ace
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2016

A. Somaraju
Tridonic GmbH & Co KG
S. Kumar
Philips Research
H. Tschofenig
ARM Ltd.

July 6, 2015

Multicast Security for the Lighting Domain
draft-somaraju-ace-multicast-00.txt

Abstract

Lighting systems have strict requirements on message latency and synchronization (typically latency less than 200 ms and jitter less than 50 ms). There are several lighting use cases that require securing such communication between a (group of) senders and a group of receivers. This draft describes initial ideas for authorization and key management required for the secure group communication within a lighting system.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 7, 2016.

Table of Contents

1. Introduction

There are several lighting related use cases that require securing communication between a (group of) senders and a group of receivers. Often, a set of lighting nodes (e.g. luminaires, wall-switches, sensors) are grouped together into a single "Application Group".

For such use-cases, three requirements need to be addressed:

1. Only authorized members of the application group must be able read and process messages.

2. Receivers of group messages must be able to verify the integrity of received messages as being generated within the group.

3. Usually, message transfer and processing must happen with low latency and in synchronous manner (typically latency less than 200 ms and jitter less than 50 ms).

This document discusses these three issues and describes initial ideas on how they can be addressed.

2. Terminology

This document uses the following terms from [I-D.gerdes-ace-actors]: Authorization Server, Resource Owner, Client, Resource Server. The terms 'sender' and 'receiver' refer to the application layer messaging used for lighting control; other communication interactions with the supporting infrastructure uses unicast messaging.

This document also assumes that the reader is familiar with the OMA Lightweight Machine-to-Machine (LWM2M) specifications [LWM2M] and the IPSO specification [IPSO].

3. Authorization Policy

When implementing an authorization policy two factors need to be considered:

1. The type of resource/service that is being offered by an end node, and

2. The group of nodes that are allowed to access a given type of resource.

The type of resources in the lighting domain can be categorized into multiple (access) levels and these levels are described below. For resources/services that belong to a category less than access level 2, there are multiple clients that need to access the same resource/ service with low latency. The scope of this document is to determine how one can implement authorization policies for group communication for resources/services that belong to access level 2 and below. We first introduce the different access levels and then examine the different types of groups that determine the authorization policy.

3.1. Access Levels

A characteristic of the lighting domain is that access control decisions are also impacted by the type of operation being performed and those categories are listed below. The following access levels are pre-defined.

Level 0: Service detection only

This is a service that is used with broadcast service detection methods. No operational data is accessible at this level.

Level 1: Reporting only

This level allows access to sensor and other (relatively uncritical) operational data and the device error status. The operation of the system cannot be influenced using this level.

Level 2: Standard use

This level allows access to all operational features, including access to operational parameters. This is the highest level of access that can be obtained using (secure) multicast.

Level 3: Commissioning use / Parametrization Services

This level gives access to certain parameters that change the day-to-day operation of the system, but does not allow structural changes.

Level 4: Commissioning use / Localization and Addressing Services

(including Factory Reset) This level allows access to all services and parameters including structural settings.

Level 5: Software Update and related Services

This level allows the change and upgrade of the software of the devices.

Note: The use of group security is disallowed for level higher than Level 2 and unicast communication is used instead.

3.2. Application, multicast and security groups

There are three types of groups that we need to consider:

Application Group:

A lighting application group that consists of the set of all lighting devices that have been configured by a commissioner to respond to events in a consistent manner. For instance, a wall mounted switch and a set of luminaires in a single room might belong to a single group and the switch may be used to turn on/off all the luminaires in the group simultaneously with a single button press. In the remainder of this document we will use GId to identify an application group.

Multicast Group:

A multicast group consists of the set of all nodes that subscribe to the same multicast IP address.

Security Group:

A security group consists of a set of sending and receiving nodes such that any sending node is able to securely send a message to all the receiving nodes. For instance, if symmetric keys are used to secure such messages then every node that has access to the symmetric key is a part of the security group. Every node in a security group can decrypt a message even if it is not addressed for its application group.

Typically, the three groups might not coincide due to the memory constraints on the devices and also security considerations. For instance, in a small room with windows, we may have three application groups: "room group", "luminaires close to the window group" and "luminaires far from the window group". However, we may choose to use only one multicast group for all devices in the room and one security group for all the devices in the room. At the other end of the spectrum, we may have an application group consisting of all the luminaires on a floor consisting of several smaller rooms. In this case, due to security considerations we may choose to not distribute a single key to all the devices on the whole floor. Therefore, the security group might be much smaller (e.g., one per room) and the application floor group is split up into smaller security groups.

The authorization policy must ensure that all the members of a security group are allowed to exchange messages with each other for services which belong to access level less than equal to 2. The services may be accessed via multicast or serial unicast messages between group members. The procedure that is used to determine the security groups based on the availability of multicast groups and the requirements of the application group are out of scope of this document.

4. Architecture

Each node in a lighting application group might be a sender, a receiver or both sender and receiver within the group (even though in Figure 1 below, we show nodes that are only senders or receivers for clarity). The requirement of low latency synchronous behaviour implies most of the communication between senders and receivers of lighting application messages are done using multicast IP messages. On some occasions, a sender in a group will be required to send unicast messages to unique receivers within the same group and the

authorization procedure must also ensure security for such communications. The procedure that is used to determine the security groups based on the availability of multicast groups and the requirements of the application group are out of scope of this document.

