Post-Quantum

Cryptography Conference

HSM Advances Supporting quantum-safe PKI Automation

Olivier Couillard
Technical Product Manager at Crypto4A Technologies, Inc.

KEYFACTOR CRYPTO4A SSL.com ENTRUST HID

October 28 - 30, 2025 - Kuala Lumpur, Malaysia

PKI Consortium

HSM Advances Supporting Quantum-Resistant PKI Automation

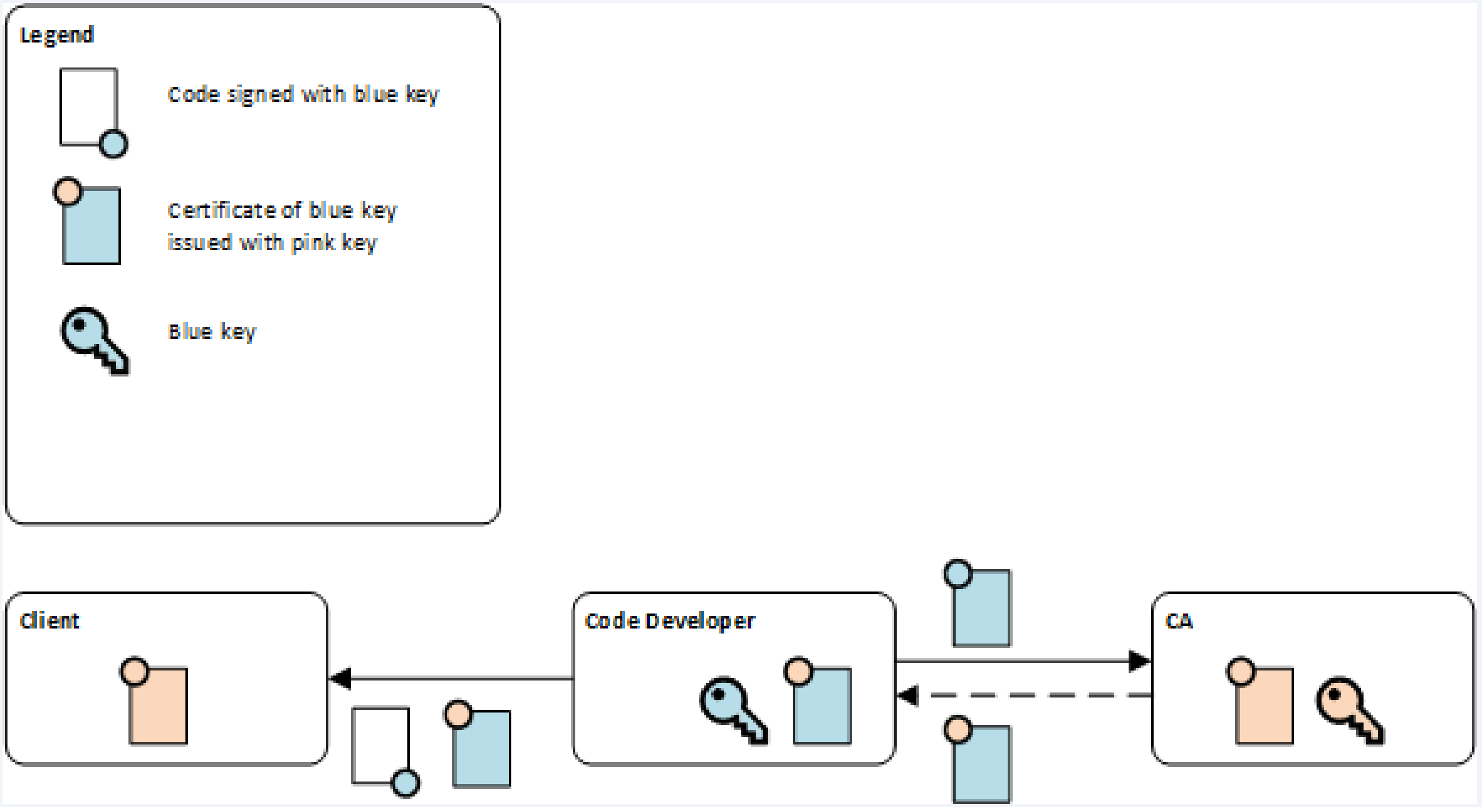
Agenda

- Why is attestation useful?
  - Overview of a code signing example
- What is attestation?
  - RFC 9334
  - IETF’s RATS for HSMs
  - Attestation format
- Why should attestation be quantum-resistant?
  - Algorithm considerations
- How can attestation be made quantum-resistant?
  - Roots of Trust (ROTs) and Trust Anchors (TAs)
- What are other use cases for attestation?
  - C2PA, audit logs, automatic HSM clustering

