fb98bc76c3574d
                d06f0e95685e5b3a
   y[25]        ef3a54c4155fe3ad817749629c30adbe
                897c4f4454c86c49
   --------------------------------------------
   LMS type     e0000001                         # LMS_SHA256_N24_H5
   path[0]      e9ca10eaa811b22ae07fb195e3590a33
                4ea64209942fbae3
   path[1]      38d19f152182c807d3c40b189d3fcbea
                942f44682439b191
   path[2]      332d33ae0b761a2a8f984b56b2ac2fd4
                ab08223a69ed1f77
   path[3]      19c7aa7e9eee96504b0e60c6bb5c942d
                695f0493eb25f80a
   path[4]      5871cffd131d0e04ffe5065bc7875e82
                d34b40b69dd9f3c1

   Test Case 2 Private Key for SHAKE256-192

   --------------------------------------------
   (note: procedure in Appendix A of RFC8554 is used)
   SEED         303132333435363738393a3b3c3d3e3f
                4041424344454647
   I            505152535455565758595a5b5c5d5e5f
   --------------------------------------------
   --------------------------------------------
```

Test Case 2 Public Key for SHAKE256-192

```
---------------------------------------------
HSS public key
levels       00000001
--------------------------------------------
LMS type     e000000b                         # LMS_SHAKE256_N24_H5
LMOTS type   e000000c                         # LMOTS_SHAKE256_N24_W8
I            505152535455565758595a5b5c5d5e5f
K            db54a4509901051c01e26d9990e55034
             7986da87924ff0b1
--------------------------------------------
--------------------------------------------
```

Test Case 2 Message for SHAKE256-192

```
--------------------------------------------
Message      54657374206d65737361676520666f72  |Test message for|
             205348414b453235362d3139320a      | SHAKE256-192.|
--------------------------------------------
```

Test Case 2 Signature for SHAKE256-192

```
--------------------------------------------
HSS signature
Nspk         00000000
sig[0]:
--------------------------------------------
LMS signature
q            00000006
--------------------------------------------
LMOTS signature
LMOTS type   e000000c                         # LMOTS_SHAKE256_N24_W8
C            84219da9ce9fffb16edb94527c6d1056
             5587db28062deac4
y[0]         208e62fc4fbe9d85deb3c6bd2c01640a
             ccb387d8a6093d68
y[1]         511234a6a1a50108091c034cb1777e02
             b5df466149a66969
y[2]         a498e4200c0a0c1bf5d100cdb97d2dd4
             0efd3cada278acc5
y[3]         a570071a043956112c6deebd1eb3a7b5
             6f5f6791515a7b5f
y[4]         fddb0ec2d9094bfbc889ea15c3c7b9be
             a953efb75ed648f5
y[5]         35b9acab66a2e9631e426e4e99b733ca
             a6c55963929b77fe
y[6]         c54a7e703d8162e736875cb6a455d4a9
```

```
               015c7a6d8fd5fe75
   y[7]        e402b47036dc3770f4a1dd0a559cb478
               c7fb1726005321be
   y[8]        9d1ac2de94d731ee4ca79cff454c811f
               46d11980909f047b
   y[9]        2005e84b6e15378446b1ca691efe491e
               a98acc9d3c0f785c
   y[10]       aba5e2eb3c306811c240ba2280292382
               7d582639304a1e97
   y[11]       83ba5bc9d69d999a7db8f749770c3c04
               a152856dc726d806
   y[12]       7921465b61b3f847b13b2635a45379e5
               adc6ff58a99b00e6
   y[13]       0ac767f7f30175f9f7a140257e218be3
               07954b1250c9b419
   y[14]       02c4fa7c90d8a592945c66e86a76defc
               b84500b55598a199
   y[15]       0faaa10077c74c94895731585c8f900d
               e1a1c675bd8b0c18
   y[16]       0ebe2b5eb3ef8019ece3e1ea7223eb79
               06a2042b6262b4aa
   y[17]       25c4b8a05f205c8befeef11ceff12825
               08d71bc2a8cfa0a9
   y[18]       9f73f3e3a74bb4b3c0d8ca2abd0e1c2c
               17dafe18b4ee2298
   y[19]       e87bcfb1305b3c069e6d385569a4067e
               d547486dd1a50d6f
   y[20]       4a58aab96e2fa883a9a39e1bd45541ee
               e94efc32faa9a94b
   y[21]       e66dc8538b2dab05aee5efa6b3b2efb3
               fd020fe789477a93
   y[22]       afff9a3e636dbba864a5bffa3e28d13d
               49bb597d94865bde
   y[23]       88c4627f206ab2b465084d6b780666e9
               52f8710efd748bd0
   y[24]       f1ae8f1035087f5028f14affcc5fffe3
               32121ae4f87ac5f1
   y[25]       eac9062608c7d87708f1723f38b23237
               a4edf4b49a5cd3d7
   --------------------------------------------
   LMS type    e000000b                         # LMS_SHAKE256_N24_H5
   path[0]     dd4bdc8f928fb526f6fb7cdb944a7eba
               a7fb05d995b5721a
   path[1]     27096a5007d82f79d063acd434a04e97
               f61552f7f81a9317
   path[2]     b4ec7c87a5ed10c881928fc6ebce6dfc
               e9daae9cc9dba690
   path[3]     7ca9a9dd5f9f573704d5e6cf22a43b04
```

