Security in oneM2M Release 2

Wolfgang Granzow* and Phil Hawkes
*Senior Director Technology at Qualcomm Germany
oneM2M Security Working Group member
< wgranzow@qti.qualcomm.com >
Recap on the oneM2M Architecture

- Hierarchical topology of M2M Nodes
  - Nodes comprised of Common Services Entities (CSE) and Application Entities (AE)
  - Field Domain: Middle Nodes, Application Dedicated Nodes, Application Service Nodes
  - Infrastructure Domain: Infrastructure Node
<table>
<thead>
<tr>
<th>Challenge</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Huge variety of deployment scenarios</strong></td>
<td>Secure Communications</td>
</tr>
<tr>
<td>vulnerabilities depend on environment</td>
<td>content confidentiality, integrity</td>
</tr>
<tr>
<td>w/o infrastructure for credential mgmt.</td>
<td>variety of authentication scenarios</td>
</tr>
<tr>
<td>Intermediate nodes trusted</td>
<td>Secure comm. between adjacent nodes</td>
</tr>
<tr>
<td>Intermediate nodes not trusted</td>
<td>Secure end-to-end communications</td>
</tr>
<tr>
<td></td>
<td>Secure Environment API (in progress)</td>
</tr>
<tr>
<td><strong>Any IoT device in any deployment</strong></td>
<td>Interoperable Remote provisioning</td>
</tr>
<tr>
<td>Off-the-shelf vs. configurable devices</td>
<td>A variety of underlying credentials.</td>
</tr>
<tr>
<td>On-boarded by professionals vs.</td>
<td>Provisioning symmetric keys</td>
</tr>
<tr>
<td>unskilled users</td>
<td>Provisioning certificates (in progress)</td>
</tr>
<tr>
<td><strong>Data privacy</strong></td>
<td>Access Control</td>
</tr>
<tr>
<td>M2M Device cannot make</td>
<td>Access Control Policies</td>
</tr>
<tr>
<td>autonomous judgement calls</td>
<td>Authorization services, including</td>
</tr>
<tr>
<td></td>
<td>Token-based Access Control</td>
</tr>
<tr>
<td></td>
<td>Role-based Access Control</td>
</tr>
<tr>
<td></td>
<td>Privacy policy management</td>
</tr>
</tbody>
</table>
oneM2M Security Frameworks

- Tie together credential management, configuration parameters, establishing security session (e.g. TLS/DTLS handshake) and protecting the messages or data.

**Security Association Establishment Framework (SAEF):** Adjacent entities

**End-to-End Security of Primitive (ESPrim):** Originator ↔ Hosting CSE

**End-to-End Security of Data (ESData):** Data producer to data consumer

---

MN-CSE can see and alter message. What if it is not trusted?

**Legend:**
- **SA**: Security Association
- **ADN**: Application Dedicated Node
- **MN**: Middle Node
- **IN**: Infrastructure Node

MN cannot see or alter messages.
oneM2M Security Frameworks

- Tie together credential management, configuration parameters, establishing security session (e.g. TLS/DTLS handshake) and protecting the messages or data
  - Security Association Establishment Framework (SAEF): between adjacent entities
  - End-to-End Security of Primitive (ESPrim): Originator ↔ Hosting CSE
  - End-to-End Security of Data (ESData): Data producer to data consumer

Protect using ESData

What if IN-CSE is not trusted with this app data?

IN-AE uses using ESData to extract app data

Protected using ESData. IN-CSE cannot see or alter app data

MN cannot see or alter messages

(opt) ESPrim

Legend:
- SA  Security Association
- ADN  Application Dedicated Node
- MN  Middle Node
- IN  Infrastructure Node
Message Security between adjacent devices

- Uses (Datagram) Transport Layer Security Protocols, TLS/DTLS Version 1.2
- Several Security Association Establishment Frameworks are supported:
  1) Authentication and session key establishment using **symmetric keys** shared by devices
  2) Authentication and session key establishment using **Certificates** provisioned to devices
  3) Authentication facilitated by an **M2M Authentication Function (MAF)** hosted by M2M-SP or third-party
     - The MAF authenticates the end-points (PSK or certificates) and facilitates establishing a symmetric key
Remote Security Provisioning Frameworks (RSPF)

- M2M device is preconfigured with credentials to establish SA with a remote provisioning server (M2M Enrolment Function, MEF)
  - MEF is operated by trusted 3rd party (device manufacturer, underlying network operator) or M2M Service Provider
  - MEF can be the Bootstrapping Server Function (BSF) of 3GPP Generic Bootstrapping Architecture (GBA)
- MEF can provision symmetric keys or (work in progress) certificates
E2E Protection of primitives (“ESPrim”)  

- Interoperable framework for securing oneM2M primitives  
  - CSEs (forwarding the primitive) do not need to be trusted  
  - ESPrim provides mutual authentication, confidentiality, integrity protection and a freshness guarantee (bounding the age of ESPrim Objects).  
  - Protocol: JSON Web Encryption (JWE) using a symmetric key  
    - Symmetric key can be established by pre-provisioning (using MEF), End-to-end Certificate-based Key Establishment (ESCertKE), or central authentication server (MAF)
E2E Protection of selected data ("ESData")

- interoperable framework for protecting a selected data portion of a primitive
  - data to be protected is called the *ESData Payload*.
  - transited CSEs do not need to be trusted with that data.
  - ESData payload could typically compose all or part of an attribute value (e.g. *content* attribute value of a `<contentInstance>` resource) or a primitive parameter (e.g. a signed, self-contained access token communicated in a request primitive to obtain dynamic authorization).
  - Protocol: JSON Web Encryption/Signature (JWE/JWS) or XML Encryption/Signature
Authorization using Access Control Lists

- Access control rules define who can do what under which circumstances

![Diagram]

- links
- contains
- with conditions on

**WHO:** entities or roles  
CSE-ID  
AE-ID  
Role-ID

**WHAT:** operations  
Create,  
Retrieve  
Update  
Delete  
Notify  
Discover

**WHICH:** contexts  
Time  
Location  
IP address  
Authentication Status  
Object Details
Dynamic Authorization

- **Dynamic Authorization**: Originator or Hosting CSE requesting authorization of Originator – provided by a Dynamic Authorization System (DAS) Server
  - Direct Dynamic Authorisation: Hosting CSE submits request to DAS, Originator is not involved
  - Indirect Dynamic Authorisation: Originator submits request to DAS Server using info provided by Hosting CSE. Similar to OAuth. See figure below.
  - DAS has multiple options for authorizing: Issue/update access control rules, assign Role(s) to the Originator, issue JSON Web Tokens (JWT)

 Req/resp: steps 1,2,5,8
 Direct DA: steps 6,7
 Indirect DA: steps 2-5 & (opt) 6,7
Privacy Policy Manager

- Manages access to Personally Identifiable Information (PII) on behalf of users
- Uses a “Terms and Condition’s Mark-up language”

**KEY**
- When User and ASP Register w/ PPM
- When User subscribes for ASP service
- When ASP requests PII