PQC AT THE IETF

Mike Ounsworth

PKI Consortium -- PQC Conference

3 March, 2023

TALK OUTLINE

- $ whois mike_ounsworth
- Problem Statement: PQC work at the IETF
- Problem Statement: Why this work is so 🤬'ing hard
  - Timely and graceful migration is hard
  - Unclear and fractured regulatory requirements
  - Challenges: HBS "at scale"
  - KEMs "don't fit"
- Case studies of PQC integration into IETF protocols
  - OpenPGP, X.509, TLS, CMS, CMP
- Hybrid and Composite PKI migration strategies

ENTRUST

SPEAKER BIO: MIKE OUNSWORTH

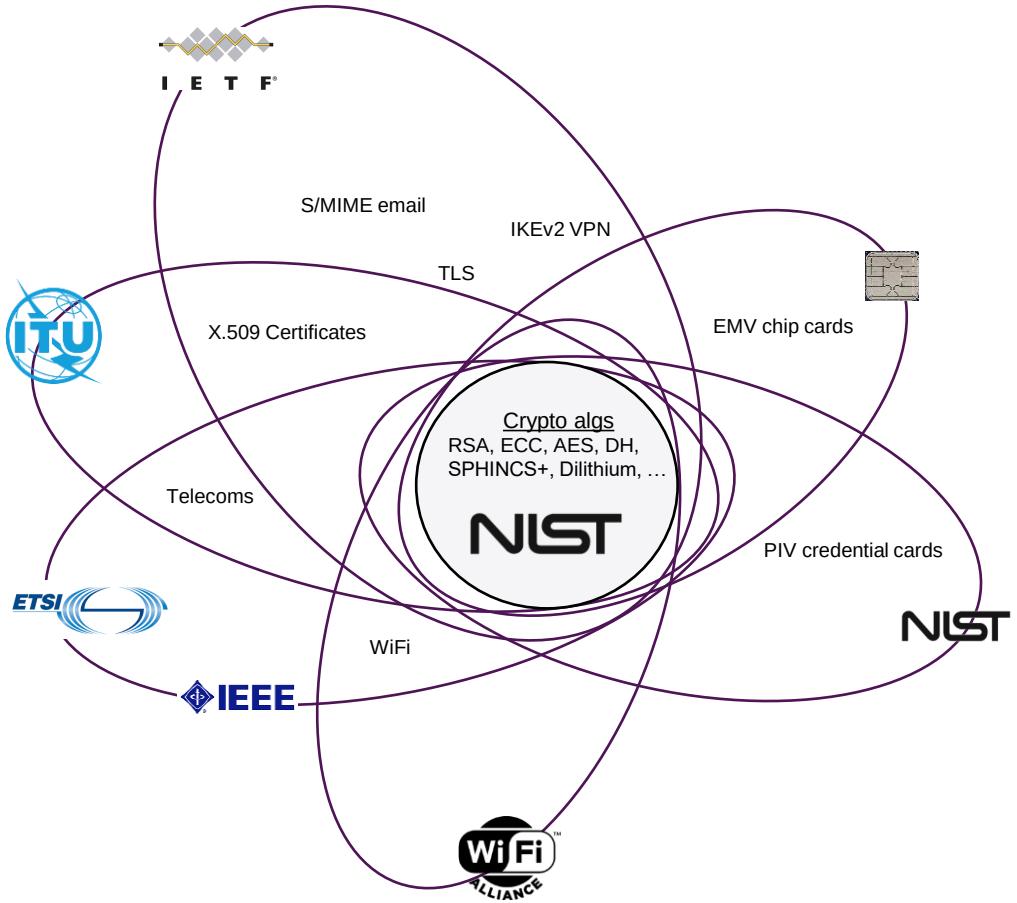
```
mike@mike-VirtualBox:~
→ ~ cat whois_mike_ounsworth.txt

Software Security Architect, Entrust
M.Sc Robotocs, AI, McGill
B.Sc CompSci, Math, Phys, Queen's University


AppSec tester
PQC research and crpyto architecture
IETF protocol design enthusiast
  > Author of 8 active Internet Drafts



They invited me (a massive nerd) to give this talk, so apologies
in advance if I get too technical.
```

ENTRUST

PROBLEM STATEMENT: PQC WORK AT THE IETF

ENTRUST

FRAGMENTED COMMUNITY



- **NIST is standardizing PQ crypto primitives.**
- Updating protocols that rely on crypto falls to each respective standards body.
- As participating members of the IETF, we can speak to action there.

ENTRUST

PROBLEM STATEMENT:
SCOPE OF PQC WORK AT THE IETF

The IETF owns the specs for many of the Internet's cryptographic and security protocols.

QUIC

TCP/IP

ACME

DNS

TLS

DNSSEC

X.509

DNS-over-HTTPS (DoH)

CMC

OCSP

CMS

OpenPGP

IPSEC / IKE

CMP

JOSE (JWT)

S/MIME

SCEP

HTTPS

COSE (CWT)

EST

Code-signing

Kerberos

SSH

ENTRUST

IETF CRYPTOGRAPHIC DEPENDENCIES
(NOT EXHAUSTIVE)

