LAMPS PQC INTERIM

Mike Ounsworth, Serge Mister, John Gray

Sept 13, 2021

ENTRUST

SECURING A WORLD IN MOTION

Outline

- LAMPS PQC Milestones
  - PQC KEMs
  - Hybrid key establishment
  - PQC Signatures
  - Dual signatures
- Adoption candidates: Composite
  - draft-ounsworth-pq-composite-keys
  - draft-ounsworth-pq-explicit-composite-keys
  - draft-ounsworth-pq-composite-encryption
  - draft-ounsworth-pq-composite-sigs

LAMPS PQC MILESTONES

ENTRUST

SECURING A WORLD IN MOTION

LAMPS PQC Milestones

Date	Milestone
Oct 2021	Adopt draft for PQC KEM algorithms in CMS
Oct 2021	Adopt draft for PQC KEM public keys in PKIX certificates
Dec 2021	Adopt draft for PQC signatures in CMS
Dec 2021	Adopt draft for PQC signatures in PKIX certificates
Dec 2021	Adopt draft for hybrid key establishment in CMS
Dec 2021	Adopt draft for public keys for hybrid key establishment in PKIX certificates
Dec 2021	Adopt draft for dual signature in CMS
Dec 2021	Adopt draft for dual signatures in PKIX certificates

Milestones -- PQC KEMs

Date	Milestone
Oct 2021	Adopt draft for PQC KEM algorithms in CMS
Oct 2021	Adopt draft for PQC KEM public keys in PKIX certificates

Tasks:

- Does LAMPS need to wait for CFRG specifications for the new PQC algorithms?
- LAMPS WG needs to reference OIDs and define ASN.1 encodings.
- Do KEMs even fit in CMS today (KeyTransRecipientInfo, KeyAgreeRecipientInfo, KEKRecipientInfo, PasswordRecipientInfo, OtherRecipientInfo)? We probably need to define a KEMRecipientInfo that handles wrapping / unwrapping of KEMs.

Status:

- No submitted drafts that I'm aware of.
- How to proceed? .. adopt a more or less empty --00 document that can grow as PQC algorithms meet maturity?
- We need volunteer author(s).

Next steps:

- Present candidate drafts at 112 and vote on adoption? Is 112 too late to meet the milestone?

Milestones – Hybrid key establishment

Date	Milestone
Dec 2021	Adopt draft for hybrid key establishment in CMS
Dec 2021	Adopt draft for public keys for hybrid key establishment in PKIX certificates

Tasks:

- LAMPS will get PQC KEMs through milestones #1,2. This work is to combine them into hybrid modes.
- Decide on what approach LAMPS wants to proceed with.
  - Multi-cert
  - Some form of multi-key cert
    - Composite keys, key exchange, and content encryption.
    - Other?

Status:

- Submitted drafts:
  - draft-ounsworth-pq-composite-keys-00
  - draft-ounsworth-pq-explicit-composite-keys-00
  - draft-ounsworth-pq-composite-encryption-00
  - Needs LAMPS and probably CFRG review on security of combiners.

Next steps:

- Present candidate drafts at 112 and vote on adoption?

Milestones – PQC signatures

Date	Milestone
Dec 2021	Adopt draft for PQC signatures in CMS
Dec 2021	Adopt draft for PQC signatures in PKIX certificates

Tasks:

- Does LAMPS need to wait for CFRG specifications for the new PQC algorithms?
- LAMPS WG needs to reference OIDs and define ASN.1 encodings.

Status:

- No submitted drafts that I'm aware of.
- How to proceed? .. adopt a more or less empty --00 document that can grow as PQC algorithms meet maturity?
- We need volunteer author(s).

Next steps:

- Present candidate drafts at 112 and vote on adoption?

Milestones – Dual signatures

Date	Milestone
Dec 2021	Adopt draft for dual signature in CMS
Dec 2021	Adopt draft for dual signatures in PKIX certificates

Tasks:

- LAMPS will get PQC sig algs through milestones #3, 4. This work is to combine them into dual modes.
- Decide on what approach LAMPS wants to proceed with.
  - Multi-cert
  - Some form of multi-key cert -- Composite keys and signatures? Other?