```
               e64c1ffc7e1c442e
   path[4]     cb495ba265f465c56291a902e62a461f
               6dfda232457fad14

   Test Case 3 Private Key for SHAKE256-256

   --------------------------------------------
   (note: procedure in Appendix A of RFC8554 is used)
   SEED        606162636465666768696a6b6c6d6e6f
               707172737475767778797a7b7c7d7e7f
   I           808182838485868788898a8b8c8d8e8f
   --------------------------------------------
   --------------------------------------------

   Test Case 3 Public Key for SHAKE256-256

   --------------------------------------------
   HSS public key
   levels      00000001
   --------------------------------------------
   LMS type    e0000006                          # LMS_SHAKE256_N32_H5
   LMOTS type  e0000008                          # LMOTS_SHAKE256_N32_W8
   I           808182838485868788898a8b8c8d8e8f
   K           9bb7faee411cae806c16a466c3191a8b
               65d0ac31932bbf0c2d07c7a4a36379fe
   --------------------------------------------
   --------------------------------------------

   Test Case 3 Message for SHAKE256-256

   --------------------------------------------
   Message     54657374206d657361676520666f7220  |Test mesage for |
               5348414b453235362d3235360a        |SHAKE256-256.|
   --------------------------------------------

   Test Case 2 Signature for SHAKE256-256

   --------------------------------------------
   HSS signature
   Nspk        00000000
   sig[0]:
   --------------------------------------------
   LMS signature
   q           00000007
   --------------------------------------------
   LMOTS signature
   LMOTS type  e0000008                          # LMOTS_SHAKE256_N32_W8
   C           b82709f0f00e83759190996233d1ee4f
```