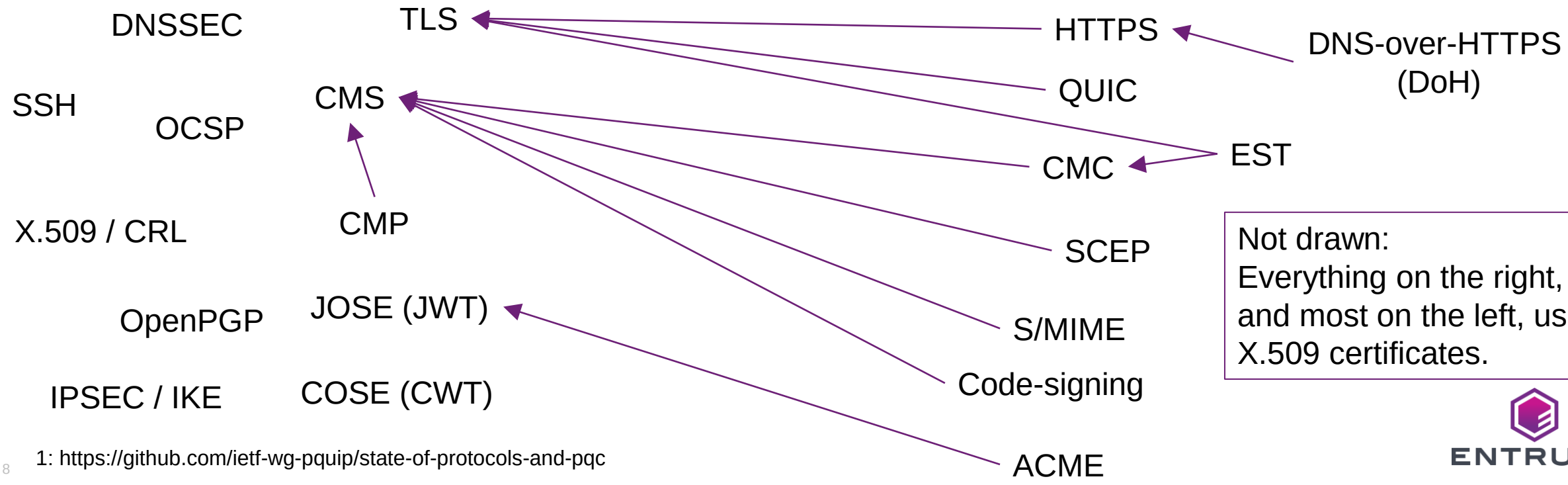
Good news: not everything needs to be touched.

Defines its own crypto

(ie needs updating)

Gets its crypto by embedding another protocol

(ie does not need updating)



Not drawn:
Everything on the right, and most on the left, use X.509 certificates.

1: https://github.com/ietf-wg-pquip/state-of-protocols-and-pqc

ENTRUST

ACTIVE PQC WORK AT IETF

- General approach: "Hurry up and wait": get drafts started, then pause them until final FIPS specs for Dilithium / Falcon / SPHINCS+, Kyber.

- LAMPS WG
  - X.509, CMP, CMS
- IPSECME WG
  - IKEv2
- TLS WG:
  - PQ KEMs, PQ certificates
- JOSE (JWT) and COSE (CWT)
- OpenPGP

- PQUIP
  - A new WG specifically to coordinate PQ work across the IETF.
- Protocols that need updating, but have no active WG:
  - SSH, Kerberos

- (other SEC area WGs excluded for brevity[2])

1: https://trac.ietf.org/trac/sec/wiki/PQCAgility
2: https://datatracker.ietf.org/wg/#SEC

ENTRUST

IETF PQ X.509 HACKATHON

- Started at IETF 115 (Nov 2022), monthly meetings since, continuing at IETF 116 (Mar 2023).
- Growing repo of test artifacts[1]
  - Samples of X.509 certs, CSRs, CRLs, and starting to do CMS objects.
  - Across all PQC signature algorithms, and hybrids.
  - Across 4 open source & 5 proprietary PKI implementations.
    - So far, no (major) interop problems 🤞

1: https://github.com/IETF-Hackathon/pqc-certificates

ENTRUST

IETF PQC WORK THAT ENTRUST IS CONTRIBUTING TO

- LAMPS WG
  - CMS: draft-ietf-lamps-cms-kemri – adding KEMs
  - CMP: draft-ietf-lamps-rfc4210bis – aka “CMPv3”
  - CMS: draft-ietf-lamps-cms-kyber
  - X.509 / CMS: draft-ounsworth-pq-composite-sigs
  - X.509 / CMS: draft-ounsworth-pq-composite-kem
- Cryptographic Research Forum (CFRG)
  - draft-fluhrer-cfrg-ntru-00
  - draft-ounsworth-cfrg-kem-combiners
- OpenPGP WG
  - draft-wussler-openpgp-pqc-00
- Active participants in the PQUIP WG

PROBLEM STATEMENT:
WHY THIS WORK IS SO 🤬'ING HARD

ENTRUST

CHALLENGES: TIMELY AND GRACEFUL MIGRATION IS HARD

“KNOBS AND DIALS”

- We had enough trouble migrating from RSA to ECDSA, or from RSA-SHA1 to RSA-SHA2.
- Way more “dials and knobs”:
  - Alg & param selection: Pub key size vs Priv key size vs keygen / sign time vs exhaustion limit.
  - PQ/T Hybrid, or pure PQ?
  - Mixed PKIs?
  - New algs to implement; “build, buy, or open source?”
- Navigating these tradeoffs will require expert knowledge of both the PQ primitives, and your PKI’s needs.

ENTRUST

CHALLENGES: HYBRIDS FOR SECURITY AND EASE OF MIGRATION?