Status:

- ITU-T X.510 has defined an approach to what we call “composite OR” which gives migratability but not strengthened crypto.
  - Note this uses ASN.1 “extension marks”, a language feature that is not compatible with the ASN.1 modules in 5280.
- Submitted drafts:
  - draft-ounsworth-pq-composite-keys-00
  - draft-ounsworth-pq-explicit-composite-keys-00
  - draft-ounsworth-pq-composite-sigs-05

Next steps:

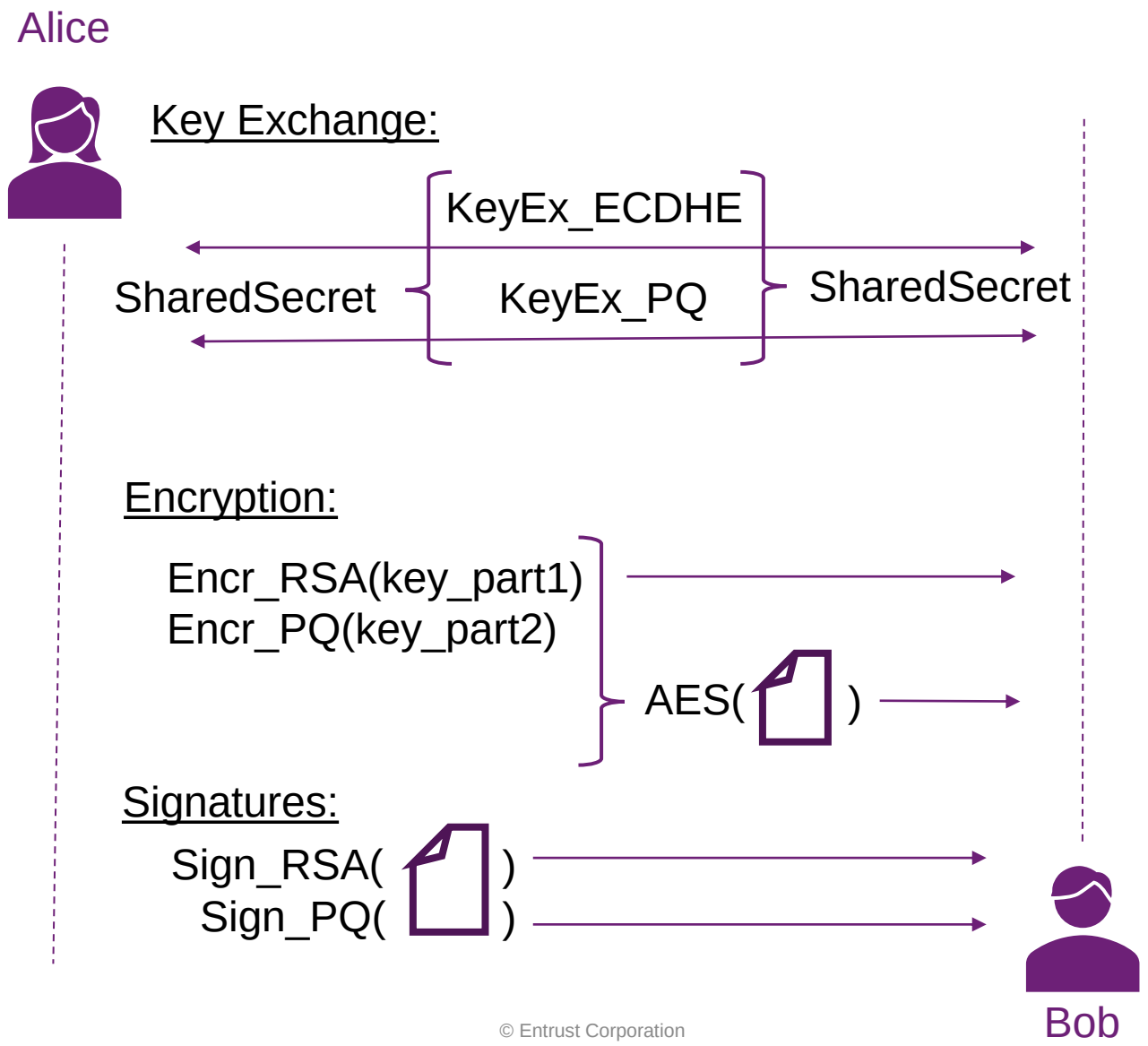
- Present candidate drafts at 112 and vote on adoption?

ADOPTION CANDIDATES

Composite keys, signatures, and encryption

ENTRUST
SECURING A WORLD IN MOTION

Hybrid and Dual crypto modes



Overview of Composite Keys drafts – generic composite

Designed for compatibility with any protocol that requires an ASN.1-encoded key.

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

```
pk-Composite PUBLIC-KEY ::= {
        IDENTIFIER id-composite-key
        KEY CompositePublicKey
        PARAMS ARE absent
        PRIVATE-KEY CompositePrivateKey
}
```

```
pk-Composite-or PUBLIC-KEY ::= {
        IDENTIFIER id-composite-or-key
        KEY CompositePublicKey
        PARAMS ARE absent
        PRIVATE-KEY CompositePrivateKey
}
```

Can be used with one of two OIDs specifying how the client must use the key:

