

ace  
Internet-Draft  
Intended status: Standards Track  
Expires: July 18, 2016

A. Somaraju  
Tridonic GmbH & Co KG  
S. Kumar  
Philips Research  
H. Tschofenig  
ARM Ltd.  
W. Werner  
Werner Management Services e.U.  
January 15, 2016

**Security for Low-Latency Group Communication**  
**draft-somaraju-ace-multicast-01.txt**

**Abstract**

Some Internet of Things application domains, such as lighting, have strict requirements on latency for group communication. From a user experience point of view latency less than 200 ms is necessary from an action triggered by a user to the visible effects. This draft describes procedures for authorization, key management, and securing group messages within a low latency application domain with a special emphasis on lighting systems. We specify the usage of object security at the application layer for group communication and assume that CoAP is used as the application layer protocol.

**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 July 18, 2016.

**Copyright Notice**

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Terminology . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Architecture . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	Assumptions . . . . .	<a href="#">7</a>
<a href="#">3.2.</a>	AT-KDC Access Tokens . . . . .	<a href="#">8</a>
<a href="#">3.3.</a>	AT-R Access Tokens . . . . .	<a href="#">9</a>
<a href="#">3.4.</a>	Multicast Message Content . . . . .	<a href="#">10</a>
<a href="#">3.5.</a>	Receiver Algorithm . . . . .	<a href="#">11</a>
<a href="#">3.6.</a>	Sender Algorithm . . . . .	<a href="#">12</a>
<a href="#">4.</a>	Security Considerations . . . . .	<a href="#">14</a>
<a href="#">4.1.</a>	Token Verification . . . . .	<a href="#">14</a>
<a href="#">4.2.</a>	Token Revocation . . . . .	<a href="#">14</a>
<a href="#">4.3.</a>	Time . . . . .	<a href="#">14</a>
<a href="#">5.</a>	Operational Considerations . . . . .	<a href="#">15</a>
<a href="#">5.1.</a>	Persistence of State Information . . . . .	<a href="#">15</a>
<a href="#">5.2.</a>	Provisioning in Small Networks . . . . .	<a href="#">15</a>
<a href="#">5.3.</a>	Client IDs . . . . .	<a href="#">15</a>
<a href="#">5.4.</a>	Application Groups vs. Security Groups . . . . .	<a href="#">15</a>
<a href="#">5.5.</a>	Lost/Stolen Device . . . . .	<a href="#">16</a>
<a href="#">6.</a>	Acknowledgements . . . . .	<a href="#">16</a>
<a href="#">7.</a>	IANA Considerations . . . . .	<a href="#">16</a>
<a href="#">8.</a>	References . . . . .	<a href="#">17</a>
<a href="#">8.1.</a>	Normative References . . . . .	<a href="#">17</a>
<a href="#">8.2.</a>	Informative References . . . . .	<a href="#">17</a>
<a href="#">Appendix A.</a>	Access Levels . . . . .	<a href="#">18</a>
	Authors' Addresses . . . . .	<a href="#">19</a>

## **[1.](#) Introduction**

There are low latency group communication use cases that require securing communication between a sender, or a group of senders, and a group of receivers. In the lighting use case, a set of lighting nodes (e.g., luminaires, wall-switches, sensors) are grouped together into a single "Application Group" and the following three requirements need to be addressed:



1. Only authorized members of the application group must be able to 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. Message communication and processing must happen with a low latency and in synchronous manner.

This document discusses a group communication security solution that satisfies these three requirements. This write-up focuses on the lighting use case but the content is equally applicable to other low-latency application domains, such as blinds control.

## 2. Terminology

This document uses the following terms from [[I-D.ietf-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.

When nodes are combined into groups there are different layers of those groups with unique characteristics. For clarity we introduce terminology for three different groups:

### Application Group:

An application group consists of the set of all nodes that have been configured to respond to a single application layer request. For example, 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 the set of all nodes that have been provisioned with the same keying material. All the nodes within a security group share a security association or a sequence of security associations wherein a single association specifies the



keying material, algorithm-specific information, lifetime and a key ID.