- A “Post-Quantum / Traditional (PQ/T) Hybrid” is one of several techniques that use both cryptosystems together.
- Reasons you may want to explore hybrid solutions:
  - Security: protection against new attacks; hybrid buys you time to mitigate.
  - Migration and Backwards Compatibility: hybrid solutions allow complex environments to migrate more gracefully and avoid a hard “flag day”.
- Regulatory fracturing:
  - Hybrids required: BSI (Germany), ANSSI (France)
  - Hybrids allowed: ENISA (EU), ETSI
  - Hybrids discouraged: NSA (US), NCSC (UK), CSE (Canada)

CHALLENGES: UNCLEAR AND FRACTURED REGULATORY REQUIREMENTS

- Does “Software/firmware signing” specifically apply to chipset ROM and secure boot?
- Does that include the publicly-trusted Windows code-signing PKI?
  - Does an LMS PKI imply LMS OCSP responders and LMS TSAs? Those are much tricker than LMS CAs due to scale *(more on this later).*
- 2025 is basically tomorrow, are vendors ready? This implicates HSM vendor up to web server and browser vendors.
  - Chicken-and-egg: browser and webservers are waiting for IETF protocol specs, which are waiting for NIST Dilithium / Falcon / Kyber specs.

1: ”Announcing the Commercial National Security Algorithm Suite 2.0”, National Security Agency

ENTRUST

CHALLENGES: HBS “AT SCALE”
ISSUE: KEY EXHAUSTION

Hash-based signature (HBS) schemes, including LMS, HSS, XMSS, and SPHINCS+ all have limited-use private keys.

ParmSet	KeyGenTime	SigSize	KeyLifetime
15	6 sec	1616 bytes	30,000 sigs
…	…	…	…
15/15	6 sec	3332 bytes	1.0 billion sigs
…	…	…	…
25/15	1.5 hour	3652 bytes	1.0 trillion sigs

This seems like a lot, but have you ever thought about how many signatures your CA, OCSP Responder, or Timestamping Authority use per year? It could be billions / year for an active CA.

1: https://datatracker.ietf.org/doc/html/rfc8554 -- Table 3

ENTRUST

CHALLENGES: HBS “AT SCALE”
ISSUE: KEY EXHAUSTION

- We now need to think about keys as *% of expiry* and *% of exhaustion*.
  - This is a new paradigm (except for FIPS 140 PIV cards)
- PKCS#11 v3.1 has added[1]:
  - An application can ask an HSM: how many signatures are left on this key?

    `CKA_HSS_KEYS_REMAINING`

  - An HSM can refuse to produce any more signatures with a given key.

    `CKR_KEY_EXHAUSTED`

- What about Denial-of-Service (DoS)?

1: “PKCS #11 Specification Version 3.1 – OASIS”

ENTRUST

CHALLENGES: HBS “AT SCALE”
ISSUE: OPERATIONAL CONCERNS

- Choosing lifetime number of signatures an keygen – and balancing that against bandwidth.
- Private keys require very large storage: 100’s of gbs.
  - Is this even feasible on, for example, a smartcard? Those typically have 80 – 140 kb of storage. Need guidance from smartcard manufacturers.
- SP 800-208:

  “due to the risks associated with copying OTS keys [and state re-use], this recommendation prohibits exporting private keying material”
  “create a single stateful HBS key in which the OTS private keys are distributed across multiple cryptographic modules.”
  - This is a fundamental shift in HSM management, and needs firmware support for tree-splitting.

ENTRUST

(PAUSE FOR BREATH)

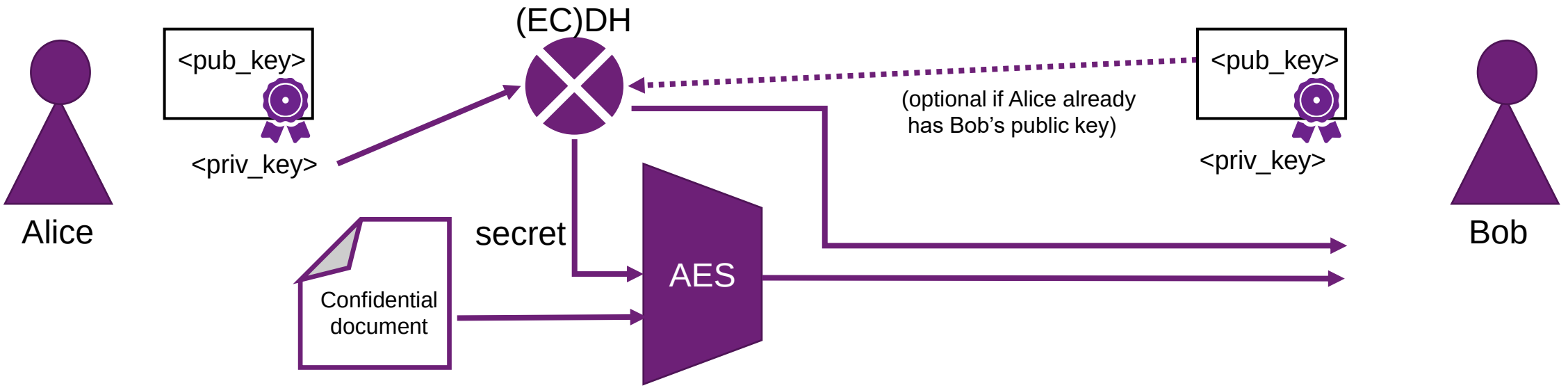
(CRYPTO NERD TIME)

ENTRUST

CHALLENGES: KEMS "DON'T FIT"
(EC)DH TODAY – AUTHENTICATED KEY EXCHANGE

Today, you can fire off (EC)DH ciphertext *and* your encrypted content all in the opening message. It's both *encrypted* and *authenticated*.

