# Composite Signature –v13 Updates

Mike Ounsworth, John Gray,
Jan Klaussner, Max Pala

March 2024

ENTRUST

SECURING A WORLD IN MOTION

# Changes affecting Interoperability

- Changed the content of the OID artifact from ASCII String to DER encoded OID
  - Included a simple table of the OID encodings in the draft
- Simplified the number of pre-hashing algorithm required by removing SHA384 and SHAKE and replacing those with SHA512.  Now only SHA256 and SHA512 are needed.
- Updated the prototype OIDs since the changes in this draft are not compatible with version -10
- Removed Falcon (FN-DSA) composites to not delay the release of this document. Additional composites can be added in separate documents as needed.



ENTRUST

# Composite Signature Generation Process

1. Compute a hash of the Message

   M' = HASH(Message)

2. Generate the n component signatures independently, according to their algorithm specifications.

   $S1 := Sign_{A1}( K1,$ DER (OID) $||$ M' )

   $S2 := Sign_{A2}( K2,$ DER (OID) $||$ M' )

3. Encode each component signature S1 and S2 into a BIT STRING according to its algorithm specification.

   signature ::= SEQUENCE { S1, S2 }

4. Output signature

Where:

K1, K1 are signing private keys for each component algorithm

A1. A2 are the component signing algorithms (for example ML-DSA, RSA, ECDSA)



ENTRUST

# Composite Signature Verification Process

1. Check keys, signatures, and algorithms lists for consistency.

2. Compute a Hash of the Message

    M' = HASH(Message)

3. Check each component signature individually, according to its algorithm specification. If any fail, then the entire signature validation fails.

    if not $\text{Verify}_{A1}$( P1, DER (OID) || M', S1) then output "Invalid signature"

    if not $\text{Verify}_{A2}$( P2, DER (OID) || M', S2) then output "Invalid signature"

    output "Valid signature"

***"applications MUST output "Valid signature" (true) if and only if all component signatures were successfully validated, and "Invalid signature" (false) otherwise."***



ENTRUST

# ML-DSA Composite Signatures

| Composite Signature Algorithm | First Algorithm | Second Algorithm | Pre-Hash |
|---|---|---|---|
| id-MLDSA44-RSA2048-PSS-SHA256 | MLDSA44 | SHA256WithRSAPSS | SHA256 |
| id-MLDSA44-RSA2048-PKCS15-SHA256 | MLDSA44 | SHA256WithRSAEncryption | SHA256 |
| id-MLDSA44-Ed25519-SHA512 | MLDSA44 | Ed25519 | SHA512 |
| id-MLDSA44-ECDSA-P256-SHA256 | MLDSA44 | SHA256withECDSA | SHA256 |
| id-MLDSA44-ECDSA-brainpoolP256r1-SHA256 | MLDSA44 | SHA256withECDSA | SHA256 |
| id-MLDSA65-RSA3072-PSS-SHA512 | MLDSA65 | SHA512WithRSAPSS | SHA512 |
| id-MLDSA65-RSA3072-PKCS15-SHA512 | MLDSA65 | SHA512WithRSAEncryption | SHA512 |
| id-MLDSA65-ECDSA-P256-SHA512 | MLDSA65 | SHA512withECDSA | SHA512 |
| id-MLDSA65-ECDSA-brainpoolP256r1-SHA512 | MLDSA65 | SHA512withECDSA | SHA512 |
| id-MLDSA65-Ed25519-SHA512 | MLDSA65 | Ed25519 | SHA512 |
| id-MLDSA87-ECDSA-P384-SHA512 | MLDSA87 | SHA512withECDSA | SHA512 |
| id-MLDSA87-ECDSA-brainpoolP384r1-SHA512 | MLDSA87 | SHA512withECDSA | SHA512 |
| id-MLDSA87-Ed448-SHA512 | MLDSA87 | Ed448 | SHA512 |



ENTRUST

# Compact Composite (v10+) Public Key and Cert Savings

| Algorithm | Public Key | Certificate |
|---|---|---|
| id-MLDSA44-ECDSA-P256-SHA256 | 41 bytes | 94 bytes |
| Id-MLDSA44-ECDSA-brainpoolP256r1-SHA256 | 42 bytes | 94 bytes |
| id-MLDSA65-ECDSA-P256-SHA512 | 41 bytes | 94 bytes |
| id-MLDSA65-RSA3072PKCS15-SHA512 | 37 bytes | 91 bytes |
| id-MLDSA87-ECDSA-P384-SHA512 | 38 bytes | 90 bytes |
| Id-MLDSA65-RSA3072-PSS-SHA512 | 37 bytes | 91 bytes |