CRYPTO4A

Code Signing Example

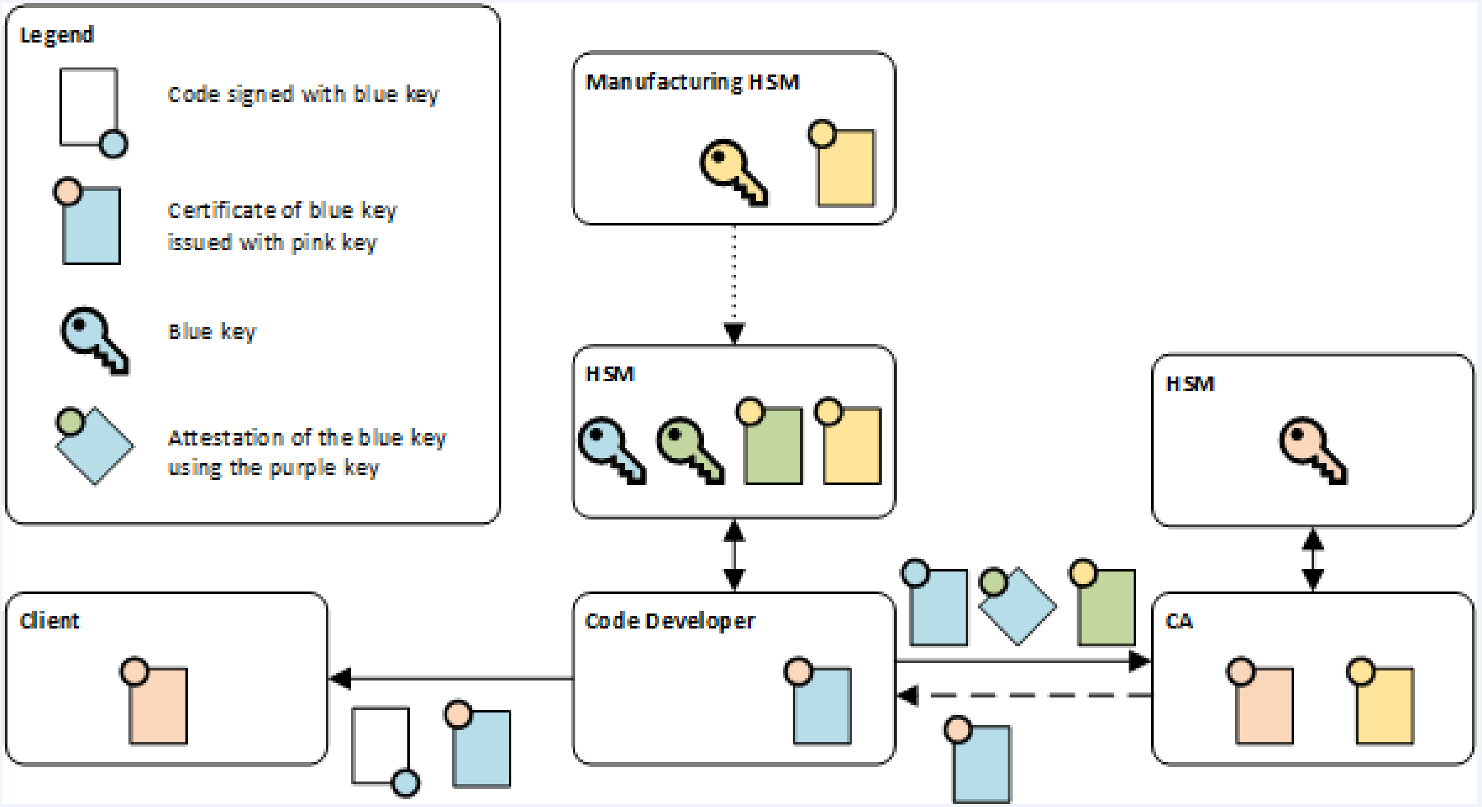
- Code developer has code signing key certified by a certificate authority (CA)
- Client inherently trusts CA
- Code is signed by code signing key and delivered to the client