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. Note that every application group belongs to a unique security group. However, the converse is not always true. This implies that the application group ID maybe used to determine the associated security group but not vice versa.

The fact that security groups may not coincide with application groups implies that

- (1) an application must be able to specify which resources on a resource server are accessible by a client that has access to the group key, and

- (2) a method is required to associate the group key to the application group(s) for which the group key may be used.

In this document we provide fields that may be used to specify the "scope of the key" and "application groups for which the key may be used". A commissioner has a lot of flexibility to assign nodes to multicast groups and to security groups while the application groups will be determined by the semantics of the application itself. The exact partitioning of the nodes into security and multicast groups is therefore deployment specific.

### **3. Architecture**

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

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 renewed 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 luminaire is associated with different groups, and may have meta-data about its location in a building.

Note that we assume that nodes are pre-configured with device credentials (e.g., a certificate and the corresponding private key) during manufacturing or during an initial provisioning phase. 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 group messages between senders and receivers are indicated using lines with star/asterisk characters. The security of the group messages is accomplished at the application level using OSCOAP - Object Security of CoAP (see [[I-D.selander-ace-object-security](#)]).

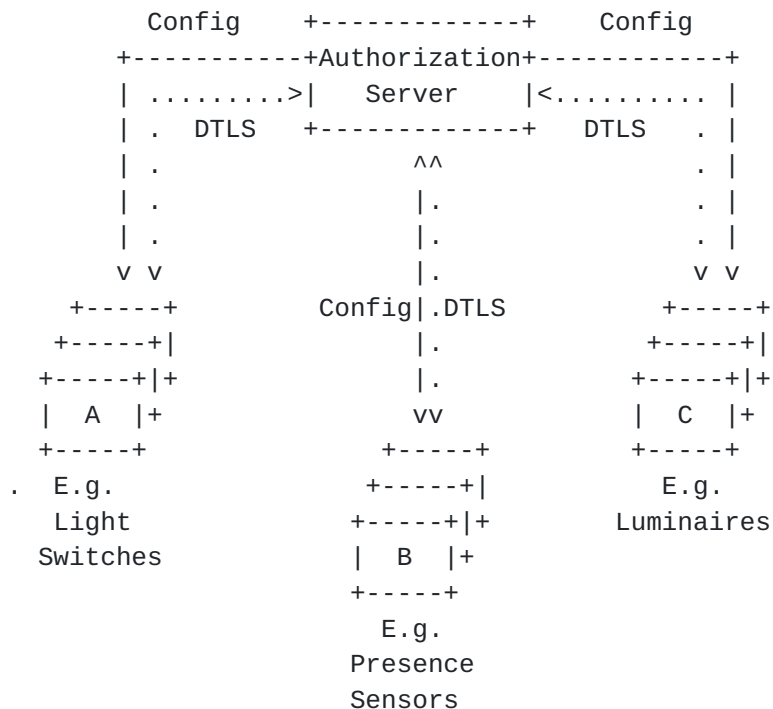
Figure 1 illustrates the information flow between an authorization server and the nodes participating in the lighting network, which includes all nodes that exchange lighting application messages. This step is typically executed during the commissioning phase for nodes that are fixed-mounted in buildings. 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. Other use cases, such as 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 [[RFC6749](#)]. The AT-KDC in this architecture may be a bearer token or a proof-of-possession (PoP) token. The bearer token concept is described in [[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 the 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 group messages. The symmetric key obtained from the KDC is used to secure the groups messages. The AT-R may be attached to the initial request.

Receivers need to perform two steps, namely to obtain the necessary group key to verify the incoming messages and to determine what



resource the requestor is authorized to access. Both pieces of information can be found in the AT-R access token.

Group messages need to be protected such that replay and modification can be detected. The integrity of the message is accomplished using a keyed message digest in combination with the group key. The use of symmetric keys is envisioned in this specification due to latency requirements. For unicast messaging between the group members and the AS or KDC, we assume the use of DTLS for transport security. However, the use of TLS, and application layer security is possible but is outside the scope of this document.

