

Post-Quantum

Cryptography Conference

PQC Standardization at the Internet Engineering Task Force (IETF)

Just as post-quantum cryptography (PQC) has presented significant challenges for academic cryptographers, so too has it posed unique challenges for cryptographic engineers. The new PQC primitives, with their distinct characteristics compared to traditional RSA and ECC algorithms, often require substantial protocol and application redesign to accommodate them effectively. Moreover, the need for a relatively abrupt transition to PQC across the Internet's vast infrastructure has introduced additional complexities. This presentation will provide a comprehensive overview of the latest developments in PQC standardization within the IETF. We will delve into the challenges and progress made in integrating PQC into common Internet protocols, highlighting key areas where work is still underway. Additionally, we will explore the implications of the newly standardized algorithms (ML-DSA, SLH-DSA, ML-KEM, LMS, XMSS) and discuss the strategies for their successful deployment. Finally, we will share insights from our research on PKI PQ/traditional hybrid modes, which offer a promising approach for enhancing both security and migration flexibility during the transition to a post-quantum world.



Mike Ounsworth

Software Security Architect at Entrust



KEYFACTOR



January 15 and 16, 2025 - Austin, TX (US) | Online

PKI Consortium Inc. is registered as a 501(c)(6) non-profit entity ("business league") under Utah law (10462204-0140) | pkic.org



PKI
Consortium

PQCPQC Standardization at the Internet Engineering Task Force (IETF)

Mike Ounsworth
Software Security Architect

PKI Consortium
January 2025
Austin, Texas

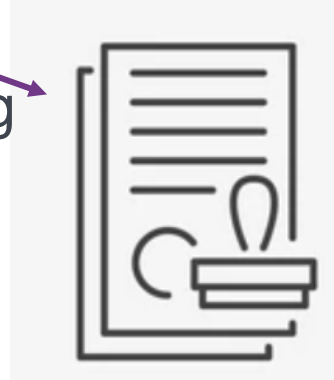




Agenda

- What needs updating?
- How's that going?
 - X.509 (LAMPS WG)
 - ACME
 - TLS & HPKE (TLS WG and CRFG)
 - IPsec (IPSECME WG)
 - OpenPGP
 - JWT / CWT (JOSE / COSE WG)

50%
Registering
new algs.



50%
Forcing KEMs
and Hybrids
into places
they don't
fit.



ENTRUST

What needs updating?

