Composite Crypto
Composite Signatures and Keys for X.509 and CMS
**Composite Signatures**

**What?**

- Address quantum timeline uncertainty by extending public keys and signatures to have 2 or more “component” algorithms.

- Automatically applies to X.509, CMS, and any protocol that uses “ASN.1-based” signatures.

---

**Certificate**

Subject: “Joe”

... PubKey: Composite

{RSA2048 key, Dilithium key}

... SigAlg: Composite

{RSA4096, SPHINCS+}

Sig: {10111010100..., 011010011010...}

---

**Document**

SigAlg: {RSA2048, Dilithium}

SigValue: {111001..., 100010...}

---

**Signatures:**

- Sign_RSA(``)
- Sign_PQ(``)

---

**Alice**

---

**Bob**
Composite Signatures
Why do it this way?

General

- **Simplicity**: list of SPKI / Signature, so inherits all flexibility of alg / param selection (for ex. vs pairwise alg OIDs).

- **Simplicity / Sec**: Alg:Composite” means that the “multiple-signature” logic is handled by crypto library, not protocol or application layer; harder for everyday programmers to get it wrong.

vs Multiple certs

- **Simplicity**: Fits into existing pubkey / sig fields in (any?) existing protocol.

- Binds multiple PubKeys / SigValues into one object.
  - **Sec**: easier to analyze, ex.: alg / key substitution attacks.
  - **Sec**: All component keys revoked together.
  - **Ops**: Still a single cert / private key to manage.
  - **Sec / Ops**: Single PKI chain/root.

Cert size

- **Objection**: “PQ algs will blow certs up to ~50 kb!!!”
  - This is unavoidable.
  - Solutions to this problem (ex.: certs contain hashes of key / sig data) would probably be made to the SPKI / SigValue objects, and therefore are orthogonal to this draft.
Composite Signatures
Open Design Questions

Verifier behavior for
Unsupported and deprecated algs?
What if a client doesn’t recognize a component AlgID?

What if RSA is deprecated, but is present as a component key?
• In single-key crypto, you reject.
• Desired behaviour in composite:
  proceed so long as “there are enough good algs left”.
• Implementation is tricky.

Key Revocation:
• Desired behaviour in composite:
  If any component key is revoked, the entire composite key / cert is revoked.
• Security Consideration:
  Does each component key need to be checked individually for previous compromise?

Key Usage:
• This draft only covers signatures; we leave encryption keys as a future work.
• This draft applies the same KeyUsage to all component keys. “Dual-usage” or other kinds of non-homogenous KeyUsages are attractive, but makes security analysis very complex.
Composite Signatures
Implementation Gotchas

“Intrinsic” Message Digests

• Some sig algs (ex. RSA) expect to be given a digest to sign, while some have an intrinsic hash (ex. EdDSA) and expect to be given a full message.

• Some crypto libs will need re-architecture to do message digesting at sig verification layer, and not higher in the call stack.

Alg Parameters

• Currently, the AlgID inside the PUBLIC-KEY structure says “I’m Composite” rather than “I’m Composite with RSA-4096, EdDSA, and SPHINCS” (ie absent PARAMS) which means the AlgID by itself carries almost no information. Will that cause problems for any protocols?

• The sa-CompositeSignature SIGNATURE-ALGORITHM structure uses the PARAMS field to list component algs. RSASSA-PSS is the only existing alg that uses SigAlg PARAMS. Some implementations hard-code RSA-PSS as an exception and may not have generic support for SigAlg PARAMS.