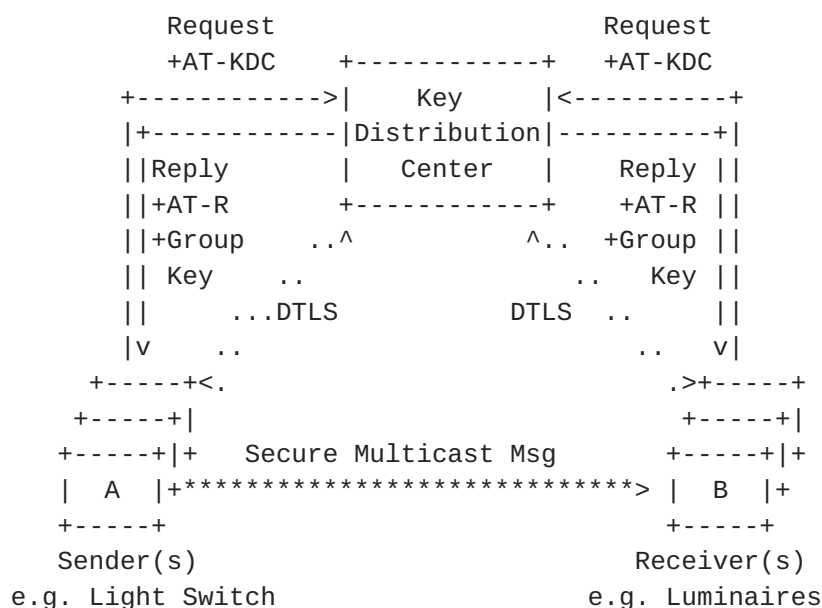


Figure 2: Architecture: Group Key Distribution Phase.

### 3.1. Assumptions

1. The AT-KDC is a manifestation of the authorization granted to a specific client (or user running a client). The AT-KDC is longer-lived and can be used to request multiple AT-Rs.
2. Each AT-R is valid for use with one or multiple application groups.
3. The AS and the KDC logical roles may reside in different physical entities.



4. The AT-KDC as well as the AT-R may be self-contained tokens or references. References are more efficient from a bandwidth point of view but require an additional lookup.
5. The AT-KDC token is opaque to the client. Data that is meant for processing by the client has to be conveyed to the client separately. The AT-R token on the other hand is meant for consumption by the client.
6. The client requests AT-Rs for different application groups by including additional information in the request to the KDC for what application groups the AT-R(s) have to be requested. The KDC may return multiple AT-Rs in a single response (for performance reasons).
7. The AT-KDC and the AT-R are encoded as CBOR Web Tokens [[I-D.wahlstroem-ace-cbor-web-token](#)] and protected using COSE [[I-D.ietf-cose-msg](#)].

### **3.2. AT-KDC Access Tokens**

The AT-KDC contains

1. Issuer: Entity creating the access token. This information needs to be cryptographically bound to the digital signature/keyed message digest protecting the content of the token, as provided by the CBOR Web Token (CWT).
2. Expiry date: Information can be omitted if tokens do not expire (for example, in a small enterprise environment).
3. Scope: Permissions of the entity holding the token. This includes information about the resources that may be accessed with the token (e.g., access level) and application layer group IDs for the groups for which the tokens may be used.
4. Recipient/Audience: Indication to whom the AT-KDC was issued to. In this case, it is the KDC.
5. Client ID: Information about the client that was authenticated by the authorization server.
6. Issued at: Indicates date and time when the AT-KDC was created by the authorization server.



### **3.3. AT-R Access Tokens**

Clients send the AT-KDC to the KDC in order to receive an AT-R.

The KDC MUST maintain a table consisting of scope values, which includes the application group id. These entries point to a sequence of security associations. A security association specifies the key material, algorithm-specific information, lifetime and a key ID and the key ID may be used to identify this security association.

The AS/KDC must guarantee the uniqueness of the client ids for its nodes. This may be accomplished by the AS/KDC assigning values to the nodes or by using information that is already unique per device (such as an EUI-64).

