strongSec GmbH
strong internet security

IKEv2 Signature Authentication using ML-DSA

Andreas Steffen
andreas.steffen@strongswan.org

FIPS PQC Signature Standards

- Module-Lattice-Based Digital Signature Standard (ML-DSA)
  FIPS 204, August 2024
  Good performance, simple implementation, moderate public key and signature size.
- Stateless Hash-Based Digital Signature Standard (SLH-DSA)
  FIPS 205, August 2024
  Solid security, signatures are much longer compared with ML-DSA.
- FFT over NTRU-Lattice-Based Digital Signature Standard (FN-DSA)
  FIPS 206, Summer 2025?
  Smaller bandwidth and fast verification but complicated and time-intensive key and signature generation.

Overview by Dustin Moody, NIST, March 2025

PQC Signature Internet Drafts

- Signature Authentication in IKEv2 using PQC
  draft-ietf-ipsecme-ikev2-pqc-auth-02, April 2025
- Internet X.509 PKI: Algorithm Identifiers for ML-DSA
  draft-ietf-lamps-dilithium-certificates-11, May 2025
- Internet X.509 PKI: Algorithm Identifiers for SLH-DSA
  draft-ietf-lamps-x509-slhdsa-09, June 2025
- Post-Quantum Algorithm Guidance
  draft-prabel-pquip-pqc-guidance-00, July 2025

PQC Key and Signature Sizes I

Signature Algorithm	Strength	Private Key	Public Key	Signature
ML-DSA-44	2	32 / 2560	1’312	2’420
ML-DSA-65	3	32 / 4032	1’952	3’309
ML-DSA-87	5	32 / 4896	2’592	4’627
SLH-DSA-SHA2-128s, SLH-DSA-SHAKE-128s	1	64	32	7’856
SLH-DSA-SHA2-128f, SLH-DSA-SHAKE-128f	1	64	32	17’088
SLH-DSA-SHA2-192s, SLH-DSA-SHAKE-192s	3	96	48	16’224
SLH-DSA-SHA2-192f, SLH-DSA-SHAKE-192f	3	96	48	35’664
SLH-DSA-SHA2-256s, SLH-DSA-SHAKE-256s	5	128	64	29’792
SLH-DSA-SHA2-256f, SLH-DSA-SHAKE-256f	5	128	64	49’856

all sizes given in bytes

PQC Key and Signature Sizes II

Signature Algorithm	Strength	Private Key	Public Key	Signature
FN-DSA-512	1	1281	897	752
FN-DSA-1024	5	2305	1793	1462

all sizes given in bytes

- SLH-DSA signatures and X.509 certificates are much larger than ML-DSA ones.
- FN-DSA signatures are more compact but computationally more expensive as ML-DSA ones. Because of the need for floating point operations, FIPS 206 hasn't been finalized yet.
- Thus for the time being the strongSwan Project will prioritize ML-DSA for IKEv2 signature authentication. FN-DSA is an interesting alternative.

strongSwan ML-DSA Implementation for IKEv2

- Currently draft-ietf-ipsecme-ikev2-pqc-auth-02 proposes some alternatives.
- Identity Hash is used as in EdDSA [RFC 8420] (pure mode).
- Authentication Method Announcement [RFC 9593] is not supported.
- By default hedged mode is used for FIPS 204 signatures with a 32 byte random seed provided by an external RNG.
  Savings by switching to deterministic mode don't seem to be worth the additional configuration effort.
- The context string is set to an empty string for FIPS 204 signatures.

Supported Crypto Libraries implementing ML-DSA

strongSec GmbH
strong internet security

Crypto Library	Version	strongSwan Plugin
Botan	3.7.1	botan
OpenSSL	3.5.1	openssl
AWS libcrypto (AWS-LC) *	1.55.0	openssl
wolfSSL	5.8.0	wolfssl
Internal plugin (based on reference source code)	6.1.0	ml

*work in progress

- Current strongSwan ML-DSA implementation can be found in ml-dsa branch: https://github.com/strongswan/strongswan/tree/ml-dsa
- ETA for strongSwan 6.1.0 release with ML-DSA support: Finalization of draft-ietf-ipsecme-ikev2-pqc-auth

ML-DSA Private Key Generation


```
$ pki --gen --type mldsa65 > moonKey.der

$ od -t x1 moonKey.der                                          # PKCS#8-encoded key
0000000 30 34 02 01 00 30 0b 06 09 60 86 48 01 65 03 04     # ML-DSA-65 OID
0000020 03 12 04 22 80 20 25 d4 da ae 50 27 2e 1c 4f da     # 32 byte key seed
0000040 2a 64 8f 4c d4 96 ee 4d 65 b3 97 83 6e fb bc 71
0000060 67 97 7a e4 3a 3b

$ pki --print --type priv --in moonKey.der
  privkey:    ML_DSA_65 15616 bits
  keyid:      3b:c3:99:a4:31:21:99:fa:89:2f:7f:8c:ec:df:83:ba:f6:42:51:f6
  subjkey:    55:ac:86:c3:05:82:79:47:07:2e:40:05:14:97:c4:57:98:a9:9c:db
```