Two logical entities are introduced and they have the following function:

Key Distribution Center (KDC): This logical entity is responsible for generating symmetric keys and distributing them to the nodes authorized to receive them. The KDC ensures that nodes belonging to the same security group receive the same key and that the keys are rotated based on certain events, such as key expiry or change in group membership.

Authorization Server (AS): This logical entity stores authorization information about devices, meta-data about them, and their roles in the network. For example, a luminare is associated with different groups, and may have meta-data about its location. This entity may also need to perform user authentication and authorization since access rights may be associated to specific persons. Directly or indirectly the resource owner specifies authorization policies that define which node is allowed to perform which actions.

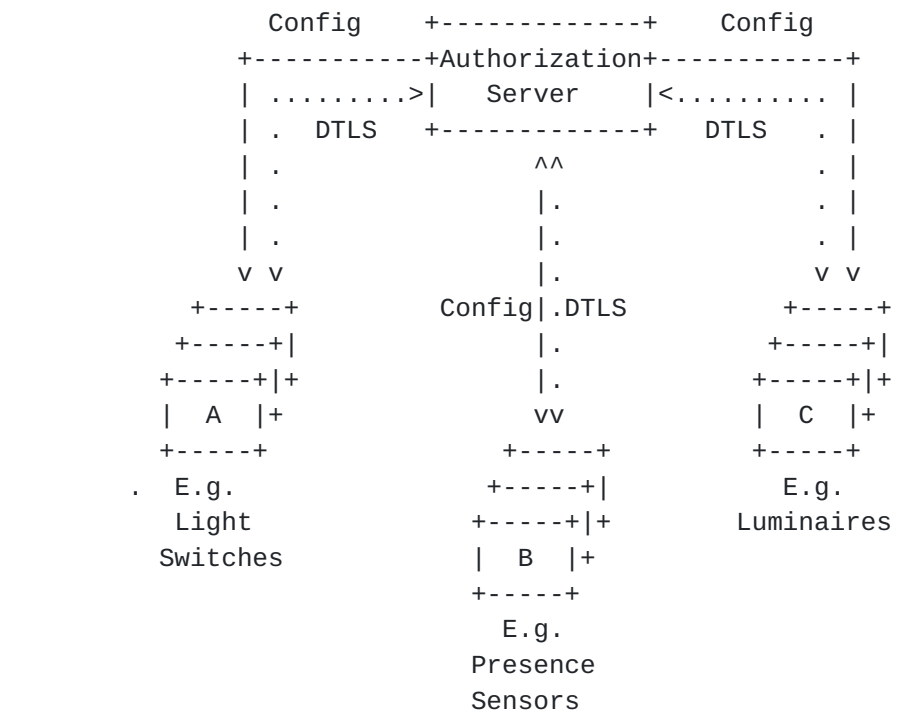
Note that we assume that nodes are pre-configured with device credentials (e.g., a certificate and the corresponding private key) during manufacturing. These device credentials are used in the interaction with the authorization server.

Figure 1 and Figure 2 provide an architectural overview. The dotted lines illustrate the use of unicast DTLS messages for securing the message exchange between all involved parties. The secured multicast messages between senders and receivers are indicated using lines with star/asterisk characters. The security of the multicast messages can be achieved either at the transport level (e.g. [I-D.kumar-dice-multicast-security]) or at the application level with object security (e.g. [I-D.selander-ace-object-security]). The details on multicast security are outside the scope of this draft.

Figure 1 illustrates the information flow between an authorization server and the nodes participating in the lighting network, which includes all nodes that exchange lighting application messages. This step is typically executed during the commissioning phase for nodes that are fixed-mounted in buildings. The authorization server, as a logical function, may in smaller deployments be included in a device carried by the commissioner and only be present during the

commissioning phase. In other use cases, employees using their smartphones to control lights may require an authorization server that dynamically executes access control decisions.

Figure 1 shows the commissioning phase where the nodes obtain configuration information, which includes the AT-KDC. The AT-KDC is an access token and includes authorization claims for consumption by the key distribution center. We use the access token terminology from RFC 6749 [RFC6749] even though it is a solution-specific concept but familiar to many developers. The AT-KDC in this architecture may be a bearer token or a proof-of-possession (PoP) token. Note that a PoP token offers a fair amount of flexibility: with the use of symmetric key cryptography it is comparable to a Kerberos ticket and when used with asymmetric cryptography it can play the role of a certificate. The bearer token concept is described in RFC 6750 [RFC6750] and the PoP token concept is explained in [I-D.ietf-oauth-pop-architecture]. The AT-KDC is created by the authorization server after authenticating the requesting node and contains authorization relevant information. The AT-KDC is protected against modifications using a digital signature or a message authentication code. It is verified in Figure 2 by the KDC.



Legend:

Config (Configuration Data): Includes configuration parameters, authorization information encapsulated inside the access token (AT-KDC) and other meta-data.

Figure 1: Architecture: Commissioning Phase.