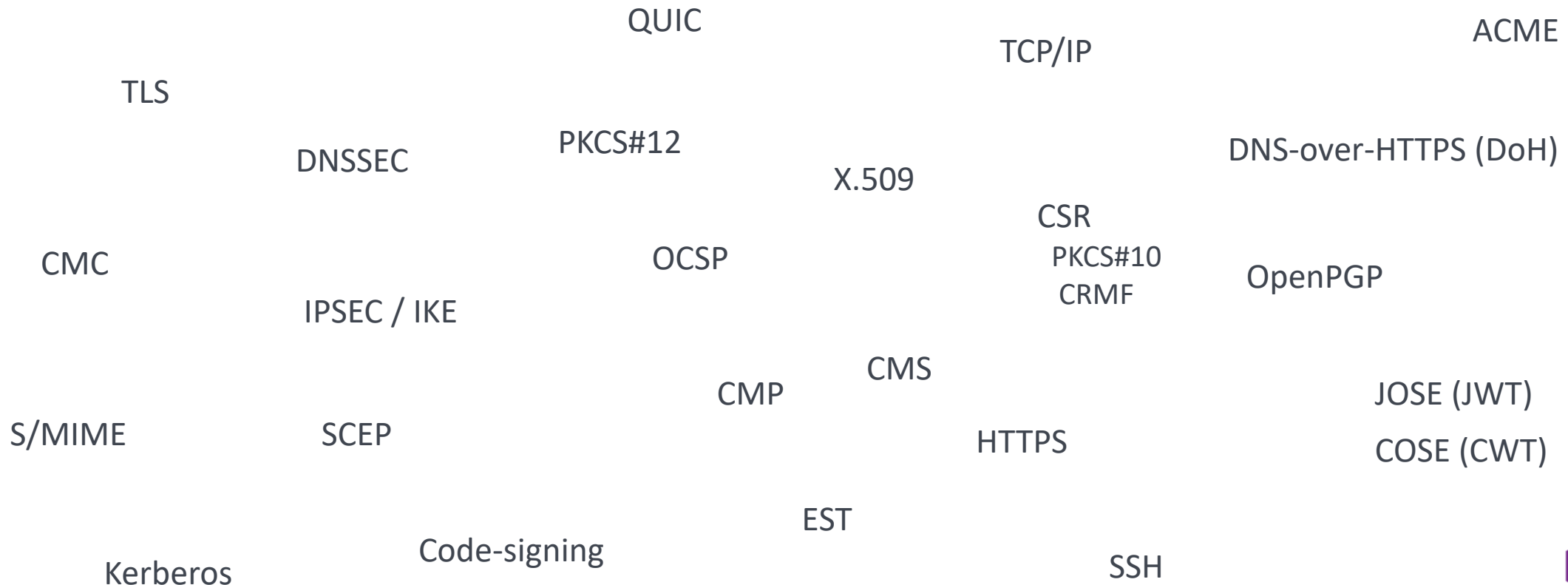


ENTRUST

PROBLEM STATEMENT: SCOPE OF PQC WORK AT THE IETF



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



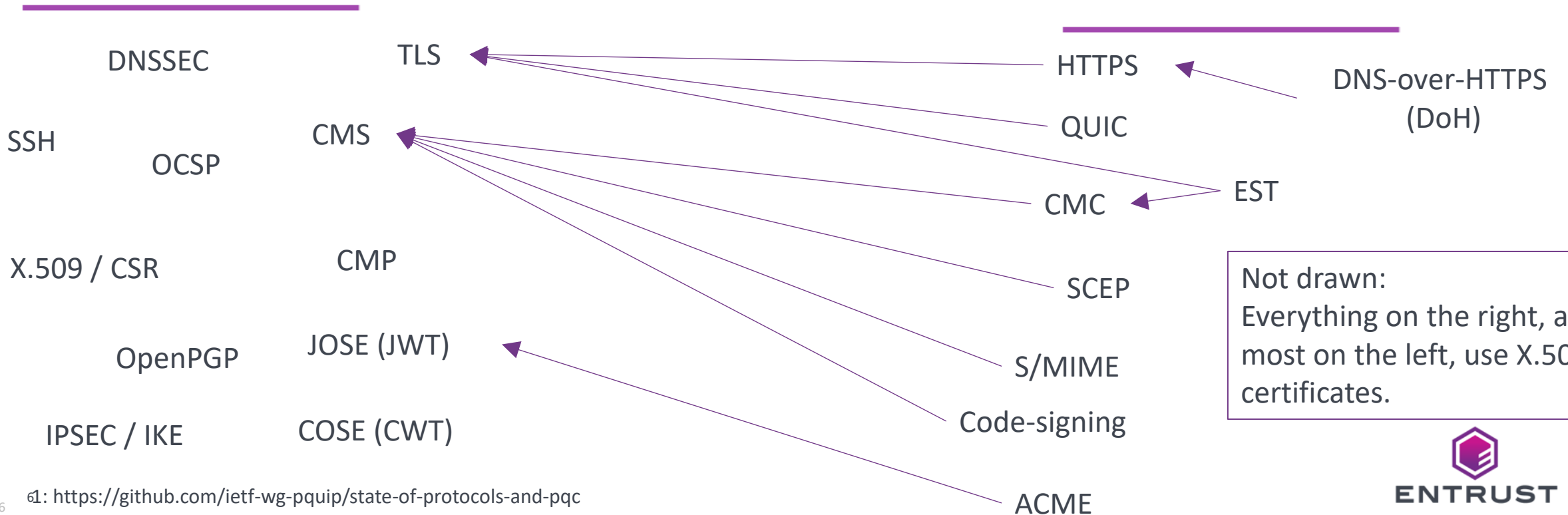
IETF CRYPTOGRAPHIC DEPENDENCIES

(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.



ENTRUST

How's that going?

Billions of bytes of emails have been spent on this effort,
... and doubtless billions still to go.