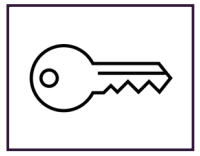
Savings come from removal of parameters and redundant AlgorithmID

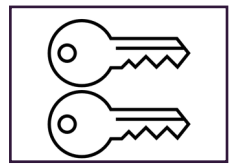


ENTRUST

# Subject Public key size in comparison to ML-DSA public key
## (in bytes)

| Algorithm | ML-DSA only | ML-DSA Composite | Difference |
|---|---|---|---|
| id-MLDSA44-ECDSA-P256-SHA256 | 1312+24 | 1413 | 77 (5.8%) |
| Id-MLDSA44-ECDSA-brainpoolP256r1-SHA256 | 1312+24 | 1413 | 77 (5.8%) |
| Id-MLDSA44-RSA2048-PSS-SHA256 | 1312+24 | 1620 | 284 (21.25%) |
| id-MLDSA65-ECDSA-P256-SHA512 | 1952+24 | 2053 | 77 (3.9%) |
| id-MLDSA65-RSA3072-PKCS15-SHA512 | 1952+24 | 2388 | 412 (20.9%) |
| Id-MLDSA65-RSA3072-PSS-SHA512 | 1952+24 | 2388 | 412 (20.9%) |
| id-MLDSA87-ECDSA-P384-SHA512 | 2592+24 | 2725 | 109 (4.2%) |
| id-MLDSA87-ECDSA-brainpoolP384r1-SHA512 | 2592+24 | 2725 | 109 (4.2%) |







ENTRUST

# Cert size in comparison to ML-DSA Certificate
## (in bytes)

| Algorithm | ML-DSA cert only | ML-DSA Composite cert | Difference |
|---|---|---|---|
| id-MLDSA44-ECDSA-P256-SHA256 | 4114 | 4274 | 160 (3.9%) |
| Id-MLDSA44-ECDSA-brainpoolP256r1-SHA256 | 4114 | 4274 | 160 (3.9%) |
| Id-MLDSA44-RSA2048-PSS-SHA256 | 4114 | 4668 | 554 (13.5%) |
| id-MLDSA65-ECDSA-P256-SHA512 | 5627 | 5786 | 159 (5.2%) |
| id-MLDSA65-RSA3072-PKCS15-SHA512 | 5627 | 6437 | 810 (14.4%) |
| Id-MLDSA65-RSA3072-PSS-SHA512 | 5627 | 6437 | 810 (14.4%) |
| id-MLDSA87-ECDSA-P384-SHA512 | 7569 | 7794 | 225 (3.0%) |
| id-MLDSA87-ECDSA-brainpoolP384r1-SHA512 | 7569 | 7793 | 224 (3.0 %) |

These are all still smaller than the smallest SLH-DSA which is **8251** bytes!



ENTRUST

# Editorial Changes

- Renamed draft to “Composite ML-DSA for use in Internet PKI”
  - Falcon (FN-DSA) will not be standardized for a few years, we don’t want to stall this draft
- Remove ambiguity and made it clear that all component signature MUST be verified
- Added language to ensure that component keys MUST not be used in any other context
- Fixed the ASN.1 module pk-CompositeSignature so it now compiles
- Shortened abstract and added text to justify where and why this mechanism would be used.
- Resolved comments from Kris Kwiatkowski, Tim Hollebeek and Falko Strenzke
- Added a security consideration about Trust Anchors
- Updated the included samples to conform to this draft
- fixed nits



ENTRUST

# Comments on v13

- Kris Kwiatkowski had some questions regarding the section on FIPS in Appendix B1 which the group will address in the next update.
- Kris also noticed that Falcon has not been completely removed from the ASN.1 module… A typo in the ASN.1 module as “Falon” foiled the authors!



ENTRUST

# Call for Adoption

- We have gone through the comments from the failed call for adoption pre IETF 117 and addressed all the comments from IETF 118
  - We believe we have addressed all objections
    - Clarified scope and added justification text from jurisdictions which are intending to use composite signatures in the document introduction
    - Collected additional use-cases and support testimonials which are specified in **draft-vaira-pquip-pqc-use-cases-00**
    - Removed Falcon to focus on ML-DSA composites once ML-DSA is standardized this year.
    - Make use of the concatenation non-separability property which is discussed further in (**draft-hale-pquip-hybrid-signature-spectrums-00**)
    - Implementations exist in openSSL, bouncy castle (Jan Oupicky from University of Luxembourg), CryptoNext provider, Digicert provider, Botan, Entrust and others.  Most of them updated to v13 in the PQ hackathon this past weekend!



ENTRUST

# Thank You

entrust.com