CRYPTO4A

Code Signing Example

- Code developer has code signing key certified by a certificate authority (CA)
- Client inherently trusts CA
- Code is signed by code signing key and delivered to the client
- Code signing key is in an HSM
- An attestation targeting the code signing key can be generated on demand
- The CA inherently trusts the root certificate in the attestation's certificate chain
- A manufacturing HSM provisions the attesting HSM with appropriate attestation keys



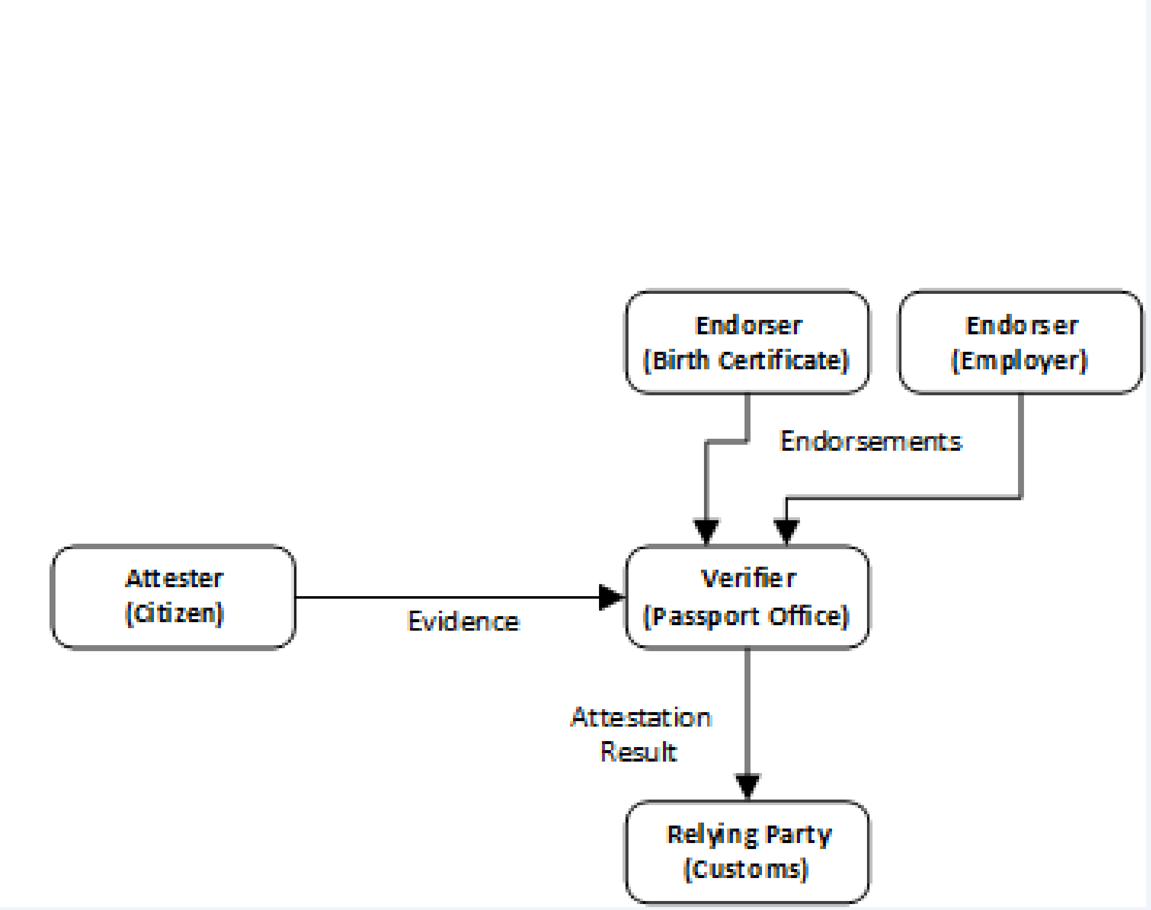
What is attestation?

- From RFC 9334:

In Remote ATtestation procedureS (RATS), one peer (the "Attester") produces believable information about itself ("Evidence") to enable a remote peer (the "Relying Party") to decide whether or not to consider that Attester a trustworthy peer. Remote attestation procedures are facilitated by an additional vital party (the "Verifier").