ENTRUST

IETF PQC Status

- This presentation pulls mainly from:
 - <https://github.com/ietf-wg-pquip/state-of-protocols-and-pqc>
 - (which I updated while writing these slides. Disclaimer: I may have missed something)
- Status legend:
(IETF process flow: reminder: “Internet Drafts” become “RFCs”)



LAMPS WG

**X.509, CRL, CSR, CMS, S/MIME, CMP,
EST, etc**



ENTRUST

LAMPS

PQC Drafts

Signatures

Algorithm	Status	Ref
LMS / XMSS	RFC	[RFC9708]
ML-DSA	WGLC	[2], [3]
Composite ML-DSA	<i>Adopted</i>	[4]
SLH-DSA	WGLC	[5], [6]

KEMs

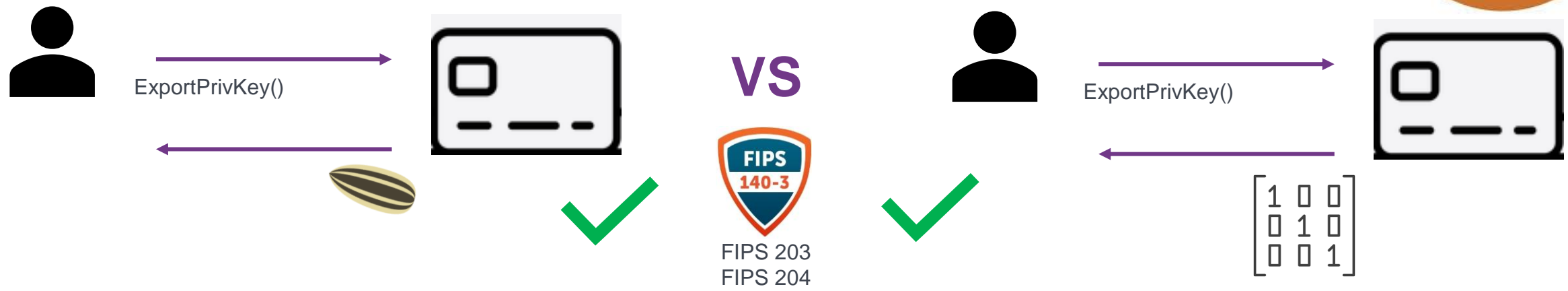
Algorithm	Status	Ref
ML-KEM	WGLC	[7], [8]
Composite ML-KEM	<i>Adopted</i>	[9]

Open Design Issues

ML-DSA and ML-KEM KeyGen() – seeds vs expanded



!!! Accurate as of Jan 10, 2025, may change quickly !!!



- IETF LAMPS wants to only use **seed** private keys in PKCS#12 files.
- PKCS#11 v3.2 (draft) says “Private value as defined in [FIPS 203]” which, I guess?, allows either?
- If your software only handles seeds, and your hardware only handles expanded, we’re gonna have problems!
- I have been screaming into the void as loudly as I can that we need to align LAMPS and PKCS#11, or we’ll have 10 years of compatibility nightmares.

Open Design Issues

ML-DSA pre-hash mode (“HashML-DSA” vs “ExternalMu-ML-DSA”)



!!! Accurate as of Jan 10, 2025, may change quickly !!!

- FIPS 204 defines ML-DSA and HashML-DSA.
- 🙌 😬 BUT WAIT ... there’s a 3rd (hidden) option!

Algorithm 7 ML-DSA.Sign_internal(sk, M', rnd)

6: $\mu \leftarrow H(\text{BytesToBits}(tr) || M', 64)$ \triangleright message representative that may optionally be computed in a different cryptographic module

- We are calling this “External Mu” mode, and we think it is a better pre-hash mode than the HashML-DSA that is defined in FIPS 204. The API for it looks like this:

```
 $\mu$  = ExternalMu-ML-DSA.Prehash( pk, M, ctx )  
sig = ExternalMu-ML-DSA.Sign( sk,  $\mu$  )
```

- LAMPS wants to forbid HashML-DSA in favour of ExternalMu-ML-DSA, *however*, PKCS#11 v3.2 draft doesn’t (currently) provide an API for External Mu mode.

