Use of Attestation with Certificate Signing Requests (CSRs)

draft-ounsworth-csr-attestation-00

Mike Ounsworth, Hannes Tschofenig

(Heavy contributions from MCR, MSJ, Carl Wallace)
Motivation

• CA/B Ballots [CSC-13](#) and [CSC 17](#) require [code signing](#) certificates to be generated and stored in HSMs by 1 June 2023.

• CSR may contain “assertions” about the storage properties of the private key as well as the platform security, such as firmware version, hardware security features, security settings, ...
Status

• Draft published as result of the “PKIX Attestation” design team work by HSM vendors: Entrust (nShield), Crypto4A, Utimaco, Thales CAs: Entrust, Digicert, Keyfactor


• Used several input documents:
  • draft-ietf-lamps-key-attestation-ext-00
  • draft-ounsworth-pkix-key-attestation-02
  • draft-stjohns-csr-attest-00

• Work mode: Regular conference calls + emails
IETF RATS Architecture
(Passport Model)

Terminology reused from RFC 9334. Nonce is optional.

Focus of draft-tschofenig-lamps-nonce-for-cmp-00

Focus of draft-ounsworth-csr-attestation-00
We are not running out of Attestation Technologies...

<table>
<thead>
<tr>
<th>Tokens</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yubico</td>
<td>✓</td>
<td>X.509</td>
</tr>
<tr>
<td>Trusted Platform Module</td>
<td>✓</td>
<td>TPMS_ATTEST/PKCS#10</td>
</tr>
<tr>
<td>Century Longmai PKI Token</td>
<td>☓</td>
<td>CMS/PKCS#7</td>
</tr>
<tr>
<td>TrustSec SLCOS - Bio/PKI token</td>
<td>☓</td>
<td></td>
</tr>
<tr>
<td>Other Devices</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple iOS</td>
<td>✓</td>
<td>CBOR/WebAuth</td>
</tr>
<tr>
<td>Android</td>
<td>✓</td>
<td>ASN.1</td>
</tr>
</tbody>
</table>

See [Remote Key Attestation | PKI Consortium](#).
CSR carries Evidence

- There are lots of different attestation technologies, such as TPM, Arm PSA, DICE, ...
  - Some standardized – some are proprietary.
  - Different mechanisms used for freshness (e.g. nonces, timestamps, epochs)
- CSR does not need to care about the details of the attestation technology
  - It treats evidence as an opaque blob.
  - Attester and Verifier need to understand the evidence format. RA/CA do not need to understand it.
Conveying evidence

• CSR contains a type – value pair.
  • Type indicates what attestation technology is used.
  • Value carries the data produced by the attester (in whatever format used by the attestation technology)

• **Open Issue:** How should the type be encoded:
  • OID
  • **Conceptual Message Wrapper developed in RATS** group (see [issue#11](#)). Integer with values registered in IANA registry.
  • The design group started with unanimous support for “*Let’s stay within ASN.1 / PKIX modules as much as possible*” – which implies a preference for OID-based type field.
Other Open Issues
https://github.com/lamps-wg/csr-attestation/issues

• Currently we define two attributes:
  • AttestAttribute
  • AttestCertsAttribute

• Should the AttestAttribute contain the certification chain need to verify it?
  • Aside: many attestation formats (ex. WebAuthn) carry their certs internally, so this is to support formats that don’t (ex. TPM 2.0 attest)
  • Pro: Easier to associate certs to attestation.
  • Con: Duplication if CSR contains multiple attestations with overlapping cert chains.

• Size: We’re going to embed multiple potentially long cert chains inside a CSR? Won’t that make the CSR huge?
  • Yes.
Next Steps

• Asking the working group to adopt
draft-ounsworth-csr-attestation-00 and drop
  • draft-ietf-lamps-key-attestation-ext-00
  • draft-ounsworth-pkix-key-attestation-02
  • draft-stjohns-csr-attest-00
  • (we believe Carl, Sean, MSJ consider this is a friendly replacement, but need
  confirmation from them).

• Design team will continue its work
  • Next step / deliverable: what “evidence“ is applicable across all “big iron“
    HSMs; “FIPS mode“, “Non-exportable“, “Dual-control“, etc.
  • Hopefully this leads to defining an attestation statement format designed for
    HSMs.