In the simplified message exchange shown in Figure 2 a sender requests a security group key and the access token for use with the receivers (called AT-R). The request contains information about resource it wants to access, such as the application group and other resource-specific information -- if applicable, and the previously obtained AT-KDC access token. Once the sender has successfully obtained the requested information it starts communicating with receivers in that group using multicast messages. The symmetric key obtained from the KDC is used to secure the multicast messages for the security group. The AT-R is used to attached to the initial request to authorize the request. The receivers on their side need to perform two steps, namely the receivers themselves need to obtain the necessary group key to verify the incoming messages and they also need information about what resource the sender is authorized to access. Both information can be found in the AT-R access token.

Multicast messages need to be protected such that replay and modification can be detected. The integrity of the message is

protected using a group key. The use of symmetric keys is envisioned here due to latency requirements and the access level level concept is described in Section 3.1. For secure unicast messaging between lighting application group members and the AS or KDC, a topic outside the scope of this document, the sender is assumed to use the DTLS handshake to establish the necessary security context for securing subsequent message interactions.

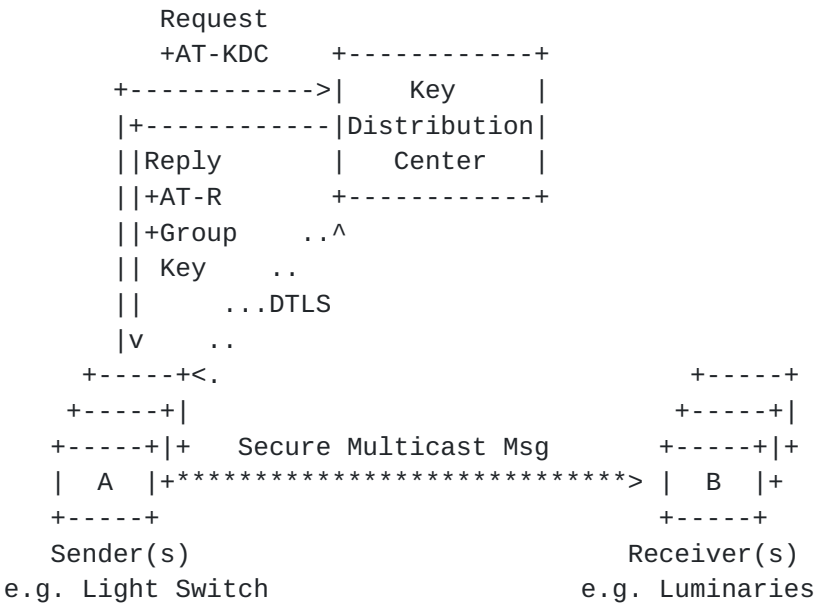


Figure 2: Architecture: Group Key Distribution Phase.

Figure 3 describes the algorithm for obtaining the necessary credentials to transmit a secure multicast message based on the architectural description shown in Figure 1 and Figure 2. When sender wants to send a message to the application group, it checks if it has the group key. If no group key is available then it checks if it has an access token for use with the KDC (AT-KDC). If no AT-KDC is found in the cache then it contacts the authorization server to obtain an access token. Note that this assumes that the authorization server is online, which is only true in scenarios where granting authorization dynamically is supported. In the other case where the AT-KDC is already available the sender contacts the KDC to obtain a group key. If a group key is already available then the sender can transmit a message secured to the group immediately.