ML-DSA X.509 Certificate Generation


```
$ pki --issue --in moonKey.der --type priv  --lifetime 3652 --flag serverAuth \
      --dn "C=CH, O=strongSwan Project, CN=moon.strongswan.org" --san moon.strongswan.org \
      --cakey strongswanKey.pem --cacert strongswanCert.pem > moonCert.der

$ pki --print --type x509 --in moonCert.der
  subject:  "C=CH, O=strongSwan Project, CN=moon.strongswan.org“
  issuer:   "C=CH, O=strongSwan Project, CN=strongSwan ML-DSA Root CA“
  validity:  not before Jul 09 12:41:18 2025, ok
             not after  Jul 09 12:41:18 2035, ok (expires in 3651 days)
  serial:    3f:51:75:6f:74:a7:97:da
  altNames:  moon.strongswan.org
  flags:     serverAuth
   authkeyId: cd:28:21:bf:6c:f0:9a:5d:0a:3f:57:ea:0d:db:86:ae:e5:40:5e:1c
  subjkeyId: 55:ac:86:c3:05:82:79:47:07:2e:40:05:14:97:c4:57:98:a9:9c:db
  pubkey:    ML_DSA_65 15616 bits
  keyid:     3b:c3:99:a4:31:21:99:fa:89:2f:7f:8c:ec:df:83:ba:f6:42:51:f6
  subjkey:   55:ac:86:c3:05:82:79:47:07:2e:40:05:14:97:c4:57:98:a9:9c:db
```

IKEv2 Negotiation with ML-KEM and ML-DSA I


```
IKE_SA_INIT request  0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) N(FRAG_SUP) N(HASH_ALG)
                         N(REDIR_SUP) N(IKE_INT_SUP) ]

IKE_SA_INIT response 0 [ SA KE No N(NATD_S_IP) N(NATD_D_IP) CERTREQ N(FRAG_SUP) N(HASH_ALG)
                         N(CHDLESS_SUP) N(IKE_INT_SUP) N(MULT_AUTH) ]

IKE_INTERMEDIATE request  1 [ KE ]
splitting  IKE message (1264 bytes) into 2 fragments
IKE_INTERMEDIATE request  1 [ EF(1/2) ]
IKE_INTERMEDIATE request  1 [ EF(2/2) ]

IKE_INTERMEDIATE response 1 [ KE ]
```

- Hybrid key exchange with CURVE_25519 / ML_KEM_512
- Default strongSwan IPv4 fragment size of 1280 bytes

IKEv2 Negotiation with ML-KEM and ML-DSA II


```
IKE_AUTH request 2 [ IDi CERT N(INIT_CONTACT) CERTREQ IDr AUTH SA TSi TSr N(MOBIKE_SUP)
                     N(ADD_6_ADDR) N(MULT_AUTH) N(EAP_ONLY) N(MSG_ID_SYN_SUP) ]
splitting  IKE message (9120 bytes) into 8 fragments
IKE_AUTH request 2 [ EF(1/8) ]
IKE_AUTH request 2 [ EF(2/8) ]
IKE_AUTH request 2 [ EF(3/8) ]
IKE_AUTH request 2 [ EF(4/8) ]
IKE_AUTH request 2 [ EF(5/8) ]
IKE_AUTH request 2 [ EF(6/8) ]
IKE_AUTH request 2 [ EF(7/8) ]
IKE_AUTH request 2 [ EF(8/8) ]
```

- CERT: ML-DSA-44 public key with ML-DSA-87 signature
- AUTH: ML-DSA-44 signature

IKEv2 Negotiation with ML-KEM and ML-DSA III


```
IKE_AUTH response 2 [ EF(1/10) ]
IKE_AUTH response 2 [ EF(2/10) ]
IKE_AUTH response 2 [ EF(3/10) ]
IKE_AUTH response 2 [ EF(4/10) ]
IKE_AUTH response 2 [ EF(5/10) ]
IKE_AUTH response 2 [ EF(6/10) ]
IKE_AUTH response 2 [ EF(7/10) ]
IKE_AUTH response 2 [ EF(8/10) ]
IKE_AUTH response 2 [ EF(9/10) ]
IKE_AUTH response 2 [ EF(10/10) ]
received fragment #10 of 10, reassembled fragmented IKE message (10592 bytes)
IKE_AUTH response 2 [ IDr CERT AUTH SA TSi TSr N(MOBIKE_SUP) N(ADD_4_ADDR) N(ADD_6_ADDR)
                      N(ADD_6_ADDR) ]
```

- CERT: ML-DSA-65 public key with ML-DSA-87 signature
- AUTH: ML-DSA-65 signature

Thank you for your attention!

Questions?