- "0.5 RTT AKE"

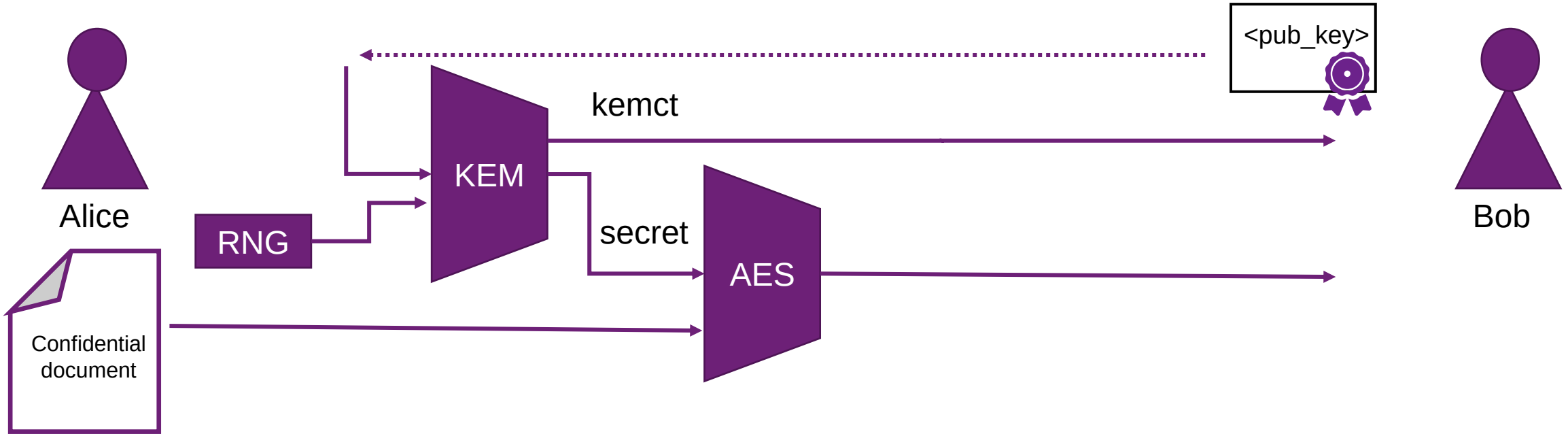


- The document is encrypted for Bob, *and* Bob knows that it was encrypted by Alice.

ENTRUST

CHALLENGES: KEMS “DON’T FIT”:
KEY ENCAPSULATION MECHANISM (KEM)

- Unfortunately, the NIST PQC encryption primitives are all in the shape of a “KEM”.



- This is different from RSA KeyTransport where Alice gets to choose the AES key.
- And different from (EC)DH KeyAgreement where both parties contribute a public key.
  - IMPORTANTLY: to get an AKE, you need to do 2 KEM exchanges: one in each direction.

ENTRUST

CHALLENGES: KEMS “DON’T FIT”:
KEM-BASED AUTHENTICATED KEY EXCHANGE (AKE)

- Getting an *authenticated* key exchange with KEMs requires 1 full round-trip (1 RTT) *before* you can encrypt anything – so 1.5 RTT for the first encrypted message.

  … and 3 calls of the KEM.Encaps() primitive.

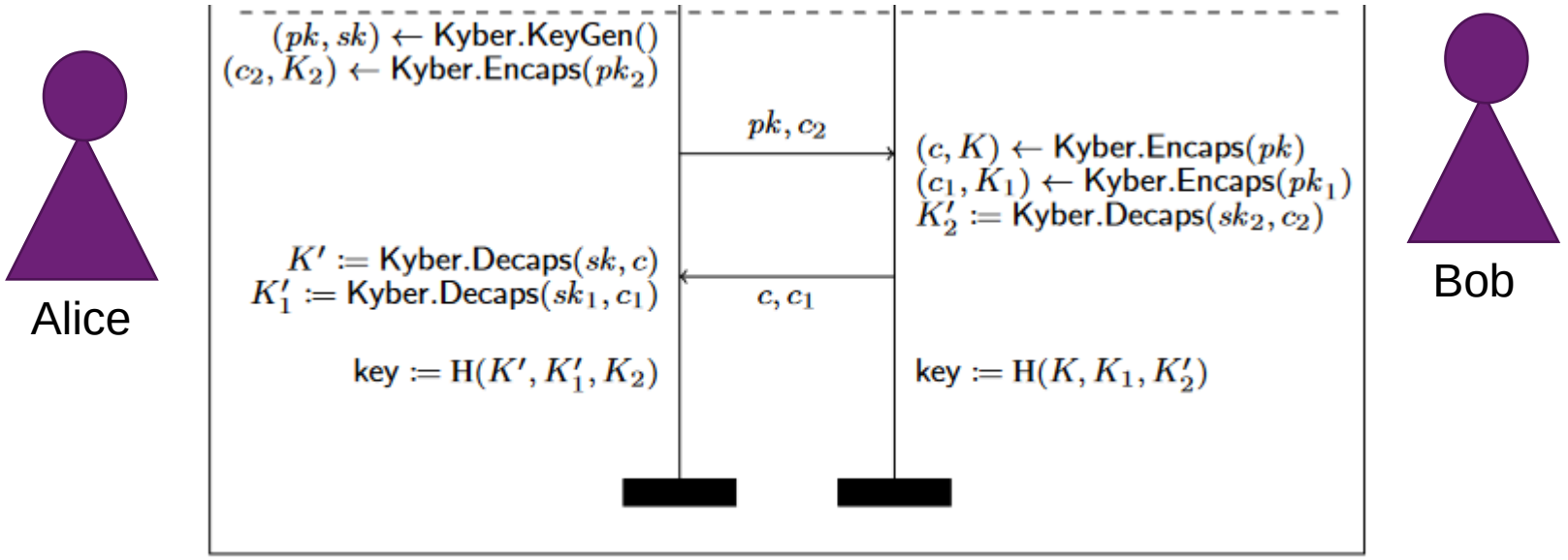


Figure 3. Kyber.AKE – Authenticated key exchange protocol using Kyber,

1: “CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM”, Bos et al. 2021

ENTRUST

CASE STUDIES OF PQC INTEGRATION INTO IETF PROTOCOLS:

X.509, OPENPGP, TLS, CMS, CMP

ENTRUST

CASE STUDY:
PQC IN OPENPGP AND X.509

- `draft-wussler-openpgp-pqc-00` (co-authored by BSI) *only* supports lattice schemes in PQ/T Hybrids:

Signatures (8)	KEMs (6)
Dilithium3 + Ed25519	Kyber768 + X25519
Dilithium5 + Ed448	Kyber1024 + X448
Dilithium3 + ECDSA-NIST-P-256	Kyber768 + ECDH-NIST-P-256
Dilithium5 + ECDSA-NIST-P-384	Kyber1024 + ECDH-NIST-P-384
Dilithium3 + ECDSA-brainpoolP256r1	Kyber768 + ECDH-brainpoolP256r1
Dilithium5 + ECDSA-brainpoolP384r1	Kyber1024 + ECDH-brainpoolP384r1
SPHINCS+-simple-SHA2	
SPHINCS+-simple-SHAKE	

- `draft-ounsworth-pq-composite-keys-03`
  Similar list for X.509, but X.509 has a more diverse set of usecases than PGP, … so lots more debate in the working group, which will result in a longer list.
  Currently: 14 signatures + 12 KEMs hybrids + all the pure PQC algs.

1: https://datatracker.ietf.org/doc/draft-ounsworth-pq-composite-keys

ENTRUST

CASE STUDY:
PQC IN TLS -- ENCRYPTION

- `draft-ietf-tls-hybrid-design-06` is fairly mature, and taking a PQ/T Hybrid approach:

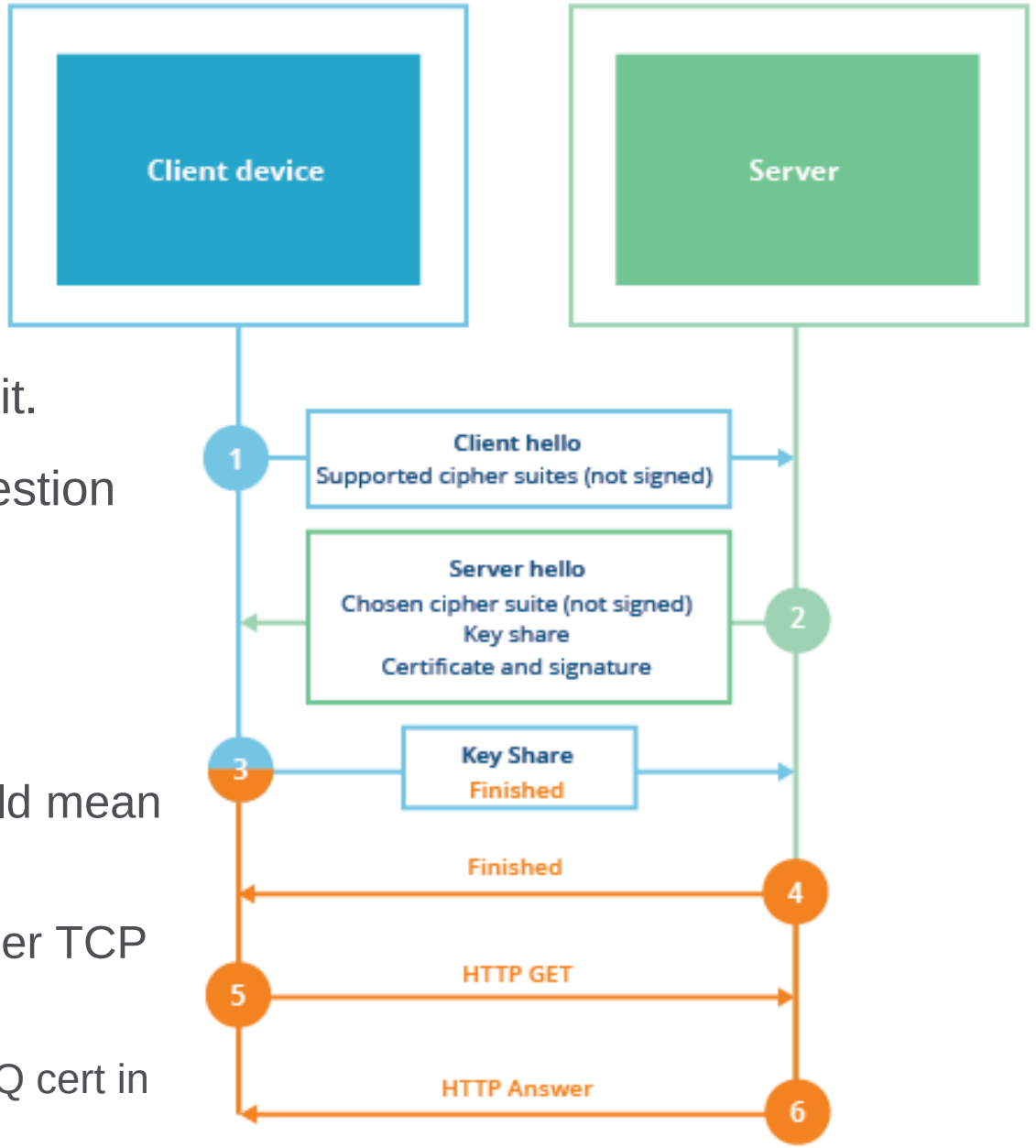
```
/* Hybrid Key Exchange Methods */
x25519_kyber768(TBD), secp384r1_kyber768(TBD),
x25519_kyber512(TBD), secp256r1_kyber512(TBD), ...,
```

- Do a traditional ECDH and a PQ and combine them together to form the session master secret.
- Easy.