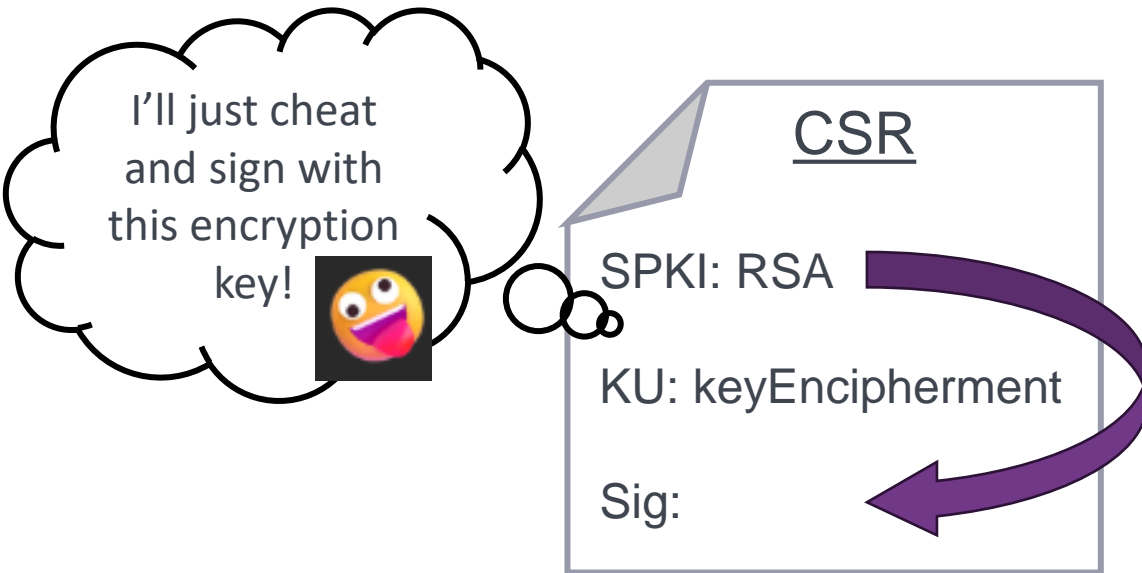


Open Design Issues

How do you issue KEM certificates from a PKI? 🙅 🤔



- Keys of type `KeyUsage: keyEncipherment` are not supposed to create digital signatures.



- But there is actually no way to create a signature with a KEM key.
- (also true for DH / ECDH keys, but we never really used those, so nobody really noticed)

Open Design Issues

How do you issue KEM certificates from a PKI? 🙅 🤔



Options:

One-shot:

(ie enrollment only requires one request-response exchange to the CA)

- Server-generated private key; ex.: cert issuance returns a p12 with a private key in it.
- Use an already-issued signing cert to sign the KEM CSR. [7]

Challenge-response Proof-of-Possession:

(ie requires at least two round-trips to the CA)

- CMPv3 [8]
- CRMF CSR [9]
- CMC-over-EST [9a]

Note: not ACME since that requires PKCS#10 CSRs.

Problem for S/MIME and IoT?



ACME

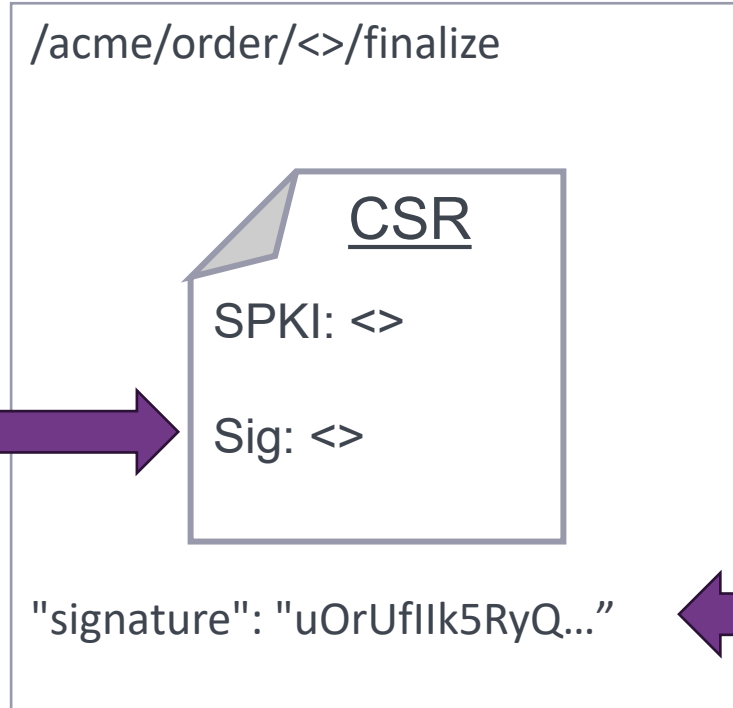


ENTRUST

ACME

PQC Drafts

This gets PQC for free once the PQC algs are registered in LAMPS / X.509.



