Chameleon Certificates

https://datatracker.ietf.org/doc/draft-bonnell-lamps-chameleon-certs/

IETF 117 – San Francisco
Corey Bonnell, John Gray, David Hook, Tomofumi Okubo, Mike Ounsworth
Problem Statement

• Subjects often have multiple related certificates which certify different keys
  • Algorithm migration (one “legacy” algorithm key and one “new” algorithm key)
  • Dual use (signing key and encryption key)
• How do relying parties obtain the most appropriate certificate?
• How can multiple certificates be transported in a protocol which expects a single certificate?
Previous Approach

• Hybrid certificates as originally defined in draft-truskovskyt-lamps-pq-hybrid-x509 (and later standardized in X.509 10/2019)
  • Alternate public key and signature algorithm and signature value conveyed in extensions
• Until recently, encumbered with patents
• Signature calculation is complex, with little upside
  • The signature (algorithm) field of the To-Be-Signed structure is omitted from the signature calculation, which necessitates new ASN.1 structures for each To-Be-Signed structure
• Requires modification of the certification path validation algorithm
• No way to separate out single-key certificates to decrease size
An alternative approach

• Two X.509 certificates with different public keys
  • Different serial numbers
  • No modification to the certification path validation algorithm is needed
• One of the certificates is issued ("Delta Certificate")
• An extension ("Delta Certificate Descriptor") which contains the certificate’s signature value and differences between the other certificate ("Base Certificate") is created
• This extension is added to the Base Certificate
  • Relying parties can recreate the other certificate using the fields in the Base Certificate and the extension value
<table>
<thead>
<tr>
<th>Traditional Certificate</th>
<th>PQ Certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject:</strong> cn=joe</td>
<td><strong>Subject:</strong> cn=joe</td>
</tr>
<tr>
<td><strong>Issuer:</strong> CACorp</td>
<td><strong>Issuer:</strong> CACorp</td>
</tr>
<tr>
<td><strong>SPKI:</strong> { alg: RSA</td>
<td><strong>SPKI:</strong> { alg: Dilihium3</td>
</tr>
<tr>
<td>key: 8eff38e8... }</td>
<td>key: 79eb4a2... }</td>
</tr>
<tr>
<td><strong>Extensions:</strong></td>
<td><strong>Extensions:</strong></td>
</tr>
<tr>
<td><strong>SANs:</strong> joe.com</td>
<td><strong>SANs:</strong> joe.com</td>
</tr>
</tbody>
</table>

**Core idea – naïve parallel certificates**
## Reconstructing a Delta Certificate from a Base Certificate

**Base Certificate**
- **Subject**: cn=joe
- **Issuer**: CACorp
- **Serial**: 07
- **SPKI**: `{ alg: Dilithium3
  key: 79eb4a2... }`
- **Extensions**:
  - **SANs**: joe.com
- **DCD**:
  - **Serial**: 08
  - **SPKI**: `{ alg: RSA
    key: 8eff38e8... }`

**Delta Certificate**
- **Subject**: cn=joe
- **Issuer**: CACorp
- **Serial**: 08
- **SPKI**: `{ alg: RSA
  key: 8eff38e8... }`
- **Extensions**:
  - **SANs**: joe.com
We have made opinionated choices about what Certificate fields are allowed to be in the diff:

\[
\text{DeltaCertificateDescriptor} ::= \text{SEQUENCE}\{\text{serialNumber CertificateSerialNumber,}\}
\]

\[
\text{signature} [0] \text{IMPLICIT AlgorithmIdentifier}\{\text{SIGNATURE_ALGORITHM, {...}}\} \text{OPTIONAL,}\]
\[
\text{issuer} [1] \text{IMPLICIT Name OPTIONAL,}\]
\[
\text{validity} [2] \text{IMPLICIT Validity OPTIONAL,}\]
\[
\text{subject} [3] \text{IMPLICIT Name OPTIONAL,}\]
\[
\text{subjectPublicKeyInfo SubjectPublicKeyInfo,}\]
\[
\text{extensions} [4] \text{IMPLICIT Extensions\{CertExtensions\}} \text{OPTIONAL,}\]
\[
\text{signatureValue BIT STRING}\}
\]
Relying party use

• If the client in an online protocol can signal its supported algorithms, then the server can selectively send either the Base Certificate or Delta Certificate. In this case, the client need not implement this specification.

• Alternatively, a relying party can selectively extract the Delta Certificate as needed.

• No modification to the certification path validation algorithm is required – only changes to certification path discovery/construction.

• Allows the Base Certificate algorithm to be seamlessly migrated to the Delta Certificate without the legacy overhead.
Running code, next steps

• Four implementations so far (Rust, 2 Java, Python)

• Feedback on the draft, suggestions for improvement
  • Thanks to those who have provided their feedback thus far

• Call for adoption