ENTRUST

CASE STUDY:
PQC IN TLS – SIGNATURES / CERTS

- No drafts yet, or even rough consensus on how to do it.
- Problem: Step 2 is limited to ~ 5 kb by the TCP Congestion Window. So PQ certs won't fit.
- Possible solutions:
  1. Be ok with packet fragmentation in step 2.
  2. Only allow *extremely* small lattice schemes – would mean very short server cert lifetimes (like < 1 month).
  3. Move the certificate to step 5 where you have larger TCP packets.
     - Possibly as a PQ/T Hybrid: trad. cert in step 2, and PQ cert in step 4.



1: https://blog.cloudflare.com/rfc-8446-aka-tls-1-3/

CLOUDFLARE

CASE STUDY:
PQC IN CMS

- Cryptographic Message Syntax is the PKI-based encryption and signature layer used by S/MIME, PDF signing, Windows Code-signing, and more.
- New PQC signature algs will drop in (almost) for free.
- `draft-ietf-lamps-cms-kemri-00` defines a new message type `KEMRecipientInfo`.
- KEM-protected messages will use this with their `EnvelopedData`, otherwise nothing changes.
- Easy.

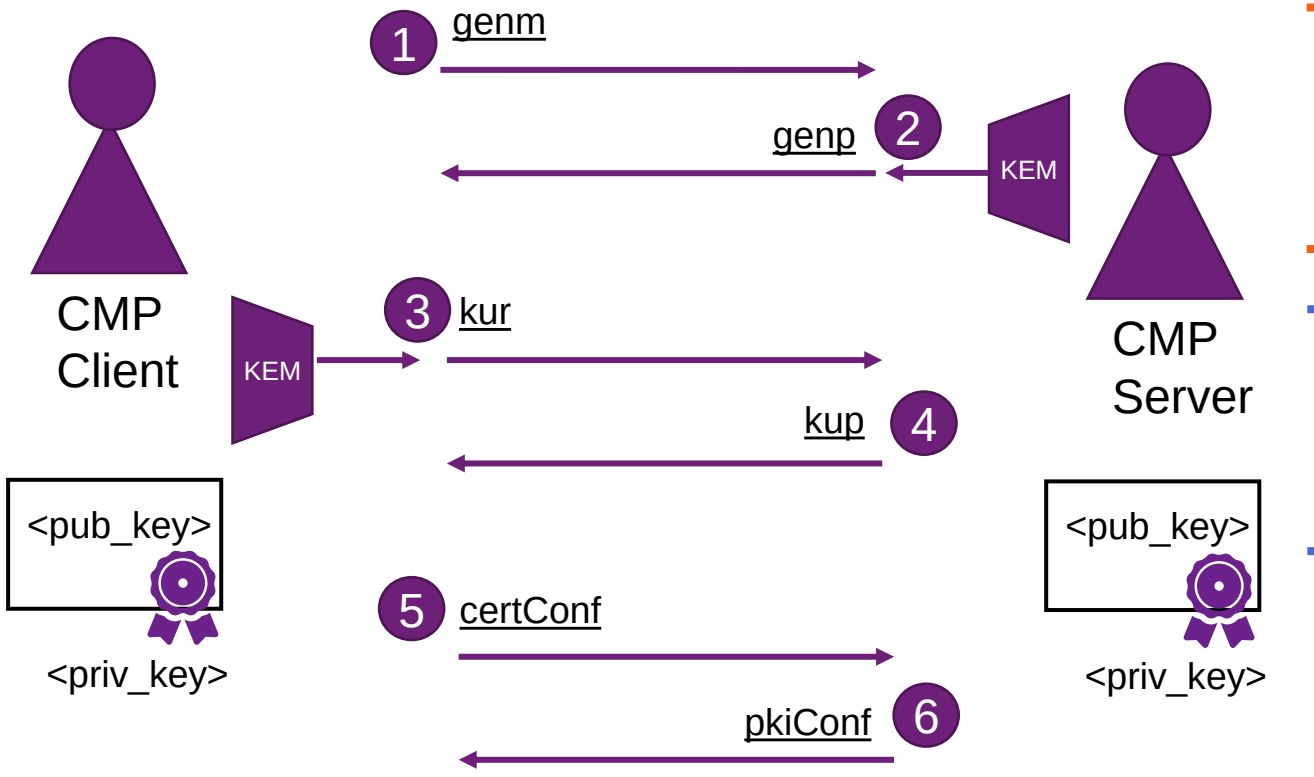
```
KEMRecipientInfo ::= SEQUENCE {
  version CMSVersion,  -- always set to 0
  rid RecipientIdentifier,
  kem KEMAlgorithmIdentifier,
  kemct OCTET STRING,
  kdf KeyDerivationAlgorithmIdentifier,
  kekLength INTEGER (1..MAX),
  ukm [0] EXPLICIT UserKeyingMaterial OPTIONAL,
  wrap KeyEncryptionAlgorithmIdentifier,
  encryptedKey EncryptedKey }
```

1: https://datatracker.ietf.org/doc/draft-ietf-lamps-cms-kemri

ENTRUST

CASE STUDY:
PQC IN CMP – EXAMPLE: KEY UPDATE (KUR)

- Certificate Management Protocol (CMP) is one of the original automated certificate enrollment protocols.



Compared with (EC)DH flow:

NEW

KEM AKE requires an extra round-trip.

MODIFIED

kur / kup message processing modified to accommodate KEM primitives, and to add HPKE (RFC 9180) hardening.