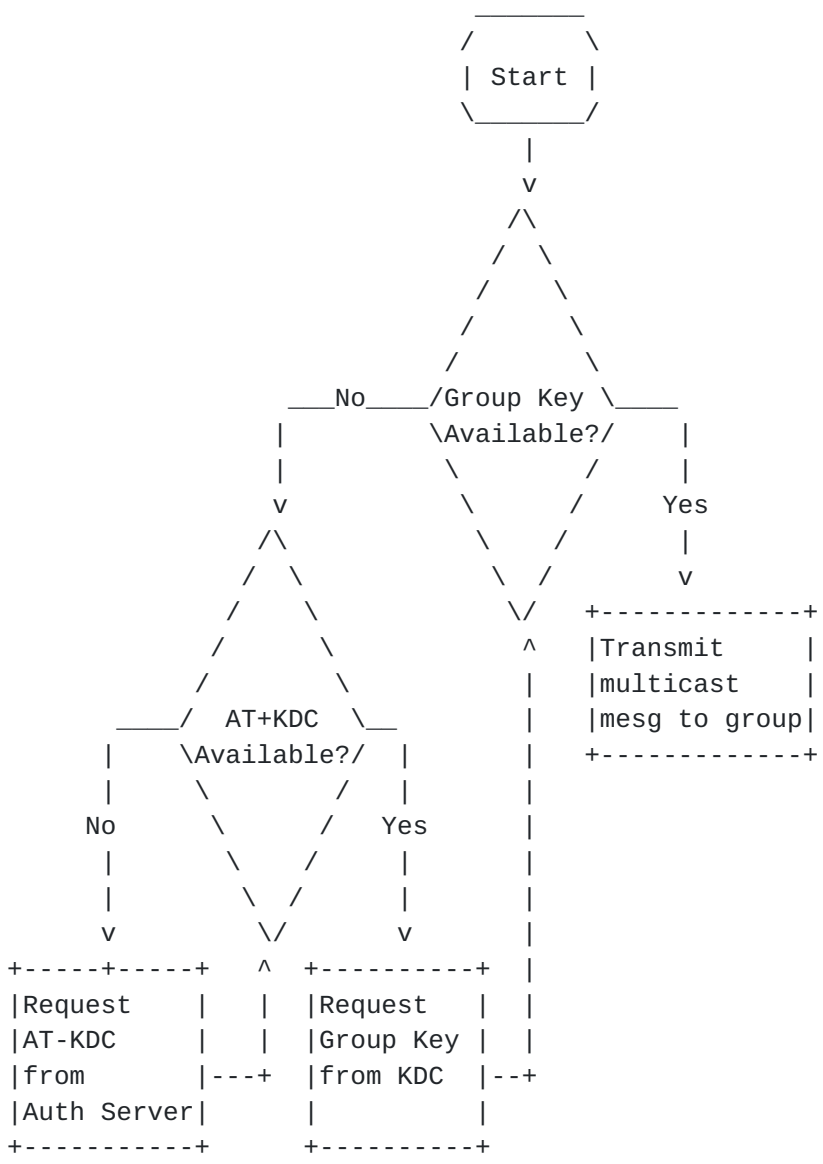


Figure 3: Steps to Transmit Multicast Message (w/o Failure Cases).

Note that the sender does not have to wait until it has to transmit a message in order to request a group key; the sender is likely to be pre-configured with information about which lighting application group it belongs to and can therefore pre-fetch the required information.

Group keys have a lifetime, which is configuration dependent, but mechanisms need to be provided to update the group keys either via the sender asking for a group key renewal or via the KDC pushing new keys to senders and receivers. The lifetime can be based on time or on the number of transmitted messages.

5. Access Tokens

Section 4 describes the architecture and makes use of access tokens, which is a generic concept to pass capabilities between entities in a distributed system. To improve interoperability a token format needs to be standardized and this section outlines the use of an existing format based on the JSON Web Token (JWT). These access tokens come in two flavors, namely as bearer tokens but also as proof-of-possession tokens. The main difference between the two is that bearer tokens are not associated with a key while proof-of-possession (PoP) tokens are. For a more detailed description of the security benefits of PoP tokens and the differences to bearer tokens please consult [I-D.ietf-oauth-pop-architecture]. In Section 1 we assume that the AT-KDC is a bearer token and the AT-R is a PoP token.

In this section we provide more details of the access token concept. The capabilities, called claims in the JWT jargon, are included inside the token and, in this example, state that the granted access level is 2 and access to the application group 2 is allowed by the sender. Most of the description focuses on the use of PoP tokens since they are more complex than bearer tokens. For the use with multicast security we envision the PoP token to contain a symmetric key encapsulated inside the JSON Web Key (JWK).

While JSON is a compact encoding format, standardization work is ongoing to define an even more efficient format for conveying the same information using a binary format (using CBOR as defined in RFC 7049 [RFC7049]). The corresponding security protection is currently being defined in the IETF COSE working group [COSE].

The content of a JSON Web Token (JWT) is protected using a JSON Web Signature (JWS). The JWS applies a message authentication code (MAC) to protect against forgery. The actual MAC computation (and the result) is omitted in this example. (Note that deployments may choose to use a digital signature to protect the JWT. While the JWT access token offers this flexiblity we assume symmetric keys in our example.

The JWT body is protected using the JWS. The JWS wraps around the body with a header and the actual message authentication code, which is not shown below.

JWS Header:

```
{"alg":"HS256",
 "kid":"123"
}
```

Legend for JWS Header:
- alg: Algorithm parameter indicating the type of cryptographic algorithm used to protect the structure. In this case HMAC-SHA 256 is used.
- kid: Key Identifier used to select the appropriate key.

The JWT body contains various claims and we included several of them in our example below. The most interesting one is the (not-yet-defined) scope (scp) claim offering information about the capabilities. In this example the scope ('scp') claim carries permissions described in Section 6. The included capabilities will depend on the type of token, namely AT-R vs. AT-KDC, and of course on the specific deployment environment.

JWT Body:

```
{
   "iss": "coaps://as.example.com",
   "exp": "1361398824",
   "scp": ["l2,g0,IP_M_R1", "l2,g1,IP_M_R1","l2,g2,IP_M_R1"],
   "cnf":{
     "jwe":
      "eyJhbGciOiJSU0ExXzUiLCJlbmMiOiJBMTI4Q0JDLUhTMjU2IiwiY3R5Ijoi
       andrK2pzb24ifQ. ... (remainder of JWE omitted for brevity)"
     }
  }
```

Legend for JWT Body:
- iss: Issuer (which is the entity creating the access token). The content does not need to be a URI but can also be a string identifying the entity issuing the token.
- exp: Expiry date of the access token. Claim can be omitted if tokens do not expire (for example, in a small enterprise environment).
- scp: Scope denoting the capabilities of the token
- cnf: Key confirmation element containing an encrypted JSON Web Key. The encryption being applied uses JSON Web Encryption (JWE).

The content from here onward is only relevant to the AT-R, which is assumed to be a PoP token. Note that a bearer token would not contain the key confirmation claim shown in the JWT body since there is no embedded key.