RFC 9334 – Passport Model

- **Endorser**: Various (contacts, birth certificate issuer, employer, etc.)
- **Attester**: Citizen
- **Verifier**: Passport-issuing agency
- **Relying Party**: Customs
- **Evidence:** Passport application
- **Attestation Result:** Passport



CRYPTO4A

Challenges with HSMs

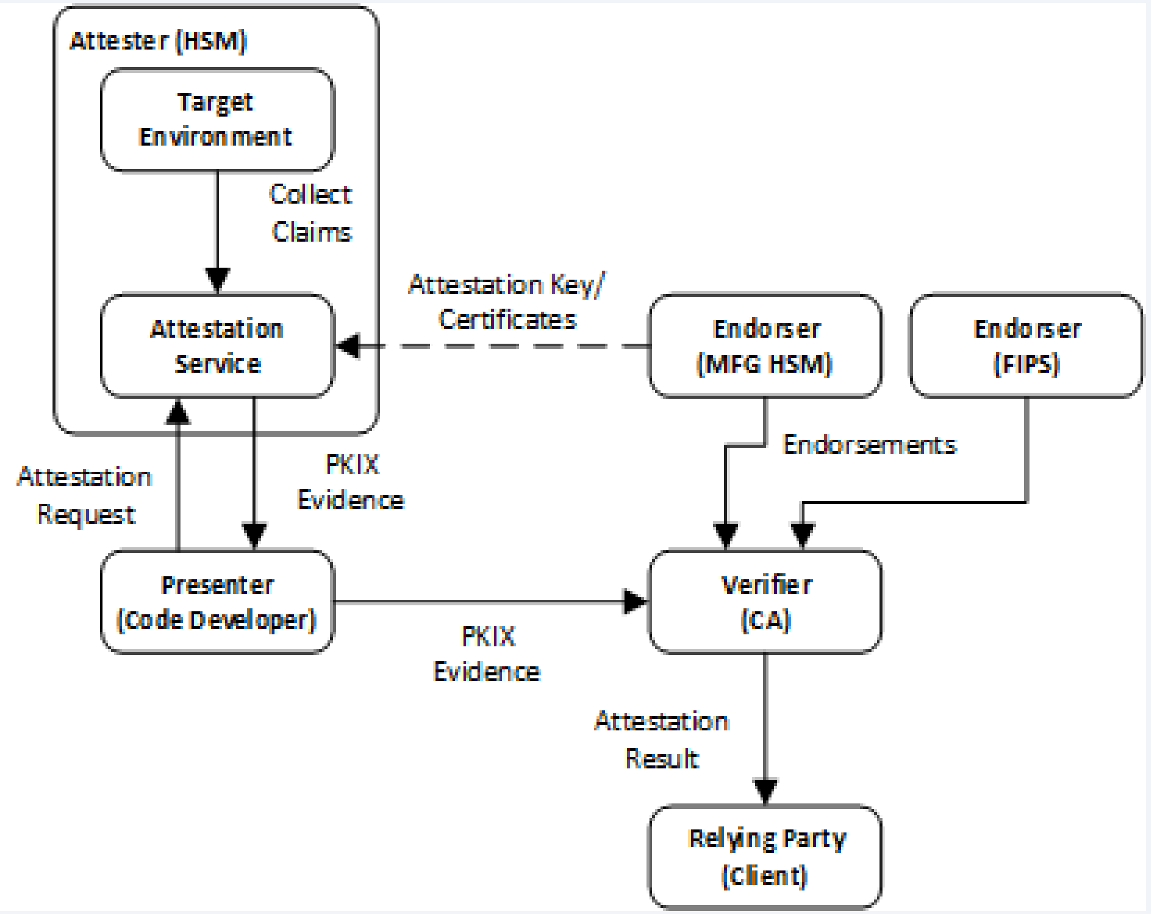
1. **Limited encoding capabilities:** Beyond ASN.1, there aren't many encoding capabilities typically found in HSMs.
2. **Freshness of an attestation:** The state of an HSM may change after an attestation of its state is made.
3. **Privacy concerns:** An HSM shouldn't divulge information that does not pertain to the process.

RATS – PKIX Key Attestation

1. **ASN.1 Format**: Defining an ASN.1 format for the attestation to facilitate interoperability.
2. **Attestation Request:** A "Presenter" role is introduced. The Presenter must submit an attestation request to the HSM to specify the subset of evidence that is required.