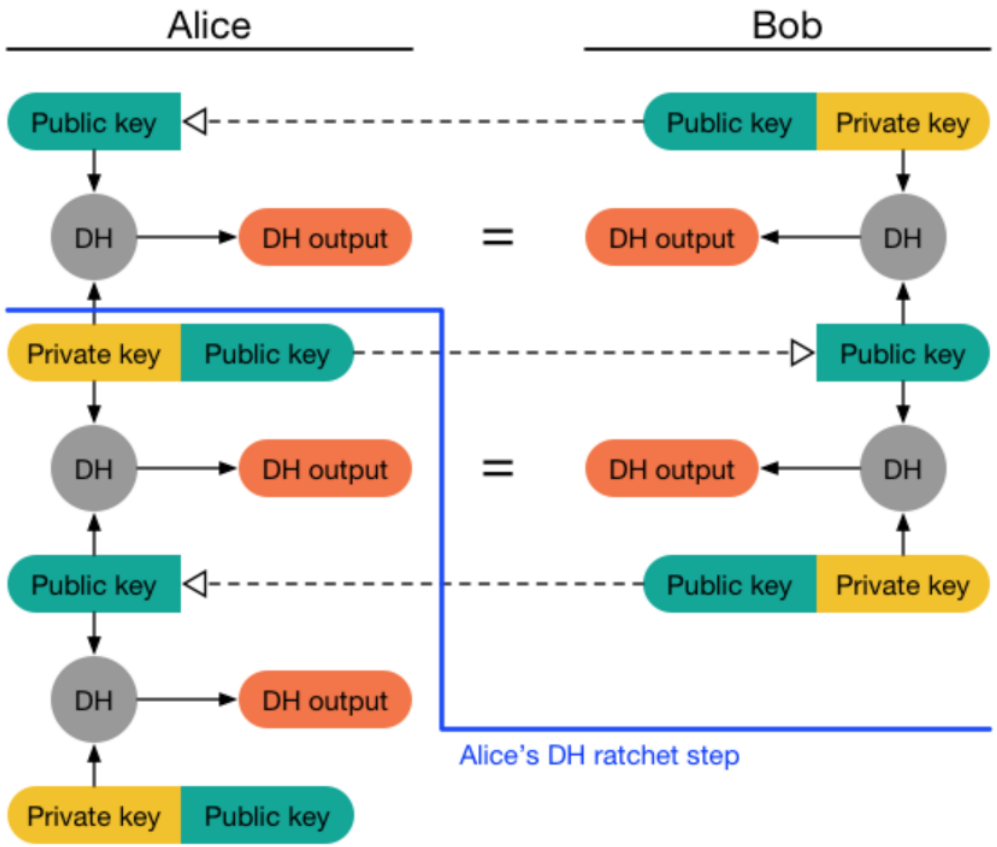
Basically, supporting management of KEM end entity certificates over CMP turns **2 RTT** protocols into **3 RTT**.

1: https://datatracker.ietf.org/doc/draft-ietf-lamps-rfc4210bis/ -- aka “CMPv3”

ENTRUST

BONUS: FITTING KEMS INTO SIGNAL PROTOCOL

- Signal Protocol's Double Ratchet is a beautiful application of the property that (EC)DH is a 0.5 RTT AKE.
- Many smart people are working on making KEM versions of the Double Ratchet.
- While it is of course possible, it won't be nearly as elegant and beautiful :'(



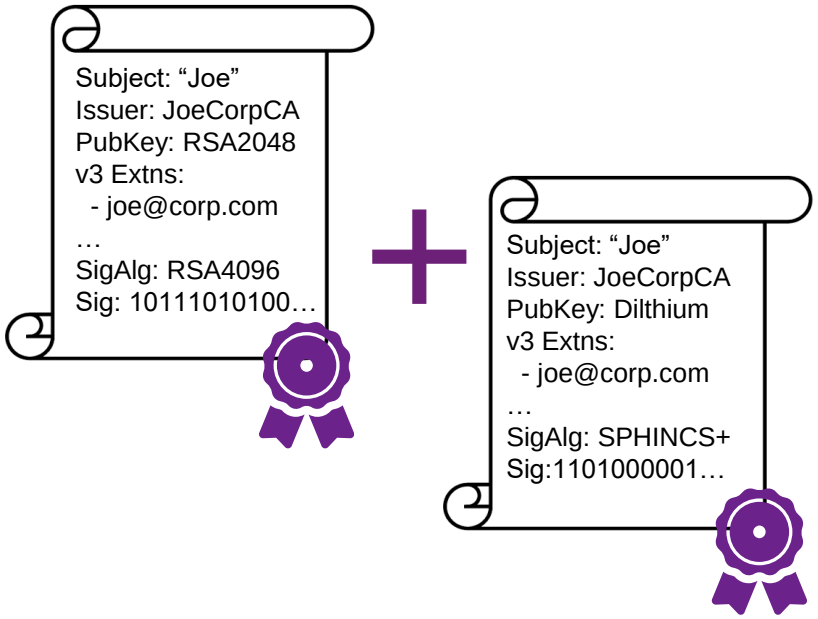
1: "The Double Ratchet Algorithm", https://signal.org/docs/specifications/doubleratchet/