This gets PQC for free once the PQC algs are registered in JOSE / JWT.



Awesome. Nothing to do here!

TLS



ENTRUST

TLS PQC Drafts

Signatures

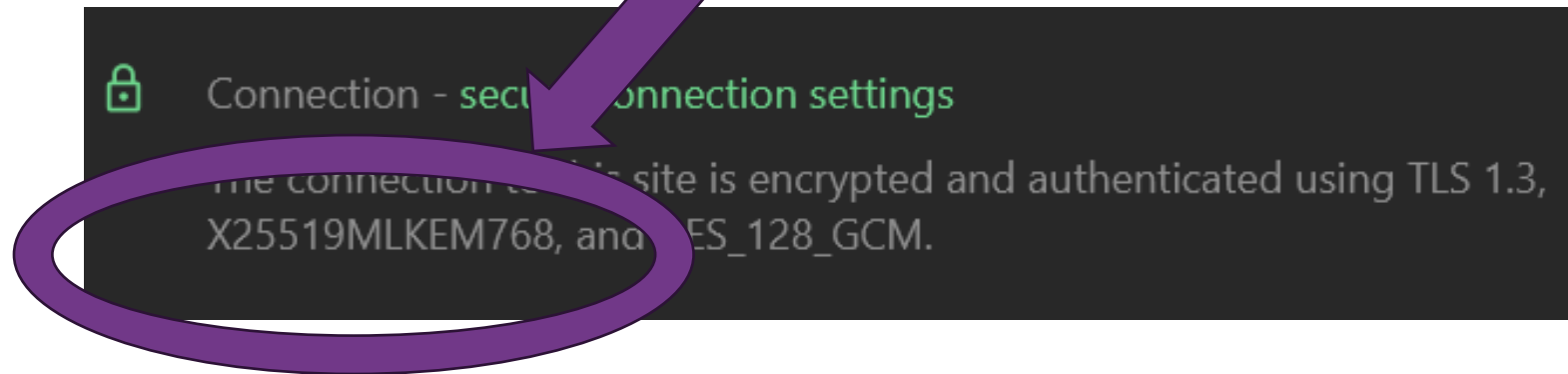
Algorithm	Status	Ref
ML-DSA	No Draft	--
SLH-DSA	Exists	[13]
Composite ML-DSA	Exists	[12]

ML-DSA is lagging, with some debate in the community about whether there is any urgency for this.

KEMs

Algorithm	Status	Ref
ML-KEM	Exists	[10]
X25519MLKEM768, SecP256r1MLKEM768, SecP384r1MLKEM1024	Exists	[11]

Despite not even being adopted by the TLS WG yet, Chrome, Google servers, and CloudFlare Servers already support this in mainline.



IPsec



ENTRUST

IPsec

PQC Drafts

Signatures

Algorithm	Status	Ref
ML-DSA / SLH-DSA	???	
Hybrid Auth <ul style="list-style-type: none">All ML-DSA composites from LAMPSAny combination of single-alg certs	Exists	[15]

KEMs

Algorithm	Status	Ref
ML-KEM	Exists	[14]
Hybrid ML-KEM	Exists	[14]
Hybrid FrodoKEM	Exists	[16]

Disclaimer: I'm not expert enough in IPsec to know if it will get ML-DSA for free once X.509 supports it, or if it needs a draft to the IPSECME WG.