Figure 4 shows the structure of the PoP token graphically:

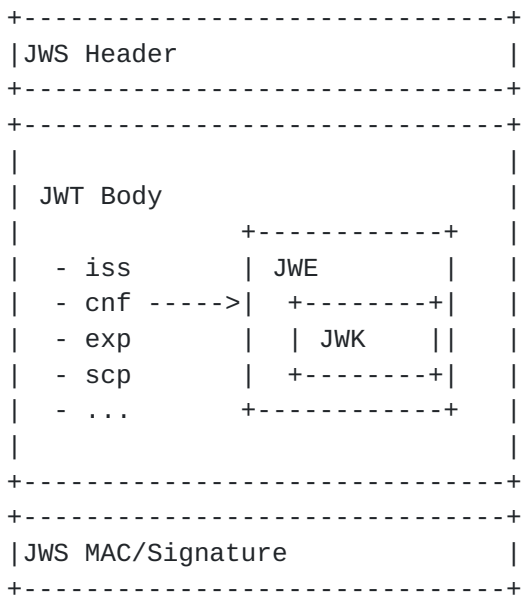


Figure 4: PoP Token Structure.

The JSON Web Encryption (JWE) contains a header followed by the content that is encrypted. Details about the JWE usage relevant for this example can be found in Appendix A.3 of RFC 7516 [RFC7516].

JWE Header:

```
{"alg":"A128KW",
 "enc":
 "A128CBC-HS256"
}
```

Legend for JWE Header:
- alg: Algorithm parameter indicating the AES 128 Key Wrap algorithm being used for encrypting the included key and the AES_128_CBC_HMAC_SHA_256 algorithm is used for authenticated encryption.

JWK to be encrypted by JWE:

```
 {"kty":"oct",
  "k":"GawgguFyGrWKav7AX4VKUg"
 }
```

Legend for the JWK to be encrypted:
- kty: Key type identifies the cryptographic algorithm family used with the key. In this example the JWK contains a symmetric key denoted as "oct" for octet sequence.
- k: Key parameter containing the actual key.

6. Lighting Application Example

In this Section, we look at a typical lighting application in which presence sensor(s) are used to actuate a group of light points via a control function based on a pre-specified set of rules. The CoAP/LWMM2M/IPSO protocol stack can be used as a foundation for the design of a lighting system. The architecture identifies three functions present in a lighting system:

- Sensor functions which detect a (physical) phenomenon like a light sensor, a presence detector or a push button.

- Actuator functions which cause action or change like setting a driver value of a light source.

- Control functions which link the inputs (from sensor functions) to outputs (from actuator functions) and enforce specific behaviour.

In typical applications, a sensor output might be used by multiple control functions and a single control function might control many actuators and a single actuator may be controlled by multiple control functions. Moreover, different functions (e.g. control and actuator) might be collocated on a single device. In the example below, we show one method that may be used to implement the above architecture using the LWM2M object model. We begin with the case of unicast communication (because the LWM2M model does not directly support multicast communication). We then explain a possible way to extend to the multicast situation.

6.1. Unicast Messages using the LWM2M Object Model

The unicast scenario considers a deployment with a single (physical) presence sensor, a single (physical) luminaire and the desired control functionality is the following: dim the luminaire to 90% when presence is detected in the room and dim the luminaire to 10% when

there is no presence. In this situation, the sensor functionality is implemented on the presence sensor, the actuator functionality is implemented on the luminaire and the control functionality could be implemented on the presence sensor or the luminaire or on an independent physical control device.

Using the LWM2M object model,

- the presence sensor function is implemented using the IPSO Presence object with Object ID 3302 [1].

- the actuator control function is implemented using the IPSO Light control object with Object ID 3311 [1].

- the control function is implemented by a LWM2M server to which the two LWM2M clients on the luminaire and presence sensor register.

The IPSO Light Control Object supports the "Dimmer" resource (Res ID 5851) which may be written to in order to change the light intensity output. The IPSO Presence Object supports the "Digital Input State" resource (Res ID 5500) which is a boolean readable resource that reflects the current state of the presence sensor. A method to implement the control functionality is the following:

1. The luminaire and the presence sensor register their objects with the control LWM2M server. This registration step happens during commissioning phase, when the device reboots or whenever IP addresses change.

2. The control LWM2M server observes the "Digital Input State" of the presence sensor.

3. When the presence sensor state changes, the sensor notifies the control LWM2M server.

4. The control LWM2M server writes the correct output intensity to the "Dimmer" resource to change the luminaire light output.

This sequence of messages is shown in Figure 5. Here, [IP_C], [IP_L] and [IP_S] are the unicast IP addresses of the devices that implement the control function, light control object and sensor object, respectively.