The KDC furthermore needs to be configured with information about the authorization servers it trusts. This may include a provisioned trust anchor store, or shared credentials (similar to a white list).

The KDC MUST generate new group keys after the validity period of the current group key expires.

The AT-R contains

1. Issuer: Entity creating the access token. This information needs to be cryptographically bound to the digital signature/keyed message digest protecting the content of the token, as provided by the CBOR Web Token (CWT).
2. Expiry date: Information can be omitted if tokens do not expire (for example, in a small enterprise environment).
3. Scope: Permissions of the entity holding the token. This includes information about the resources that may be accessed with the token (e.g., access level) and application layer group IDs for the groups for which the tokens may be used.
4. Security Group Key: Key to use for the group communication.
5. Algorithm: Used for secure group communication.
6. KID: Sequentially increasing ID of the key for the security group (the devices may store an older key to help with key rolling.)
7. Issued at: Indicates date and time when the AT-R was created by the KDC.





### **3.4. Multicast Message Content**

The following information is needed for the cryptographic algorithm, which is assumed to be in the COSE header:

1. Nonce value consisting of
  - \* Client ID (unencrypted, integrity protected): Every sender managed by a key distribution center MUST have a unique client ID.
  - \* Sequence Number (unencrypted, integrity protected): Used for replay protection.
  - \* An implicit IV that is either derived from the keys at the end-points or fixed to a certain value by standard (not sent in the message)
2. MAC (not integrity protected): For integrity protection.

The following information is additionally required to process the secure message:

1. Destination IP address and port (not encrypted, integrity protected): Integrity protection of the IP address and port ensures that the message content cannot be replayed with a different destination address or on a different port.
2. CoAP Path (encrypted, integrity protected): Uniquely identifies the target resource of a CoAP request.
3. Application Group id in CoAP header (unencrypted, integrity protected): Is used to identify a sequence of security associations to use to decrypt the message. The CoAP header option is TBD.
4. Key ID (unencrypted, integrity protected): Is used to select the current security association from the sequence of security associations identified by the application group id.
5. CoAP Header Options other than application group id (encrypted - if desired, integrity protected)
6. CoAP Payload (encrypted, integrity protected).



### 3.5. Receiver Algorithm

All receiving devices MUST maintain a table consisting of mappings of application group id, to a sequence of security associations.

When a node receives an incoming multicast message it looks up the application group id and the key id (which are both found in the CoAP header) to determine the correct security association.

The key id is used for situations where the group key is updated by the KDC (for example in situations where a device in a group is lost or stolen).

To check for replay attacks the receiver has to consult the state stored with the security association to obtain the current sequence number and to compare it against the sequence number found in the request payload for that sender based on the Sender ID. The receiver needs to store the latest correctly verified nonce values to detect replay attacks

The receiver MUST silently discard an incoming message in the following cases:

- o Application Group ID lookup does not return any security association.
- o Key ID lookup among the previously retrieved sequence of security associations does not identify a unique security association.
- o Integrity check fails.
- o Decryption fails.
- o Replay protection check failed. The (client ID || sequence number), which are both part of the nonce, have already been received in an earlier message.

Once the cryptographic processing of the message is completed, the receiver must check whether the sender is authorized to access the protected resource, indicated by the CoAP request URI at the right level. For this purpose the receiver consults the locally stored authorization database that was populated with the information obtained via the AT-R token and the static authorization levels described in [Appendix A](#).

Once all verification steps have been successful the receiver executes the CoAP request and returns an appropriate response. Since the response message will also be secured the message protection



processing described in [Section 3.6](#) must be executed. Additionally, the nonce value corresponding to the security association MUST be updated to the nonce value in the message.

### **3.6. Sender Algorithm**

Figure 3 describes the algorithm for obtaining the necessary credentials to transmit a secure group message. When the sender wants to send a message to the application group, it checks if it has the respective group key. If no group key is available then it determines whether it has an access token for use with the KDC (i.e., AT-KDC). If no AT-KDC is found in the cache then it contacts the authorization server to obtain that AT-KDC. 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 secured message to the group immediately.



