An Efficient Certificate Format for ECC

Warwick Ford and Yuri Poeluev

TrustPoint Innovation Technologies

June 2015
Overview

• Background
• X.509 Certificate Size
• The M2M Certificate Format – Design Plan
• Specific Optimizations
• Size Comparisons X.509 and M2M
• Status and Next Steps
Background

- Driving application: Near Field Communications (NFC)
  - Short range wireless; Touch transactions – tag to mobile device
  - Signatures (with certificates) needed for authentication
  - Tags have very limited storage, bandwidths very limited
  - An obvious application for ECC
    - But X.509 certificates overloaded the protocols
- Need to prune down X.509 certificate size
- This need common for many emerging constrained protocols
- Decision to design an application-independent cut-down certificate – the Machine-to-Machine (M2M) certificate
X.509 Certificate Size

- X.509 designed in 1990s
  - Very broad specification with many options and extensibility
- X.509 was developed in the RSA/DSA era
  - For 2048 bit, typical certificate 790 bytes (65% crypto fields)
  - Proposals to shorten certificates did not get off the ground
- ECC reduced key size 9x
  - For 224 bit, typical certificate 360 bytes (25% crypto fields)
- Elliptic Curve Qu-Vanstone (ECQV) reduces crypto sizes further
  - Only one ECC point per cert (15% crypto fields)
- There is a compelling case for cut-down X.509 today
M2M Certificate Format – Design Plan

• Keep with X.509 semantics and security properties
• Continue to support the X.509 features that are in common use
  • The SECG SEC4 MES format for ECQV failed in this respect
• Support both ECDSA and ECQV (and RSA/DSA)
• Reign-in extensibility
• Eliminate redundancies
• Build-in any other obvious field optimizations
• Stick with ASN.1
  • Multiple variable length fields are needed
  • Potential for code reuse, interworking with X.509
Specific Optimizations 1

- Limit DN names to RFC 5280 mandatory attributes plus a few others in common use
- Only one of each attribute, no more than 4 total, no multi-level names
- An attribute has a fixed character encoding (usually UTF8 or IA5)
- Modest length constraints on name fields
- Use UNIX time not ASN.1 time (adopted from SEC4 MES)
- Drop redundant outer envelope algorithm id
Specific Optimizations 2

- Closed set of built-in extensions (RFC 5280 mandatories plus a couple more)
  - Issuer & subject key ids, key usage (7 bits), cert policies (1 OID)
  - Subject & issuer alt name, ext key usage (1 OID), auth info access
  - Basic constraints
- No criticality – implied by semantics
- Parameter inheritance:
  - When certificate is transmitted with its superior certificate, omit issuer name and inherit it from the superior
## Comparative Certificate Sizes

<table>
<thead>
<tr>
<th>Certificate size in bytes (All 224-bit ECC)</th>
<th>ECDSA X.509</th>
<th>ECDSA M2M</th>
<th>ECDSA M2M with parameter inheritance</th>
<th>ECQV X.509</th>
<th>ECQV M2M</th>
<th>ECQV M2M with parameter inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Entity Small</td>
<td>241</td>
<td>155</td>
<td>136</td>
<td>177</td>
<td>89</td>
<td>70</td>
</tr>
<tr>
<td>End Entity Medium</td>
<td>364</td>
<td>218</td>
<td>189</td>
<td>300</td>
<td>152</td>
<td>123</td>
</tr>
<tr>
<td>CA Certificate</td>
<td>338</td>
<td>207</td>
<td>N/A</td>
<td>274</td>
<td>134</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- **Small**: 1-component 8-char names. Extensions: key usage
- **Medium**: 2-component 16-char names. Extensions: key usage, cert policy, 20-char OCSP URL, 10-char subject alt name
- **CA Certificate**: 2-component 16-char names. Extensions: key usage, basic constraints, 20-char OCSP URL
Comparative Certificate Sizes

<table>
<thead>
<tr>
<th>Certificate size in bytes (All 224-bit ECC)</th>
<th>ECDSA X.509</th>
<th>ECDSA M2M</th>
<th>ECDSA M2M with parameter inheritance</th>
<th>ECQV X.509</th>
<th>ECQV M2M</th>
<th>ECQV M2M with parameter inheritance</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Entity Small</td>
<td>241</td>
<td>155</td>
<td>136</td>
<td>177</td>
<td>89</td>
<td>70</td>
</tr>
<tr>
<td>End Entity Medium</td>
<td>364</td>
<td>218</td>
<td>189</td>
<td>300</td>
<td>152</td>
<td>123</td>
</tr>
<tr>
<td>CA Certificate</td>
<td>338</td>
<td>207</td>
<td>N/A</td>
<td>274</td>
<td>134</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- ECDSA: **M2M 40% smaller than X.509**
  - 45% with parameter inheritance
- ECQV: **M2M 50% smaller than X.509**
  - 60% with param inheritance
Status and Next Steps

• M2M has been adopted by NFC Forum for tag signature infrastructure
  • Included in Signature Record Type Definition
• M2M has been submitted to SECG
  • Proposed draft revision of SEC4
• M2M has been published as an IETF Internet-Draft
  • But there is no WG with a charter to standardize a general purpose certificate
  • When format is published, can include as an option in TLS/DTLS
• Seeking suggestions for other standardization vehicles
For More Information

Warwick Ford, TrustPoint Innovation, wford@wyltan.com

• NFC Reference:

• SECG Reference:
  • Draft SEC4: Elliptic Curve Qu-Vanstone Implicit Certificates, Draft Version 1.2. www.secg.org/draft-sec4-1.2.pdf

• Internet-Drafts:
  • Certificate definition: draft-ford-m2mcertificate-00
  • TLS/DTLS Use of M2M: draft-ypoeluev-tls-m2m-certs-00