Secure and Authorized Client-to-Client Communication for LwM2M

Leandro Lanzieri¹, Peter Kietzmann¹, Thomas C. Schmidt¹, Matthias Wählisch²

RIOT Summit 2023
18th September 2023 – Frankfurt, Germany

leandro.lanzieri@haw-hamburg.de
IoT Challenges

- Vendor Incompatibility
- Security Requirements
- Device Constraints
- Edge Collaboration
IoT Challenges

LwM2M

Vendor Incompatibility
Security Requirements
Device Constraints
Edge Collaboration
IoT Challenges

LwM2M

Vendor Incompatibility
Security Requirements
Device Constraints
Edge Collaboration

Our LwM2M Extensions
LwM2M Overview
LwM2M Protocol Stack
LwM2M Protocol Stack

- LwM2M
- CoAP
- OSCORE
- DTLS
- UDP
- SMS
- CIoT
- LoRa
- WAN
- TCP
- HTTP
- MQTT
- TLS
- TCP

Used stack
LwM2M Background

- IoT Device management
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates
LwM2M Background

- IoT Device management
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates

- Two main entity types
  - LwM2M Clients (IoT devices)
  - LwM2M Servers
LwM2M Background

- **IoT Device management**
  - Semantic interoperability across vendors
  - Resource access control
  - Bootstrapping and software updates

- **Two main entity types**
  - LwM2M Clients (IoT devices)
  - LwM2M Servers

- **Only Servers perform operations on Clients**
  - Using established secure communication
  - Credentials and access rights are required
  - IoT applications interact with Clients only through Servers
LwM2M Background

- IoT Device management
- Semantic interoperability across vendors
- Resource access control
- Bootstrapping and software updates

- Two main entity types
  - LwM2M Clients (IoT devices)
  - LwM2M Servers

- Only Servers perform operations on Clients
  - Using established secure communication
  - Credentials and access rights are required
  - IoT applications interact with Clients only through Servers

Server-Centric communication **prevents edge collaboration.**
LwM2M Objects, Resources and Access Control

- LwM2M Clients expose resources
  - Resources are logically grouped into objects

- Objects accept multiple operations
  - Read, write, execute, create, etc.

- Access control policies apply to objects
  - Determine which operations a server may perform
  - Different servers may have different access
LwM2M
Client-to-Client (C2C) Communication
Extended LwM2M Deployment
Extended LwM2M Deployment

- Clients operate on other Clients resources
Extended LwM2M Deployment

- Clients operate on other Clients resources
- **Application logic is distributed on the edge**
  - Reduces latency
  - Increases bandwidth
  - Local communication
Extended LwM2M Deployment

- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients
Extended LwM2M Deployment

- Clients operate on other Clients resources
- Application logic is distributed on the edge
  - Reduces latency
  - Increases bandwidth
  - Local communication
- Servers monitor and manage Clients
- New LwM2M objects
  - Client object: communication information
  - Client Security object: credentials for secure channel
  - Client Access Control object: remote clients access rights
- Extended interfaces
  - Allow client operation
  - Handle client access rights
Extended LwM2M Deployment

• Clients operate on other Clients resources
• Application logic is distributed on the edge
  • Reduces latency
  • Increases bandwidth
  • Local communication
• Servers monitor and manage Clients

• New LwM2M objects
  • Client object: communication information
  • Client Security object: credentials for secure channel
  • Client Access Control object: remote clients access rights
• Extended interfaces
  • Allow client operation
  • Handle client access rights

Contribution 1
Third Party Authorization of LwM2M Clients
Third Party Authorization of LwM2M Clients

Contribution 2

LwM2M client 1

LwM2M server

LwM2M client 2
Third Party Authorization of LwM2M Clients

Server Hints (Optional)

Unauthorized "Observe" Operation on Resource R

Unauthorized Return Code + Owner Server Hints
Third Party Authorization of LwM2M Clients

**SERVER HINTS (OPTIONAL)**
- Unauthorized "Observe" Operation on Resource R
- Unauthorized Return Code + Owner Server Hints

**CREDENTIAALS & ACCESS INSTALLATION**
- Access Request to Resource R on Client 1
- Install Credentials for Client 2
- Install Credentials and Access Rights for Client 1
Third Party Authorization of LwM2M Clients

- LwM2M client 1
  - Server Hints (Optional)
    - Unauthorized "Observe" Operation on Resource R
    - Unauthorized Return Code + Owner Server Hints
  - Credentials & Access Installation
    - Access Request to Resource R on Client 1
    - Install Credentials for Client 2

- LwM2M server
  - Install Credentials and Access Rights for Client 1

- LwM2M client 2
  - Handshake (only with (D)TLS security)
Third Party Authorization of LwM2M Clients

LwM2M client 1

SERVER HINTS (OPTIONAL)
- Unauthorized "Observe" Operation on Resource R
- Unauthorized Return Code + Owner Server Hints

LwM2M server

CREDENTIALS & ACCESS INSTALLATION
- Access Request to Resource R on Client 1
- Install Credentials for Client 2

LwM2M client 2

Handshake (only with (D)TLS security)

AUTHORIZED OPERATION
- Authorized "Observe" Operation on Resource R
- "Notify" from Resource R
Experimental Evaluation
Experimental Setup