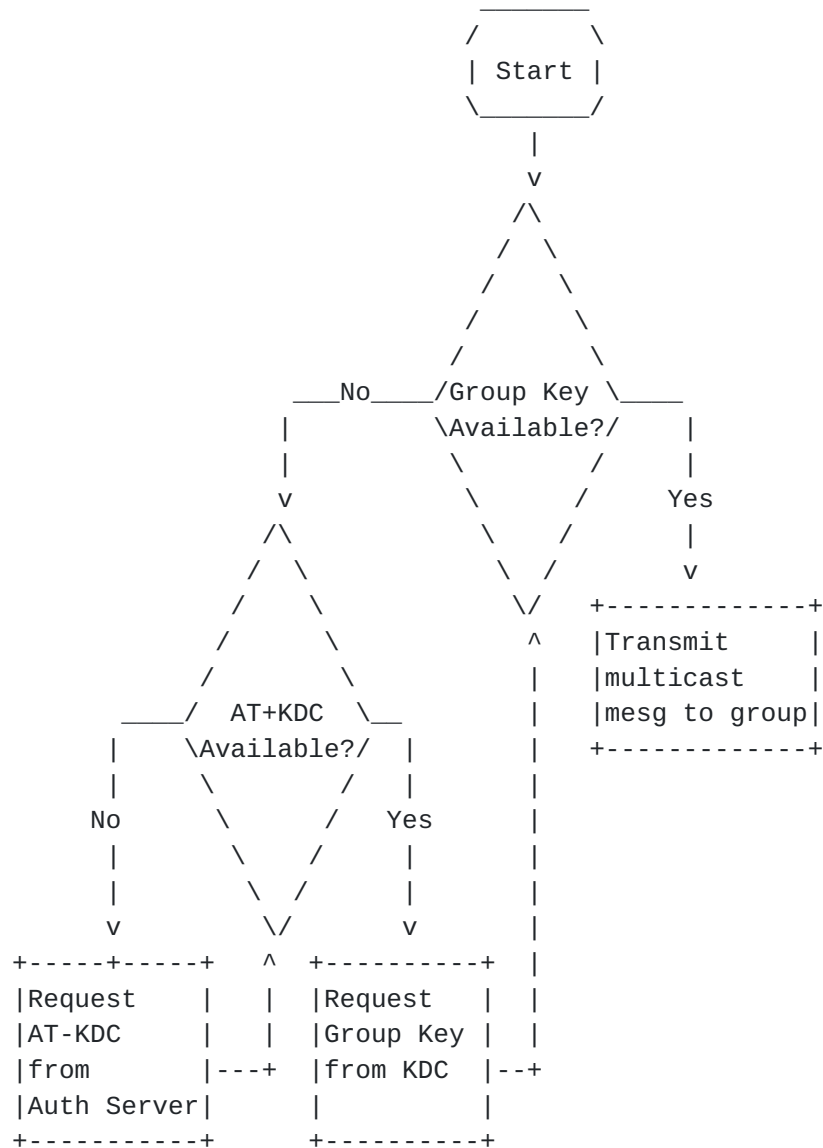


Figure 3: Steps to Transmit Multicast Message (w/o Failure Cases).

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 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.





## **4. Security Considerations**

### **4.1. 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 a 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.

### **4.2. 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 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.

### **4.3. 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.



## **5. Operational Considerations**

### **5.1. 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 persistent storage like flash. This is especially required if the authorization server is no more online because 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 the 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.

### **5.2. 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.

### **5.3. Client IDs**

A single device should not be managed by multiple KDCs. However, a group of devices in a domain (such as a lighting installation within an enterprise) should either be managed by a single KDC or, if there are multiple KDCs serving the devices in a given domain, these KDCs MUST exchange information so that the assigned client id and application group id values are unique within the devices in that domain. We assume that only devices within a given domain communicate with each other using group messages.