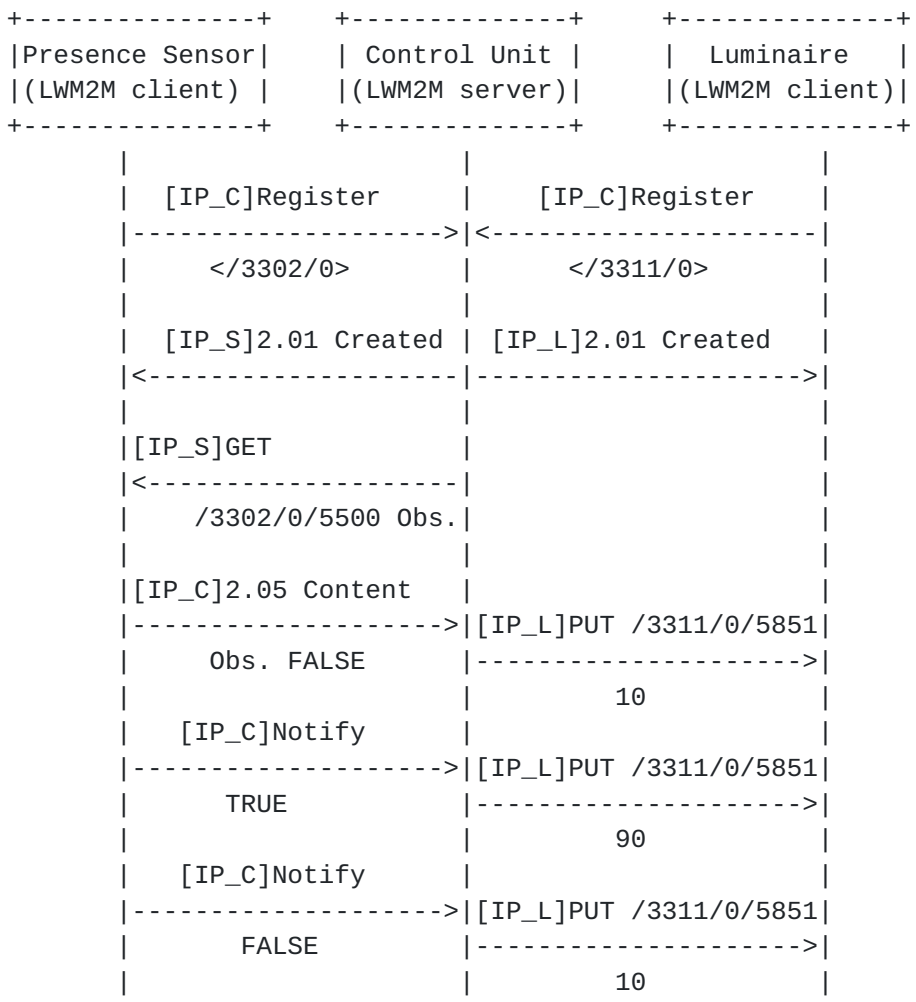


Figure 5: Unicast messaging between control unit and luminaire.

6.2. Multicast Communication using the LWM2M Object Model

We now see how the above unicast model may be extended to the group communication case and explain the security implications of the group communication case. Let us now look at a typical lighting application that requires group communication: 1) A set of rooms are attached to a single corridor; 2) Each room consists of 8 luminaires with 4 luminaires close to a window and four luminaires close to a wall; 3) Every room has a presence sensor and the corridor also has a presence sensor. 4) Every room has an individual control function that maybe implemented on the room presence sensor device, one of the luminaries or an independent control device.

The control functionality we wish to implement is the following:

- If presence is detected in the room, then dim-up the wall luminaires to 90% and window luminaires to 50%.

- If no presence is detected in the room or corridor, then turn off all luminaires.

- If no presence is detected in the room but presence is detected in the corridor, then turn all the luminaires to 10% (for all rooms attached to corridor).

For multicast communication, we can not use the LWM2M model directly. However, we can make the following assumptions:

- All luminaires are CoAP servers whose resource tree supports the IPSO Light Control object.

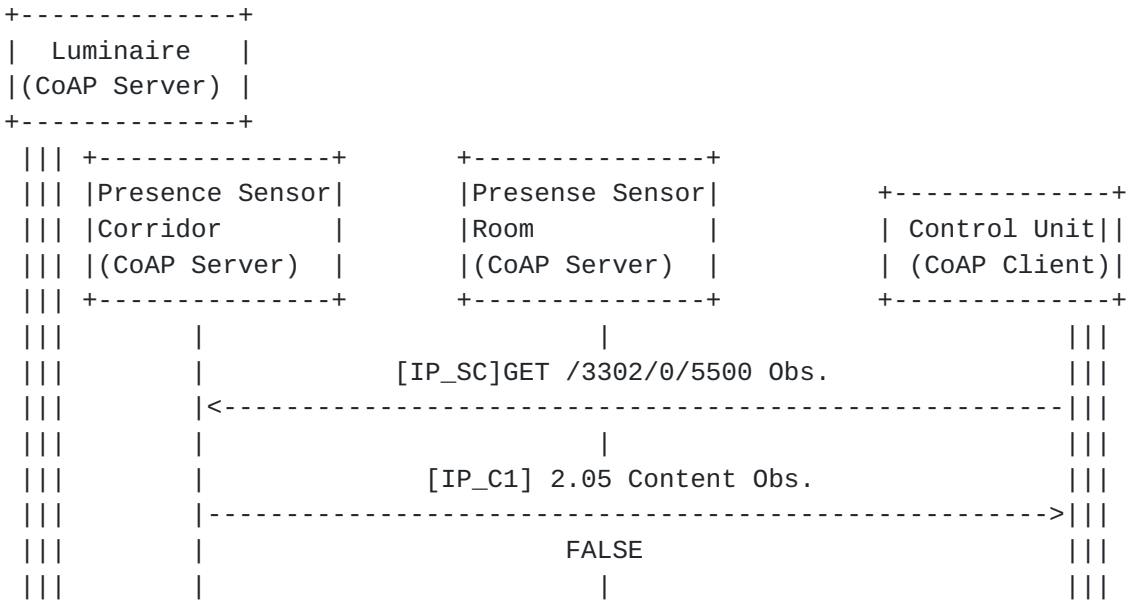
- All presence sensors are CoAP servers whose resource tree supports the IPSO Presence Object.