```
                 4ec50534473c02ffa145e8ca2874e32b
   y[0]          16b228118c62b96c9c77678b33183730
                 debaade8fe607f05c6697bc971519a34
   y[1]          1d69c00129680b67e75b3bd7d8aa5c8b
                 71f02669d177a2a0eea896dcd1660f16
   y[2]          864b302ff321f9c4b8354408d0676050
                 4f768ebd4e545a9b0ac058c575078e6c
   y[3]          1403160fb45450d61a9c8c81f6bd69bd
                 fa26a16e12a265baf79e9e233eb71af6
   y[4]          34ecc66dc88e10c6e0142942d4843f70
                 a0242727bc5a2aabf7b0ec12a99090d8
   y[5]          caeef21303f8ac58b9f200371dc9e41a
                 b956e1a3efed9d4bbb38975b46c28d5f
   y[6]          5b3ed19d847bd0a737177263cbc1a226
                 2d40e80815ee149b6cce2714384c9b7f
   y[7]          ceb3bbcbd25228dda8306536376f8793
                 ecadd6020265dab9075f64c773ef97d0
   y[8]          7352919995b74404cc69a6f3b469445c
                 9286a6b2c9f6dc839be76618f053de76
   y[9]          3da3571ef70f805c9cc54b8e501a98b9
                 8c70785eeb61737eced78b0e380ded4f
   y[10]         769a9d422786def59700eef3278017ba
                 bbe5f9063b468ae0dd61d94f9f99d5cc
   y[11]         36fbec4178d2bda3ad31e1644a2bcce2
                 08d72d50a7637851aa908b94dc437612
   y[12]         0d5beab0fb805e1945c41834dd6085e6
                 db1a3aa78fcb59f62bde68236a10618c
   y[13]         ff123abe64dae8dabb2e84ca705309c2
                 ab986d4f8326ba0642272cb3904eb96f
   y[14]         6f5e3bb8813997881b6a33cac0714e4b
                 5e7a882ad87e141931f97d612b84e903
   y[15]         e773139ae377f5ba19ac86198d485fca
                 97742568f6ff758120a89bf19059b8a6
   y[16]         bfe2d86b12778164436ab2659ba86676
                 7fcc435584125fb7924201ee67b535da
   y[17]         f72c5cb31f5a0b1d926324c26e67d4c3
                 836e301aa09bae8fb3f91f1622b1818c
   y[18]         cf440f52ca9b5b9b99aba8a6754aae2b
                 967c4954fa85298ad9b1e74f27a46127
   y[19]         c36131c8991f0cc2ba57a15d35c91cf8
                 bc48e8e20d625af4e85d8f9402ec44af
   y[20]         bd4792b924b839332a64788a7701a300
                 94b9ec4b9f4b648f168bf457fbb3c959
   y[21]         4fa87920b645e42aa2fecc9e21e000ca
                 7d3ff914e15c40a8bc533129a7fd3952
   y[22]         9376430f355aaf96a0a13d13f2419141
                 b3cc25843e8c90d0e551a355dd90ad77
   y[23]         0ea7255214ce11238605de2f000d2001
```

```
               04d0c3a3e35ae64ea10a3eff37ac7e95
   y[24]       49217cdf52f307172e2f6c7a2a4543e1
               4314036525b1ad53eeaddf0e24b1f369
   y[25]       14ed22483f2889f61e62b6fb78f5645b
               dbb02c9e5bf97db7a0004e87c2a55399
   y[26]       b61958786c97bd52fa199c27f6bb4d68
               c4907933562755bfec5d4fb52f06c289
   y[27]       d6e852cf6bc773ffd4c07ee2d6cc55f5
               7edcfbc8e8692a49ad47a121fe3c1b16
   y[28]       cab1cc285faf6793ffad7a8c341a49c5
               d2dce7069e464cb90a00b2903648b23c
   y[29]       81a68e21d748a7e7b1df8a593f3894b2
               477e8316947ca725d141135202a9442e
   y[30]       1db33bbd390d2c04401c39b253b78ce2
               97b0e14755e46ec08a146d279c67af70
   y[31]       de256890804d83d6ec5ca3286f1fca9c
               72abf6ef868e7f6eb0fddda1b040ecec
   y[32]       9bbc69e2fd8618e9db3bdb0af13dda06
               c6617e95afa522d6a2552de15324d991
   y[33]       19f55e9af11ae3d5614b564c642dbfec
               6c644198ce80d2433ac8ee738f9d825e
   --------------------------------------------
   LMS type    e0000006                         # LMS_SHAKE256_N32_H5
   path[0]     71d585a35c3a908379f4072d070311db
               5d65b242b714bc5a756ba5e228abfa0d
   path[1]     1329978a05d5e815cf4d74c1e547ec4a
               a3ca956ae927df8b29fb9fab3917a7a4
   path[2]     ae61ba57e5342e9db12caf6f6dbc5253
               de5268d4b0c4ce4ebe6852f012b162fc
   path[3]     1c12b9ffc3bcb1d3ac8589777655e22c
               d9b99ff1e4346fd0efeaa1da044692e7
   path[4]     ad6bfc337db69849e54411df8920c228
               a2b7762c11e4b1c49efb74486d3931ea
```

Authors' Addresses

Scott Fluhrer
Cisco Systems
170 West Tasman Drive
San Jose, CA
USA

Email: sfluhrer@cisco.com

Quynh Dang
NIST
100 Bureau Drive
Gaithersburg, MD
USA

Email: quynh.dang@nist.gov