LwM2M Clients
Experimental Setup
Experimental Setup
Experimental Setup
Experimental Setup

- Inria Grenoble site
- 350 nodes evenly spread
Experimental Setup

- Inria Grenoble site
- 350 nodes evenly spread

LwM2M Clients
Experimental Setup

- Inria Grenoble site
- 350 nodes evenly spread

ARM Cortex-M3 @ 72 MHz
64 KiB RAM + 512 KiB ROM
Experimental Setup

- Inria Grenoble site
- 350 nodes evenly spread

IEEE 802.15.4
2.4 GHz transceiver
Firmware Size

![Graph showing firmware size distribution]

- Authorization: 18.8%
- Client-to-client
- Client handling
- Server object
- Security object
- Access ctrl. object
- Utilities
- LwM2M core

![Graph showing RAM size distribution]

- Baseline:
  - 32.6%
Firmware Size
Firmware Size
**Firmware Size**

C2C only requires additional **3% ROM** and **1% RAM**. Authorization an extra **5% ROM**.
Notification Arrival Time

![Graph showing CDF vs Completion time for Server-centric (1 hop)]
Notification Arrival Time

![CDF of Completion time vs. Median Internet delay](image-url)
Notification Arrival Time

![Graph showing CDF of completion time]

- Median Internet delay
- Server-centric (1 hop)
- Server-centric (2 hops)
- Server-centric (3 hops)
- Server-centric (4 hops)
Notification Arrival Time

OSCORE is faster
C2C reduces notification arrival times by 90%.
Experiments using *randomized topologies* show the *same results*. 
Authorization Request & First C2C Operation

![Graph showing CDF vs Completion time [ms]]

- Green line: DTLS credentials distribution
- Red line: OSCORE credentials distribution

CDF (Cumulative Distribution Function)

Completion time [ms]

0 1000 2000 3000 4000 5000 6000 7000
Authorization Request & First C2C Operation

![Graph showing the completion time in milliseconds for different distributions. The x-axis represents completion time in milliseconds, ranging from 0 to 7000, and the y-axis represents the CDF (Cumulative Distribution Function). The graph compares two distributions: DTLS credentials distribution and OSCORE credentials distribution. The CoAP retransmissions are indicated by the green and pink lines, respectively.](image-url)
OSCORE credential distribution is slower due to additional transmitted LwM2M object.
Authorization Request & First C2C Operation

![Graph showing CDF vs Completion time for different operations: DTLS credentials distribution, OSCORE credentials distribution, Initial C2C operation (DTLS), and Initial C2C operation (OSCORE).]
Authorization Request & First C2C Operation
Initial C2C operation is **slower with DTLS due to handshake.**
Maximum Goodput with One Hop

![Graphs showing goodput and delivery rate vs. notification interval for C2C OSCORE and C2C DTLS.]
Maximum Goodput with One Hop

[C2C OSCORE graph showing Goodput [B/s] vs Notification interval [ms].]

[C2C DTLS graph showing Goodput [B/s] vs Notification interval [ms].]

[Server-centric graph showing Delivery rate [%] vs Notification interval [ms].]

Legend:
- Delivery rate at 250 Kbit/s (right axis)
- Theoretical goodput
- Goodput at 250 Kbit/s
C2C goodput is 8 times higher than Server-Centric.
Energy Consumption

- Server-centric (4 hops)
- Server-centric (3 hops)
- Server-centric (2 hops)
- Server-centric (1 hop)

Energy consumption [J]
Energy Consumption

Server-centric (4 hops)
Server-centric (3 hops)
Server-centric (2 hops)
Server-centric (1 hop)
C2C OSCORE
C2C DTLS

Energy consumption [J]
C2C adds no energy overhead.
Using less hops reduces energy requirement.
Conclusion & Outlook

- We contributed
  - A third party authorization mechanism for LwM2M Clients
  - New LwM2M objects and extended interfaces for **C2C communication**
  - An empirical *performance analysis* on real hardware
  - Public and *open source implementation* of the extensions
Conclusion & Outlook

- **We contributed**
  - A third party authorization mechanism for LwM2M Clients
  - New LwM2M objects and extended interfaces for **C2C communication**
  - An empirical performance analysis on real hardware
  - Public and open source implementation of the extensions

- **Our results show that**
  - C2C reduces data arrival times **by up to 90%**
  - C2C yields a more reliable and **8 times higher goodput**
  - Our extensions produce a relatively **small memory footprint**
Conclusion & Outlook

- **We contributed**
  - A third party authorization mechanism for LwM2M Clients
  - New LwM2M objects and extended interfaces for C2C communication
  - An empirical performance analysis on real hardware
  - Public and open source implementation of the extensions

- **Our results show that**
  - C2C reduces data arrival times by up to 90%
  - C2C yields a more reliable and 8 times higher goodput
  - Our extensions produce a relatively small memory footprint

- **In future work we will**
  - Analyse the applicability of ACE-OAuth framework to LwM2M
  - Explore the integration with Group OSCORE for multiple observations
Thank You!

Our code can be found online

https://github.com/inetrg/ipsn-2022-lwm2mc2c