- AND mode: id-composite-key
- OR mode: id-composite-or-key

Overview of Composite Keys drafts – explicit composite

```
sa-explicitCompositeSignatureAlgorithm {
   OBJECT IDENTIFIER:algId,
   SIGNATURE-ALGORITHM:firstAlg,
   PUBLIC-KEY:firstPublicKey,
   FirstPublicKeyType,
   SIGNATURE-ALGORITHM:secondAlg,
   PUBLIC-KEY:secondPublicKey,
   SecondPublicKeyType} SIGNATURE-ALGORITHM ::= {
       ...
}
```

Ex.:

```
sa-entrust-sha256RSAandECDSA SIGNATURE-ALGORITHM ::=
  sa-explicitCompositeSignatureAlgorithm {
     id-sa-entrust-sha256RSAandECDSA,
     sa-sha256WithRSAEncryption,
     pk-rsa,
     RSAPublicKey,
     sa-ecdsaWithSHA256,
     pk-ec,
     ECPoint
}
```

Explicit vs Generic composite

- Generic: much flexibility
- Explicit: much like TLS cipher suites, multi-algorithm complexity is “hidden” inside a single OID

- Question1: would applications prefer the more flexible or more rigid structures? (or both?)
- Question2: if explicit, should LAMPS specify OIDs for alg pairs / tuples?
  - PKIX / CMS has not historically specified mandatory-to-implement algorithms; does this use-case justify it?

Overview of Composite Signatures draft

Designed for compatibility with any protocol that requires an ASN.1-encoded signature.

```
sa-CompositeSignature SIGNATURE-ALGORITHM ::= {
       IDENTIFIER id-alg-composite
       VALUE CompositeSignatureValue
       PARAMS TYPE CompositeParams ARE required
       PUBLIC-KEYS { pk-Composite }
       SMIME-CAPS { IDENTIFIED BY id-alg-composite } }
   }
CompositeParams ::= SEQUENCE SIZE (2..MAX) OF AlgorithmIdentifier
CompositeSignatureValue ::= SEQUENCE SIZE (2..MAX) OF BIT STRING
```

Draft provides generation and verification processes.

Can be used with one of two OIDs specifying how the client must use validate the signature:

- AND mode: id-alg-composite
  - Strict “all must be present and valid” mode.
- OR mode: id-alg-composite-or
  - Allows generator to omit signatures at generation time, for example if client does not have an implementation of one of its algorithms.
  - Allows verifier to stop after one successful verification.
  - Allows any other custom behaviour.

Overview of Composite Encryption / Key Exchange draft

Designed for compatibility with CMS' KeyTransRecipientInfo and KeyAgreeRecipientInfo.

Defines three mechanisms:

1. Composite Key Transport using Encryption primitives
   - Supports only Encryption primitives. Compatible with CMS' KeyTransRecipientInfo.
   - Given n keypairs: one-time-pad encryption of provided CEK under n-1 one-time-pad keys for a total of n "key shares", each encrypted for one recipient pub key. All n "key shares" must be decrypted in order to recover the CEK.
2. Composite Key Transport using Encryption and KEM primitives
   - Generalization of (1) to support combinations of 0 or more KEM and 1 or more Encryption primitives.
   - Could alternatively have designed it to allow arbitrary combinations of KEM and Encr, but then either CEK would have been output of KEMs rather than a provided input, breaking compatibility with CMS KeyTrans API, or would have required to transmit n+1 ciphertexts (n encrypted one-time-pad keys + XOR ciphertext).
   - If design converges, could be rolled into (1).
3. Composite Key Exchange
   - Most generic mode: supports arbitrary combination of KeyEx, KEM, Encryption primitives. Compatible with CMS' KeyAgreeRecipientInfo -- outputs a shared secret rather than taking CEK as input.
   - Uses NIST SP 800-56Cr2 combiner.

All modes support both AND and OR in a similar way to Sigs draft.

Open questions for Composite drafts

1. We have submitted draft-ounsworth-pq-composite-keys and draft-ounsworth-pq-explicit-composite-keys, which can be viewed as competing drafts – do we want one, or the other, or both, or a combination approach as standards?

2. We would love WG feedback on whether generic composite (IE SEQUENCE SIZE 2..MAX OF SubjectPublicKeyInfo), or explicit pairs (ex. pk-RSAwithSPHINCS), or both, are desired for standardization.

3. Whose responsibility is it to produce lists of acceptable pairs / tuples of algorithms? With generic composite, this could be left to protocols (TLS, CMS, CMP, IPSEC, etc), or even to runtime. With explicit composite, I think such a document is necessary to specify the OIDs at the PKIX level.

   This kinda boils down to whether LAMPS should be in the business of specifying OIDs for algorithm pairs for hybrid / dual, or whether this should be left to the runtime agents.

4. If explicit pairs is desired, should we continue developing the ASN.1 Information Object approach that we currently have in draft-ounsworth-pq-explicit-composite-keys, or would it be better to go with a syntactically simpler but less formal way of specifying the structures of pairs? Is there a syntax that would allow the ASN.1 Object Information Classes to accept an arbitrary number (2+) of input keys?