### **5.4. Application Groups vs. Security Groups**

Multiple application groups may use the same key for performance reasons, reducing the number of keys needed to be stored - leading to less RAM needed by each node. This is only a reasonable option if the attack surface is not increased. For example, a room A is configured to use three application groups to address a subset of the device. In addition to configuring all nodes in room A with these three application groups the nodes are configured with a special group that allows them to access all devices in room A, referred as the all-nodes-in-room-A group. In this case, having the nodes to use the same key for the all-nodes-in-room group and the three groups does not increase the attack surface since any node can already use



the all-nodes-in-room-A group to control other devices in that room. The three application groups in room A are a subset of the larger all-nodes-in-room-A group.

### **5.5. Lost/Stolen Device**

The following procedure **MUST** be implemented if a device is stolen or keys are lost.

1. The AS tells the KDC to invalidate the AT-KDC.
2. The KDC no longer returns a new group key if the invalidated AT-KDC is presented to it.
3. The KDC generates new keys for all security groups to which the compromised device belongs.

The KDC **SHOULD** inform all devices in the security group to update their group key. This requires the KDC to maintain a list of all devices that belong to the security group and to be able to contact them reliably.

## **6. Acknowledgements**

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

Parts of this document are a byproduct of the OpenAIS project, partially funded by the Horizon 2020 programme of the European Commission. It is provided "as is" and without any express or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the OpenAIS project or the European Commission.

## **7. IANA Considerations**

This document defines one CoAP Header Option Application Group ID that **MUST** be allocated in the Registry "CoAP Option Numbers" of [\[RFC6749\]](#). IANA is requested to allocation TBD option number to application group ID in this specification.



## 8. References

### 8.1. Normative References

- [I-D.ietf-ace-actors]  
Gerdes, S., Seitz, L., Selander, G., and C. Bormann, "An architecture for authorization in constrained environments", [draft-ietf-ace-actors-02](#) (work in progress), October 2015.
- [I-D.ietf-cose-msg]  
Schaad, J., "CBOR Encoded Message Syntax", [draft-ietf-cose-msg-09](#) (work in progress), December 2015.
- [I-D.wahlstroem-ace-cbor-web-token]  
Wahlstroem, E., Jones, M., and H. Tschofenig, "CBOR Web Token (CWT)", [draft-wahlstroem-ace-cbor-web-token-00](#) (work in progress), December 2015.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<http://www.rfc-editor.org/info/rfc7252>>.

### 8.2. Informative References

- [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-07](#) (work in progress), December 2015.
- [I-D.selander-ace-object-security]  
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security of CoAP (OSCOAP)", [draft-selander-ace-object-security-03](#) (work in progress), October 2015.
- [RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", [RFC 6749](#), DOI 10.17487/RFC6749, October 2012, <<http://www.rfc-editor.org/info/rfc6749>>.
- [RFC6750] Jones, M. and D. Hardt, "The OAuth 2.0 Authorization Framework: Bearer Token Usage", [RFC 6750](#), DOI 10.17487/RFC6750, October 2012, <<http://www.rfc-editor.org/info/rfc6750>>.





[RFC7519] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", [RFC 7519](https://tools.ietf.org/html/rfc7519), DOI 10.17487/RFC7519, May 2015, <<http://www.rfc-editor.org/info/rfc7519>>.

## **Appendix A. 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.



## Authors' Addresses

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

Email: [abhinav.somaraju@tridonic.com](mailto:abhinav.somaraju@tridonic.com)

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

Email: [ietf.author@sandeep-kumar.org](mailto:ietf.author@sandeep-kumar.org)

Hannes Tschofenig  
ARM Ltd.  
Hall in Tirol 6060  
Austria

Email: [Hannes.tschofenig@gmx.net](mailto:Hannes.tschofenig@gmx.net)  
URI: <http://www.tschofenig.priv.at>

Walter Werner  
Werner Management Services e.U.  
Josef-Anton-Herrburgerstr. 10  
Dornbirn 6850  
Austria

Email: [werner@werner-ms.at](mailto:werner@werner-ms.at)

