Entity Attestation Token (EAT)

Laurence Lundblade

July 2018
Entity Attestation Token

- Chip & device manufacturer
- Device ID (e.g. serial number)
- Boot state, debug state…
- Firmware, OS & app names and versions
- Geographic location
- Measurement, rooting & malware detection…

All Are Optional

Cryptographically secured by signing

Banking risk engine
IoT backend
Network infrastructure
Car components
Enterprise auth risk engine
Electric company
End-End Attestation Flow

Entity (e.g., Chip, Device...)

- Claims data
- Private key for signing. Stored securely on device
- Token Creation and Signing

Entity Manufacturer (e.g. chip or device vendor)

- Provision private keys during manufacturing
- Verification Keys

Relying Party (e.g., Server / Service)

- EAT Token
- Verification Process
- Verified claims
- Processing such as a payment or authentication risk engine

Self-secured token carried in protocol messages

Other flows are possible where verification is done by a service or by the entity vendor.
## EAT Format

### Overall structure: COSE_Sign1

<table>
<thead>
<tr>
<th>Protected headers</th>
<th>Unprotected headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm -- Examples: ECDSA 256, RSA 2048, ECDAA</td>
<td>Key ID -- identifies the key needed to verify signature</td>
</tr>
<tr>
<td>Signing Scheme -- Examples: IEEE IDevID, EPID, X.509 Hierarchy</td>
<td>Certs (optional) -- to chain up to a root for some signing schemes</td>
</tr>
</tbody>
</table>

### Signed payload

- CBOR formatted map of claims that describe device and its disposition
- Few and simple or many, complex, nested...
- All claims are optional -- no minimal set
- The format and meaning of a basic set of claims should be standardized for interoperability
- Should be adaptable to cover many different use cases from tiny IoT devices to complex mobile phones
- Privacy issues must be taken into account

### Signature

- Examples: 64 byte ECDSA signature, 256 byte RSA signature

### Summary

- COSE format for signing
- Small message size for IoT
- Allows for varying signing algorithms, carries headers, sets overall format
- CBOR format for claims
- Small message size for IoT
- Labelling of claims
- Very flexible data types for all kinds of different claims.
- Translates to JSON
- Signature proves device and claims (critical)
- Accommodate different end-end signing schemes because of device manufacturing issues
- Privacy requirements also drive variance in signing schemes
## Similar and Related Technologies

<table>
<thead>
<tr>
<th>Technology</th>
<th>Use Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIDO Attestation</td>
<td>Attestation of FIDO Authenticator implementations</td>
</tr>
<tr>
<td>Android Key Store</td>
<td>Attestation key pairs in the key store</td>
</tr>
<tr>
<td>NEA</td>
<td>Collect and send endpoint security posture (e.g. anti-virus SW state and config) to enterprise collection / monitoring point</td>
</tr>
<tr>
<td>RATS / NSF</td>
<td>Attestation / Measurement of SW on Network Security Functions (e.g., firewalls)</td>
</tr>
<tr>
<td>TPM</td>
<td>Attestation / Measurement of SW running on a device</td>
</tr>
<tr>
<td>BRSKI / Zero Touch</td>
<td>Authenticates IoT devices for enrollment in IoT management system</td>
</tr>
</tbody>
</table>