- The control function is implemented using a CoAP client.

- All devices in the nth room subscribe to the multicast address [IP_M_Rn] and the device that implements the control function in this room has unicast address [IP_Cn].

- Every room has three application groups and only one security group. The application groups are: "room group" (GId = 0), "window group" (GId = 1), "wall group" (GId = 2). The security group is defined by the symmetric key that is shared with the members.

- The GId is carried as a CoAP header (query?) option.



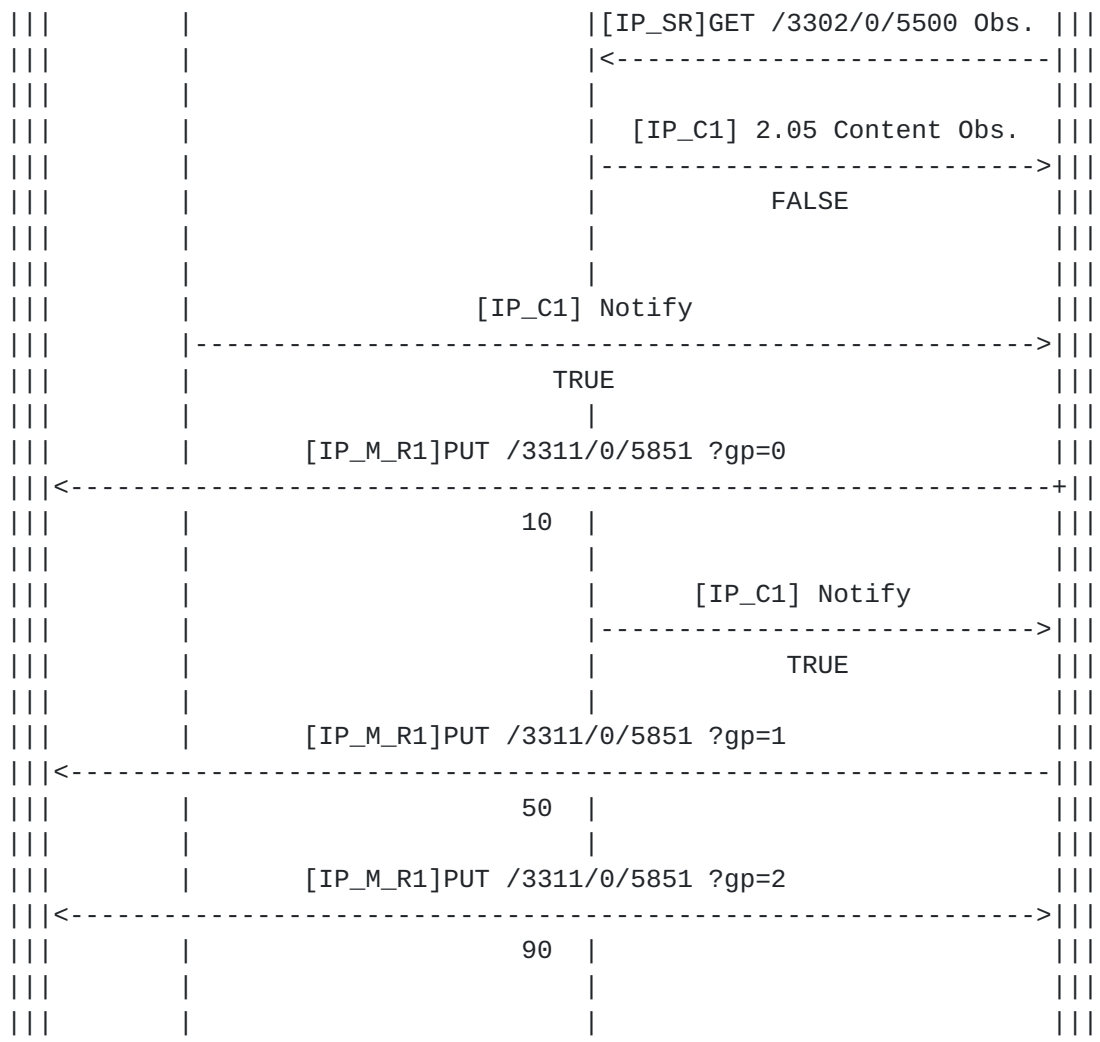


Figure 6: Multicast messaging between control unit and luminaires.

Figure 6 shows a typical sequence of messages that occur. For, simplicity, we only show the messages exchanged with the control function in room 1 and luminaires in room 1 though the same exchange of messages applies to every room. Initially, the control function in every room observes resource 5500 on the presence sensor in the corridor and also the presence sensor in it's own room. When the presence state changes in the corridor, the corridor notifies the control function in every room using a sequence of unicast notification messages. Once a controller in the room receives this notification, it sends out a multicast message to all luminaires in the group (GId = 0). If presence is then detected in the room, then the room controller is notified and the room controller sends a multicast message to the window group (GId = 1) to dim-up to 50% and wall group (GID = 2) to dim-up to 90%. Note the separation of the two

types of sensors in this problem: The presence sensor in a room is a part of the room group and therefore will receive the room group key which allows it to directly talk to the luminaires in the room to which the sensor belongs. However, the corridor presence sensor is not a part of the room group and does not receive the room group key. The corridor presence sensor must only authorized to communicate with the room control function which then controls the luminaires.