Mapping:

- **Endorser**: Manufacturing HSM/FIPS Certification
- **Attester**: HSM protecting the attestation key
- **Presenter**: Code developer
- **Verifier**: CA
- **Relying Party**: Client
- **Evidence:** PKIX Evidence signed by attestation keys
- **Attestation Result:** X.509 certificate for the code signing key.



CRYPTO4A

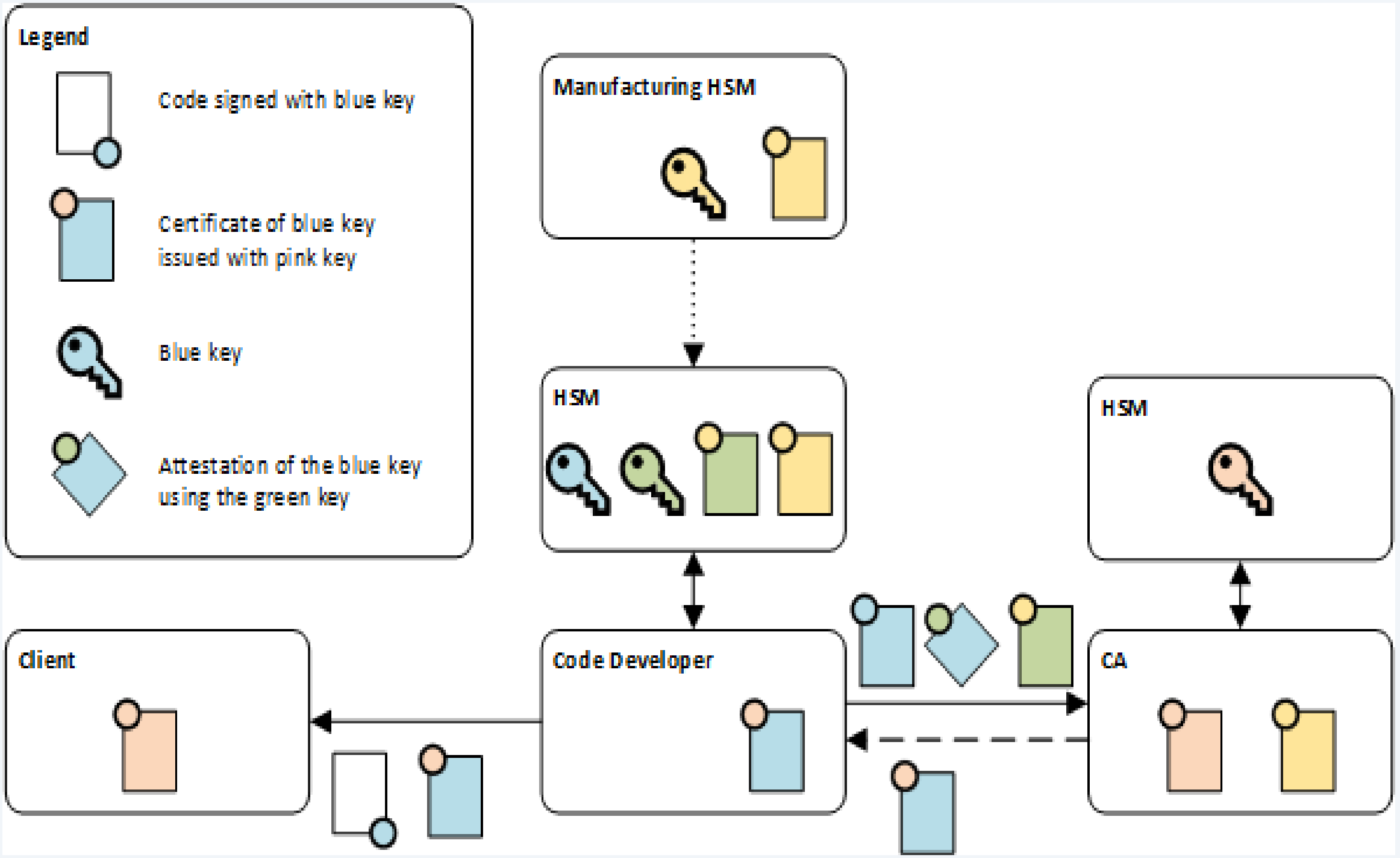
Format of an Attestation

- ASN.1 Encoding
- Evidence is the "to-be-signed" (TBS) structure, similar to X.509
- Allows for multiple independent signatures
- Format is extensible and can include proprietary evidence

```
PkixEvidence ::= SEQUENCE {
  tbs                        TbsPkixEvidence,
  signatures                 SEQUENCE SIZE (0..MAX) of SignatureBlock,
  intermediateCertificates [0] IMPLICIT SEQUENCE of Certificate OPTIONAL
                               -- As defined in RFC 5280
}

TbsPkixEvidence ::= SEQUENCE {
  version            INTEGER,
  reportedEntities SEQUENCE SIZE (1..MAX) OF ReportedEntity
}

ReportedEntity ::= SEQUENCE {
  entityType          OBJECT IDENTIFIER,
  reportedAttributes SEQUENCE SIZE (1..MAX) OF ReportedAttribute
}

SignatureBlock ::= SEQUENCE {
  sid                 SignerIdentifier,
  signatureAlgorithm AlgorithmIdentifier,
  signatureValue      OCTET STRING
}

SignerIdentifier ::= SEQUENCE {
  keyId                  [0] EXPLICIT OCTET STRING OPTIONAL,
  subjectKeyIdentifier [1] EXPLICIT SubjectPublicKeyInfo OPTIONAL,
                           -- As defined in RFC 5280
  certificate            [2] EXPLICIT Certificate OPTIONAL
                           -- As defined in RFC 5280
}
```

Why does attestation need to be quantum-resistant

- **Security:** An attacker will attack the weakest link. Any chain that is not quantum-resistant is susceptible to be attacked.
- **Compliance:** CA Browser forum is pushing the industry towards using HSMs to protect code signing keys, and thus there is a need for proving a key was generated in hardware.
- **Crypto-Agility/Cost:** Attestation is a tool provided by HSM vendor which relies on trust anchors. Non-quantum-resistant attestation capabilities imply a future costly transition of hardware devices.



CRYPTO4A

Algorithm Considerations

Algorithm	Signature Size	Public Key Size	Signature Verification Latency and Complexity	Confidence in Security	Key Management Complexity	Suitability
