COMPOSITE CRYPTO FOR PKIX AND CMS

IETF LAMPS 113

Mike Ounsworth, John Gray, Serge Mister (Entrust), Max Pala (CableLabs), Jan Klaussner, Klaus-Dieter Wirth (D-Trust).
Outline

» Status of composite drafts

» Changes coming in composite-keys-02
  ◦ Generic and Explicit now share a wire encoding
  ◦ Combiner Modes in alg params
  ◦ Added “K of N” mode

» Changes coming in composite-kem-00
  ◦ Defines a KEM algorithm which composes arbitrary KeyEx, KeyTrans, KEM components.

» Terminology:
  ◦ “Hybrid” is problematic
  ◦ We need a hero to volunteer for an Informative terminology draft.
Composite at IETF LAMPS (current drafts)

**Drafts**

- **draft-ounsworth-pq-composite-keys-01**
  - Defines composite public and private keys

- **draft-ounsworth-pq-explicit-composite-keys-01**
  - Defines a structure for defining explicit pairs of algorithms
  - Ex.: RsaAndDilithium

- **draft-ounsworth-pq-composite-sigs-06**
  - Defines composite signatures

- **draft-ounsworth-pq-composite-encryption-01**
  - Defines composite encryption using EnvelopedData
**Composite at IETF LAMPS (upcoming drafts)**

- **draft-ounsworth-pq-composite-keys-02**
  - Defines composite public and private keys
  - -02 is coming *very soon*
  - Working copy:
    - https://github.com/EntrustCorporation/draft-ounsworth-pq-composite-keys

- **draft-ounsworth-pq-composite-sigs-06**
  - Defines composite signatures
  - Mature and stable
  - Needs minor work to sync with new modes added to keys-02

- **draft-ounsworth-pq-composite-encryption-01**
  - Defines composite encryption using EnvelopedData
  - Replaced by kem-00

- **draft-ounsworth-pq-composite-kem-00**
  - NEW (not yet published)
  - Defines composite as a KEM (Key Encapsulation Mechanism)
  - Useable anywhere that accepts KEMs.

---

**Notes:**

- *Almost* ready for WG Adoption.

- Merged into and deprecated by keys-02.
Composite-Keys: Changes from -01 to -02

- **Backwards compatible**: keys produced under -01 (mostly) still parse with the same semantics under -02.

- **Working copy in github**¹

- **Additions**:
  - Merged with Intelligent Composed Algorithms spec from D-Trust².
  - Generic and Explicit now share a wire encoding; only differ in the OID used.
  - Combiner Modes (AND, OR, ANY, K of N, Custom) now specified via alg params (breaking change for modes other than AND).
    (please don’t hate me Russ, we tried so hard to avoid params >.<)
  - Differentiated between OR, ANY, and Custom modes.
  - Added “K of N” mode.

1: https://github.com/EntrustCorporation/draft-ounsworth-pq-composite-keys
Generic and Explicit now share a wire encoding

**Wire encoding (s. 2.2, 2.3)**

```
CompositePublicKey ::= SEQUENCE SIZE (2..MAX) OF SubjectPublicKeyInfo

CompositePrivateKey ::= SEQUENCE SIZE (2..MAX) OF OneAsymmetricKey
```

**Generic (s. 2.5.1)**

Top-level AlgorithmIdentifier is: `id-composite-key` OBJECT IDENTIFIER

Component algs may use any algorithm.

**Explicit (s. 2.5.2)**

Top-level AlgorithmIdentifier is defined by explicit algorithm.

Component algs MUST use the algorithms defined by the explicit algorithm.
Generic and Explicit examples

ASN.1 decoding of the same \{EC, RSA\} pair in Generic and Explicit modes. Identical except for OIDs.

