IoT Device Management Security

Mikko Saarnivala
Senior Director IoT Architecture / ARM

ARM Tech Symposia India
December 7th 2016

©ARM 2016
Agenda

- Device management and the LWM2M architecture
- Gentle introduction to LWM2M
- Securing LWM2M and bootstrapping
- Outlook
- Summary
Agenda

- Device management and the LWM2M architecture
- Gentle introduction to LWM2M
- Securing LWM2M and bootstrapping
- Outlook
- Summary
Diverse IoT deployments with common needs

- Service provisioning
- Key management
- Provisioning of access control lists
- Update application and system software
- Apply bug fixes and add new features
- Changes to settings
- Trigger actuators
- Report errors from devices
- Query status of devices
- Notify changes in sensor values
- Retrieve configuration settings and device status
LWM2M version 1.0 architecture

Figure 1: Entities in the LWM2M Architecture.

Figure 2: Protocol Stack
Agenda

- Device management and the LWM2M architecture
- **Gentle introduction to LWM2M**
- Securing LWM2M and bootstrapping
- Outlook
- Summary
The LWM2M RESTful API

High-level message pattern hiding details of networking and security protocols

• **Bootstrap interface**
  – Configure servers info, credentials & ACLs

• **Registration interface**
  – Informs server about “existence” and supported functionality (e.g., objects, transport bindings)

• **Device management & service enablement interface**
  – Ability to access object instances and resources

• **Information reporting interface**
  – Publish/subscribe interaction for observing changes in resources.
Building blocks for LWM2M version 1.0

**CoAP**
- Specified in RFC 7252 and uses UDP.
- Short, binary header.
- Designed for small data transmissions but capable of transferring large data as well with [RFC 7959](https://tools.ietf.org/html/rfc7959).
- Built-in support for discovery.
- Lots of [open source implementations](https://github.com/coap) available.

**DTLS**
- Specified in [RFC 6347](https://tools.ietf.org/html/rfc6347) and builds on TLS 1.2.
- Offers communication security by providing confidentiality, integrity and authentication.
- Performance depends on selected ciphersuite and settings.
- The full list of standardized ciphersuites can be found at [IANA](https://www.iana.org/assignments/dtls-cipher-suites).

**Object Model**
- Objects add functionality for security foundation and for applications.
- Large number of objects defined and listed in [repository](https://github.com/ARM-lwm2m/examples).
Object model

- Objects/Resources are accessed with simple URIs: `/{Object ID}/{Object Instance}/{Resource ID}`

- Example:

```
3303  0  5700
```

Temperature Sensor | Instance #0 | Sensor Value (last or current value measured from a sensor)
Objects

The LWM2M technical specification itself defines eight objects; the repository contains many more contributed by IPSO alliance, oneM2M, and from vendors.

<table>
<thead>
<tr>
<th>Object Name</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWM2M Security</td>
<td>0</td>
<td>Keying material of a LWM2M Client to access a LWM2M server.</td>
</tr>
<tr>
<td>LWM2M Server</td>
<td>1</td>
<td>Data related to a LWM2M server.</td>
</tr>
<tr>
<td>Access Control</td>
<td>2</td>
<td>Information used to check whether a LWM2M Server has access to object.</td>
</tr>
<tr>
<td>Device</td>
<td>3</td>
<td>Device related information, including device reboot and factory reset function.</td>
</tr>
<tr>
<td>Connectivity Monitoring</td>
<td>4</td>
<td>Parameters related to network connectivity.</td>
</tr>
<tr>
<td>Firmware</td>
<td>5</td>
<td>Capability to update firmware</td>
</tr>
<tr>
<td>Location</td>
<td>6</td>
<td>Device location information</td>
</tr>
<tr>
<td>Connectivity Statistics</td>
<td>7</td>
<td>Information like transmit and receive counters</td>
</tr>
</tbody>
</table>
Limiting notifications

- Publish/subscribe is great but *think about bandwidth consumption*.
- Information reporting interface offers features to limit the amount of reported information.

**Time-based limits** with minimum period and maximum period:
- Pmin – Minimum time to wait (in seconds) before sending a new notify
- Pmax – Maximum time in seconds between two consecutive notify

**Threshold-based limits** with less than and greater than:
- Lt – Low limit measurement notification, like low alarm
- Gt – High limit measurement notification, like a high alarm

**Limits based on rate of change:**
- St – Minimum delta change required to notify
Agenda

- Device management and the LWM2M architecture
- Gentle introduction to LWM2M
- **Securing LWM2M and bootstrapping**
- Outlook
- Summary
Securing LWM2M

- IoT devices may exchange sensitive information and need to be protected.
- IoT end points need to be authenticated and communication requires protection.
- LWM2M supports **three credential types**:
  - Pre-shared secrets,
  - Raw public keys, and
  - Certificates
- Relies on AES-128 and ECC
- DTLS requires a **random number generator**.
  - The danger is that there is little (to no) randomness in embedded systems.
  - Paul Bakker talks about “Entropy Requirements in IoT” on the [ARM mbed Youtube channel](https://www.youtube.com/watch?v=example_video_id).
Bootstrapping architecture

- LWM2M client needs credentials to securely communicate with the LWM2M server using DTLS. Configuration and access rights might change.

Where does this information come from?

IoT Device

LWM2M client (C)
- ACLs
- Config
- Device credentials

LWM2M server
- Device credentials

Specification several deployment choices:
- Factory bootstrap
- Bootstrap from smartcard
- Client initiated bootstrap
- Server initiated bootstrap
Agenda

- Device management and the LWM2M architecture
- Gentle introduction to LWM2M
- Securing LWM2M and bootstrapping
- Outlook
- Summary
Outlook

- Version 1.0 is being finalized now and work on version 1.1 has been started.
- Envisioned additions:
  - Support for CoAP over TCP/TLS
  - Protocol gateway support
  - Support for additional transports, such as HTTP/2.
  - Additional security features, such as DTLS IoT profile compliance, TLS 1.3 support, application layer security based on COSE, and new bootstrapping modes.
  - Support for low power WANs, such as the 3GPP NB-IoT.
Agenda

- Device management and the LWM2M architecture
- Gentle introduction to LWM2M
- Securing LWM2M and bootstrapping
- Outlook
- Summary
Summary

1. LWM2M is a big step in standardization that will simplify secure IoT device management.
2. It re-uses proven internet security technologies.
3. You can get involved in standardization and contribute code today.