LMS	~2 kB	56 bytes	Low	Highest	High	• Small code size for verification logic • Systems that are difficult to transition • Key operators that have the resources to manage the state complexity
SLH-DSA	~40 kB	64 bytes	High	High	Low	• Medium code size for verification logic • Systems that are difficult to transition • Key operators that do not have the resources to manage the complexity
ML-DSA	~6 kB	~2.5 kB	Medium	Medium (relatively new)	Low	• Medium code size for verification logic • Systems that can transition relatively easily • Limited bandwidth or latency • Key operators that do not have the resources to manage the complexity

CRYPTO4A

Roots of Trust and Trust Anchors

- Root of Trust (RoT):
  - Typically, a hardware-based system meant to guarantee the security and the integrity of cryptographic material.
- Trust Anchor (TA):
  - Cryptographic asset (e.g., x509 root certificate, TA certificate, public key) inherently trusted.

CRYPTO4A

Importance of quantum-resistant TAs

- Implementing and maintaining a RoT is not easy.
- TAs must be provisioned at manufacturing time and be immutable thereafter to be trustworthy.
- Deploying quantum-resistant TAs is akin to replacing a hardware-based infrastructure. It’s costly and takes time.
- Crypto-agility is impossible if you can’t rely on TAs to transition your systems.

CRYPTO4A

Other Use Cases for Attestation

- Secure audit logs.
  - Implement “blockchain-style” append-only logs. Having the chain of logs allows one to verify the integrity from the start to the end.
  - Attestation is needed to confirm the head of the chain and associate it to a particular device.
- C2PA.
  - In a digital world where “real” images and AI-generated ones are becoming increasingly hard to distinguish, attestation can be used to prove the authenticity of a picture.
- Automatic Clustering.
  - Securely transferring keys to another HSM can either be achieved through a resource-intensive key ceremony, or it could rely on attestation to establish confidence in the transport key.

Questions?

- References:
  - https://datatracker.ietf.org/doc/rfc9334/
  - https://ietf-rats-wg.github.io/key-attestation/draft-ietf-rats-pkix-key-attestation.html
  - https://www.ietf.org/archive/id/draft-ietf-rats-reference-interaction-models-14.html