**id-composite-key** is defined in the draft.

**Assume:** \texttt{id-pk-example-ECandRSA} OBJECT IDENTIFIER ::= \{ 1 2 3 4 \}

**Generic**

```
algorithm: AlgorithmIdentifier\{id-composite-key\}
params: CompositeParams\{id-composite-or\}

subjectPublicKey: CompositePublicKey {
    SubjectPublicKeyInfo {
        algorithm: AlgorithmIdentifier {
            algorithm: \texttt{ecPublicKey}
            parameters: prime256v1
        }
        subjectPublicKey: <ec key octet string>
    },
    SubjectPublicKeyInfo {
        algorithm: AlgorithmIdentifier {
            algorithm: \texttt{rsaEncryption}
            parameters: NULL
        }
        subjectPublicKey: <rsa key octet string>
    }
}
```

**Explicit**

```
algorithm: AlgorithmIdentifier\{id-pk-example-ECandRSA\}
params: CompositeParams\{id-composite-or\}

<same as Generic>
```
Combiner mode is now specified via algorithm params (s. 2.2.1, 2.6)

2.2.1. CompositeParams

CompositeParams ::= AlgorithmIdentifier

where the algorithm and parameters represent a combiner mode as defined in Section 2.6.

Section “2.6 Combiner Modes” defines OIDs for:

- **AND**: implicit by ABSENT params
- **OR**: id-composite-or
  - Params ABSENT
- **ANY**: id-composite-any
  - Params ABSENT
- **K of N**: id-composite-k-of-n
  - CompositeKofNParams ::= INTEGER
- **Custom**
  - Allows implementors to define their own mode under their own OID.
K of N Mode – semantics

- K=N and K=1 are equivalent to AND and OR respectively
  - AND / OR modes should be used instead to improve interop.

- Exact semantics to be defined when we work on composite sigs and composite kem / encryption drafts.
  - Proposal: signer / encryptor chooses k of the n pub keys (or more??).
    Proceed as in AND mode with the chosen k keys; leave NULLs for the other sig / ciphertext values.

```plaintext
AlgorithmIdentifier {
  algId : id-composite-key
  params : AlgorithmIdentifier {
    algId : id-composite-k-of-n
    params: 3
  }
}
```

You get n by counting the components
Combining Security of AND mode with flexibility of OR mode

- Security of PQCs is uncertain in near future

→ PKIs with 1 traditional, 2 PQC

  - Traditional algorithm shall be kept due to known security
  - If 1st PQC is broken, 2nd can be used without reissuing certificate
Composite KEM

- Defines a KEM algorithm which internally combines two or more Key Ex, Key Trans, KEM component algorithms.

- Follows the RSA-KEM pattern from RFC 5990 + NIST SP 800-56Cr2 KDF combiner.

```plaintext
ForEach component pub key PKi:
  if (isKeyExOrKEM(PKi):
    SSi, CTi := encaps(PKi);
  if (isKeyTrans(PKi):
    SSi := random_bits(SIZE);
    CTi := encrypt(SSi, PKi);

SS := KDF( SS1 || SS2 || ... )
Transmit: CT1, CT2, ...
```

- Open questions:
  1. Is this better presented as a KEM where SS is derived (as shown here), or as a KeyTrans by including an AES_WRAP() step of a provided content encryption key?
     - Perret & Prat are working on a generic KEM-TRANS structure for CMS\(^1\) which does the AES_WRAP(), so we have opted to present a KEM and fit in their draft.

  2. Cryptographic review of combiner, and security considerations around choice of algorithms.
     - See maillist discussion “Re: draft-ounsworth-pq-composite-encryption key combination”

  3. Synchronize with TLS WG hybrid KEMs.

\(^1\): draft-perret-prat-cms-pq-kem-00
“Hybrid” and “dual” are the NIST terms for multi-algorithm key exchange and signatures, respectively.

“Hybrid” is problematic because it collides with

https://en.wikipedia.org/wiki/Hybrid_cryptosystem

- Especially with “hybrid” CMS content encryption aligned with draft-ietf-tls-hybrid-design.

Multiple terms get used with confusing overlap:

- Hybrid / dual / composite / multi-certificate / multi-key

We need a hero to volunteer for an Informative terminology draft.