OpenPGP



ENTRUST

OpenPGP PQC Drafts

Signatures

Algorithm	Status	Ref
ML-DSA	Adopted	[17]
ML-DSA-65+Ed25519 ML-DSA-87+Ed448	Adopted	[17]
SLH-DSA	Adopted	[17]

KEMs

Algorithm	Status	Ref
ML-KEM	Adopted	[17]
ML-KEM-768+X25519 ML-KEM-1024+X448	Adopted	[17]



No hybrids with P256 or Brainpool curves.
→The openpgp-pqc draft has been adopted, but not yet debated very heavily, so these choices may change.

JWT / CWT



ENTRUST

JWT / CWT

PQC Drafts

Signatures

Algorithm	Status	Ref
LMS	RFC	[RFC8778]
ML-DSA	Adopted	[20]
Composite ML-DSA	Exists	[23]
SLH-DSA	Adopted	[21]

KEMs

Algorithm	Status	Ref
ML-KEM	Adopted	[18]
MLKEM768 + X25519 (X-Wing)	Exists	[22]

Summary

- LAMPS (X.509) is leading with ML-DSA, ML-KEM, and SLH-DSA in WGLC, and Composite ML-DSA, Composite ML-KEM following shortly.
- TLS has mature implementations of X25519MLKEM768 (Google Chrome, CloudFlare), but draft is technically not even adopted yet.
 - ML-DSA is lagging, with some debate in the community about whether there is any urgency for this.
- Other WGs / protocols are mostly at the “Individual Submission” or early “Adopted” state.
 - LOTS of debate about the correct way to do hybrids within each protocol, so expect that hybrids may still need some design iterations (other than X.509 and TLS where this is fairly stable).

Bonus

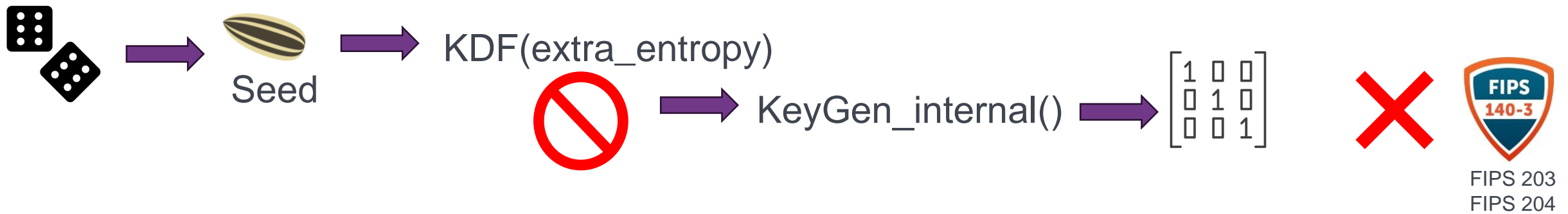
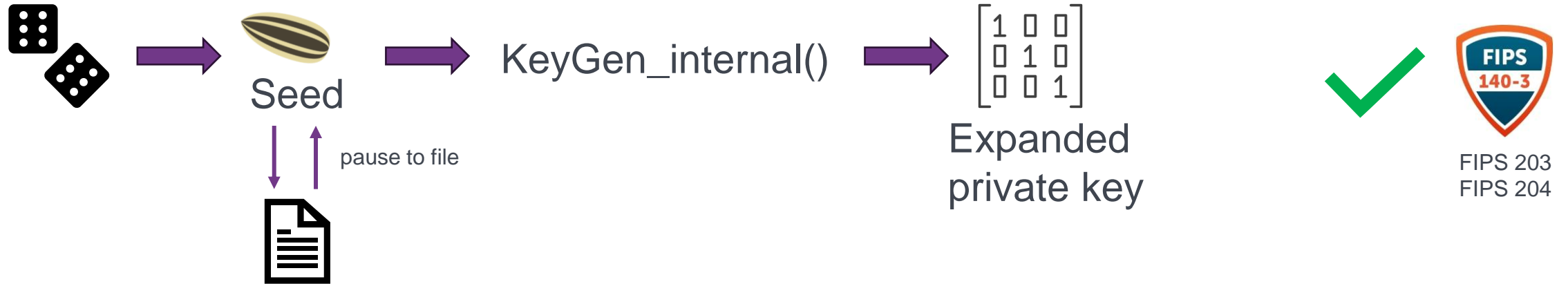
More landmines

(if time)