In this Example, there are three application groups per room and one multicast group per room. There are two types of security groups: one security group per room and a security group that has the corridor presence sensor and all the control units attached to the corridor. Therefore, the control unit in room 1 requires access tokens with the following scope, "l2,g0,IP_M_R1", "l2,g1,IP_M_R1", "l2,g2,IP_M_R1" for room control and also "l2,g0,IP_SC" for corridor presence sensor. The KDC generates 2 keys - KeyRoom1 and KeyCor that need to be delivered to the control unit in room 1: KeyRoom1 is used to communicate with the room luminaire group for all three application groups - g0, g1, g2 and KeyCor is used to communicate with the corridor presence sensor.

7. Security Considerations

7.1. Token Verification

Due to the low latency requirements, token verification needs to be done locally and cannot be outsourced to other parties. For this reason self-contained token must be used and the receivers are required to follow the steps outlined in Section 7.2 of RFC 7519 [RFC7519]. This includes the verification of the message authentication code protecting the contents of the token and the encryption envelope protecting the contained symmetric group key.

7.2. Token Revocation

Tokens have a specific lifetime. Setting the lifetime is a policy decision that involves making a trade-off decision. Allowing a longer lifetime increases the need to introduce a mechanism for token revocation (e.g., a real-time signal from the KDC/Authorization Server to the receivers to blacklist tokens) but lowers the communication overhead during normal operation since new tokens need to be obtained only from time to time. Real-time communication with the receivers to revoke tokens may not be possible in all cases either, particularly when off-line operation is demanded or in small networks where the AS or even the KDC is only present during commissioning time.

We therefore recommend to issue short-lived tokens for for dynamic scenarios like users accessing the lighting infrastructure of buildings using smartphones, tablets and alike to avoid potential security problems when tokens are leaked or where authorization rights are revoked. For senders that are statically mounted (like traditional light switches) we recommend a longer lifetime since re-configurations and token leakage is less likely to happen frequently.

To limit the authorization rights tokens should contain an audience restriction, scoping their use to the intended receivers and to their access level.

7.3. Time

Senders and receivers are not assumed to be equipped with real-time clocks but these devices are still assumed to interact with a time server. The lack of accurate clocks is likely to lead to clock drifts and limited ability to check for replays. For those cases where no time server is available, such as in small network installations, token verification cannot check for expired tokens and hence it might be necessary to fall-back to tokens that do not expire.

8. Operational Considerations

8.1. Persistence of State Information

Devices in the lighting system can often be powered down intentionally or unintentionally. Therefore the devices may need to store the authorization tokens and cryptographic keys (along with replay context) in persistence storage like flash. This is especially required if the authorization server is no more online since it was removed after the commissioning phase. However the decision on the data to be persistently stored is a trade-off between how soon the devices can be back online to normal operational mode and the memory wear caused due to limited program-erase cycles of flash over 15-20 years life-time of the device.

The different data that may need to be stored are access tokens AT-KDC, AT-R and last seen replay counter.

8.2. Provisioning in Small Networks

In small networks the authorization server and the KDC may be available only temporarily during the commissioning process and are not available afterwards.

9. Acknowledgements

The author would like to thank Walter Werner and Esko Dijk for his help with this document.

10. References

10.1. Normative References

[I-D.gerdes-ace-actors]
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", draft-gerdes-ace-actors-05 (work in progress), April 2015.

[IPSO] IPSO Alliance, "IPSO Smart Object Guidelines - Starter Pack 1.0", 2015.

[LWM2M] Open Mobile Alliance, "Lightweight Machine-to-Machine, Technical Specification, Candidate Version 1.0", December 2013.

[RFC7516] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, May 2015.

10.2. Informative References

[COSE] IETF, "CBOR Object Signing and Encryption (cose) Working Group", https://datatracker.ietf.org/wg/cose/charter/, 2015.

[I-D.ietf-oauth-pop-architecture]
Hunt, P., Richer, J., Mills, W., Mishra, P., and H. Tschofenig, "OAuth 2.0 Proof-of-Possession (PoP) Security Architecture", draft-ietf-oauth-pop-architecture-01 (work in progress), March 2015.

[I-D.kumar-dice-multicast-security]
Kumar, S. and R. Struik, "Transport-layer Multicast Security for Low-Power and Lossy Networks (LLNs)", draft-kumar-dice-multicast-security-00 (work in progress), March 2015.

[I-D.selander-ace-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "June 29, 2015", draft-selander-ace-object-security-02 (work in progress), June 2015.

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012.

[RFC6750] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", RFC 6750, October 2012.

[RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", RFC 7049, October 2013.

[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", RFC 7519, May 2015.

Authors' Addresses

Abhinav Somaraju
Tridonic GmbH & Co KG
Farbergasse 15
Dornbirn 6850
Austria

Email: abhinav.somaraju@tridonic.com

Sandeep S. Kumar
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: ietf.author@sandeep-kumar.org

Hannes Tschofenig
ARM Ltd.
110 Fulbourn Rd
Cambridge CB1 9NJ
Great Britain

Email: Hannes.tschofenig@gmx.net
URI: http://www.tschofenig.priv.at