ENTRUST

HYBRID APPROACHES FOR MIGRATING PKI

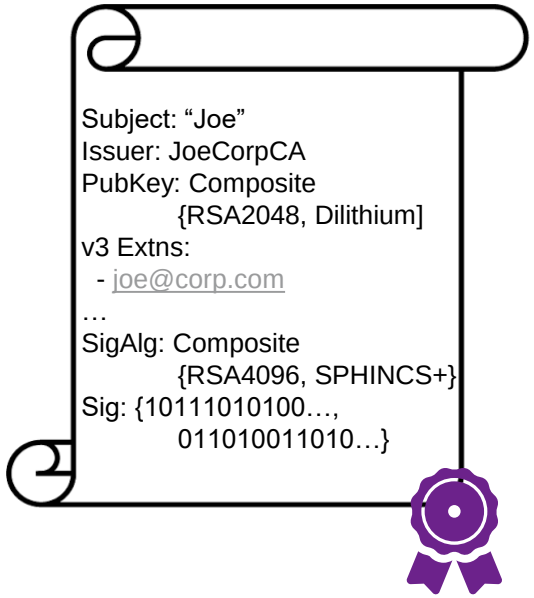
MULTI-CERT

"Parallel PKIs"



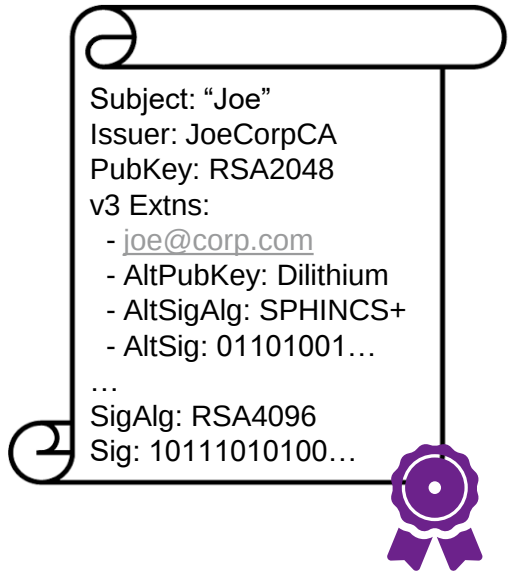
COMPOSITE [1]

IETF: draft-ounsworth-pq-composite-keys
draft-ounsworth-pq-composite-sigs



"HYBRID" CATALYST™ [2]

ITU-T: X.509 (10/2019)

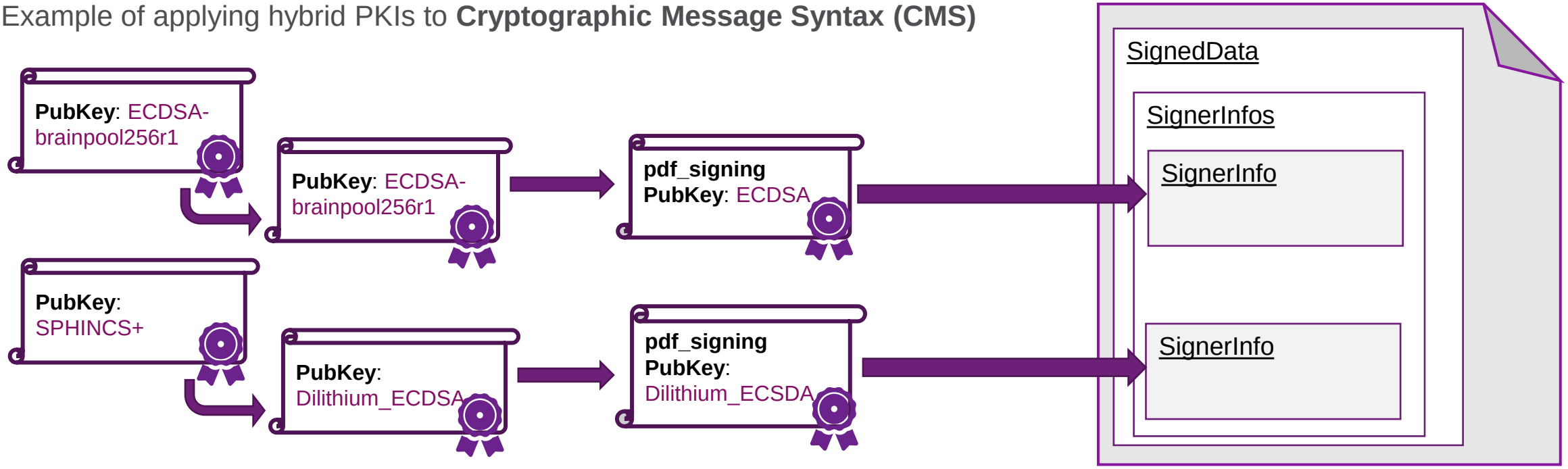


[1] Entrust – CableLabs -- D-Trust – Cisco collaboration; IETF draft
[2] ISARA - Entrust - Cisco collaboration; IETF and ISO drafts
> https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509

ENTRUST

CASE STUDY:
HYBRID PKIS

Example of applying hybrid PKIs to **Cryptographic Message Syntax (CMS)**



- Backwards compatibility: CMS clients (code-signing, PDF, S/MIME) already handle multiple *SignerInfos* today.
  - So legacy clients ***should*** gracefully skip the PQ signature.
- Redundancy gives migration flexibility. PQ-aware clients can validate either:
  - PQ signature only, or
  - Both parallel signatures independently.

RFC5652 - SignerInfos:
"When the collection represents more than one signature, **the successful validation of one of the signatures** from a given signer ought to be treated as a successful signature by that signer..."

ENTRUST

SUMMARY

- This crypto migration will be the hardest we've ever done, full of *"square peg, round hole"* problems in all areas:
  - Protocol and application design.
  - Regulatory requirements and timelines.
  - Operational procedures.

- … to be continued. Keep watching:
  - The NIST PQC "competition", and NCCoE PQC Migration Project.
  - Updates from regulatory bodies – NSA, ENISA BSI, ANSSI, ETSI
  - IETF Working Group discussions

ENTRUST

END