ENTRUST

Direct Seed vs Derived Seed



Direct Seed vs Derived Seed

X-Wing

- X-Wing is trying to be the one simple ML-KEM-768 + X25519 hybrid for use everywhere. It is *mostly* compatible with LAMPS Composite id-MLKEM768-X25519, **except:**

```
def XWing.expandDecapsulationKey(sk):
```

```
    expanded = SHAKE256(sk, 96)
```

```
    (pk_M, sk_M) = ML-KEM-768.KeyGen_internal(expanded[0:32], expanded[32:64])
```

```
    sk_X = expanded[64:96]
```

```
    pk_X = X25519(sk_X, X25519_BASE)
```

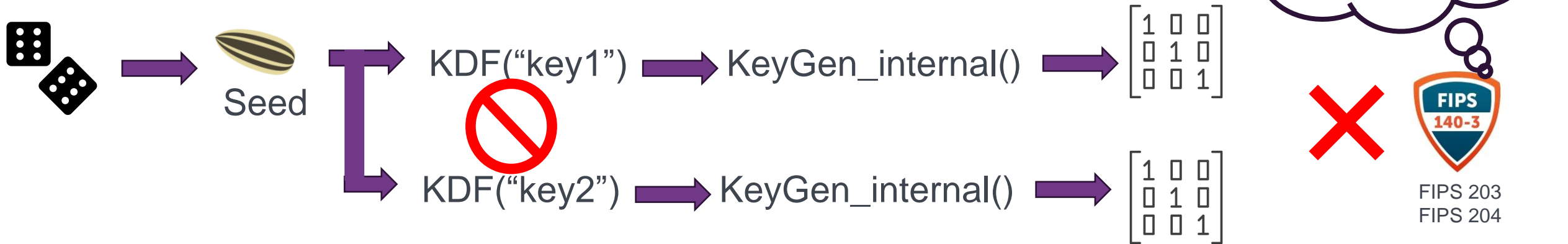
```
    return (sk_M, sk_X, pk_M, pk_X)
```




Not allowed by FIPS 203.
So an implantation of X-Wing
(at least its KeyGen) cannot be
FIPS validated.



Direct Seed vs Derived Seed



- But some devices really need to be able to do this.
- Consider, for example, a FIDO2 token  which is too small to have a good onboard RNG, but needs unique keys per website. Here,

$\text{KDF}(\text{high_entropy_seed} + \text{website_url}) \rightarrow \text{KeyGen_internal}()$
is a totally reasonable strategy.

- So, if you make a device like this, be aware that there is (currently) no way to do it and be compliant with FIPS 203 / 204.

References



ENTRUST

References

- [RFC9708]: LMS in X.509
- [2]: draft-ietf-lamps-dilithium-certificates
- [3]: draft-ietf-lamps-cms-ml-dsa
- [4]: draft-ietf-lamps-composite-sigs
- [9]: draft-ietf-lamps-composite-kems
- [5]: draft-ietf-lamps-x509-slhdsa
- [6]: draft-ietf-lamps-cms-sphincs-plus
- [7]: draft-ietf-lamps-kyber-certificates
- [8]: draft-ietf-lamps-cms-kyber
- [7]: draft-housley-lamps-private-key-attest-attr
- [8]: cmpv3 - draft-ietf-lamps-rfc4210bis
- [9] KEM PoP in CRMF – maybe does not exist yet?
- [9a]: draft-ietf-lamps-rfc5272bis
- [10]: draft-connolly-tls-mlkem-key-agreement/
- [11]: draft-kwiatkowski-tls-ecdhe-mlkem
- [12]: draft-reddy-tls-composite-mldsa
- [13]: draft-reddy-tls-slhdsa
- [14]: draft-kampanakis-ml-kem-ikev2
- [15]: draft-hu-ipsecme-pqt-hybrid-auth
- [16]: draft-wang-hybrid-kem-ikev2-Frodo
- [17]: draft-ietf-openpgp-pqc
- [18]: draft-ietf-jose-pqc-kem
- [RFC8778]: LMS in COSE
- [20]: draft-ietf-cose-Dilithium
- [21]: draft-ietf-cose-sphincs-plus
- [22]: draft-reddy-cose-jose-pqc-hybrid-hpke
- [23]: draft-prabel-jose-pq-composite-sigs