Survey for Secure IoT group communication

Jiye Park
Innovation
Osram GmbH
Munich, Germany
j.park@osram.com

Markus Jung
Innovation
Osram GmbH
Munich, Germany
m.jung@osram.com

Erwin P. Rathgeb
Computer networking technology group
University of Duisburg-Essen
Essen, Germany
erwin.rathgeb@uni-due.de

***Abstract*—The Internet of Things (IoT) paradigm is progressing fast toward Internet adoption and reaching out to various industrial domains such as smart lighting. New IoT use cases leveraging multicast group communication increase the demand for security to protect a number of devices. Providing dedicated multicast security for constrained IoT environment is the key to make IoT service successful. In this paper, we survey the state of the art of secure IoT group communication among devices. We focus on the context of commercial smart lighting, as it has as several use cases that rely on secure group communication. The paper summarizes use cases and security requirements for the multicast group communication, state of the art approaches focusing on standardization activities. Furthermore, the paper shows the evaluation result of the new application layer security protocol, OSCORE. We concludes with a research outlook on open problems.**

***Index Terms*—Internet of Things, Internet of Lights, security, CoAP, group communication, multicast security.**

I. Introduction

Several IoT use cases rely on or benefit from a group communication pattern. The basic idea is to increase the communication efficiency by avoiding a number of similar unicast messages to a group of receivers. This is the case for **discovery**, **firmware/software updates**, **update on group of similar devices**. Considering the constraint nature of IoT devices and networks, the UDP based protocol CoAP [1] opens the possibilities of multicast group communication within low-power wireless meshed networks based on IEEE 802.15.4.

With multicast group communication the efficiency can be improved. This results in considerably easier configuration and management of multiple devices at once. In particular, when group communication is used to support latency sensitive applications as found within lighting. The required network bandwidth and network propagation latency can be significantly reduced.

On the other hand, the same security requirements typically fulfilled in the presence of unicast communication are expected to be effectively fulfilled also in the presence of group communication. These include, among others: availability, authenticity, message integrity, confidentiality and freshness.

Furthermore, enforcing security in a group communication context results in additional non-trivial management operations, in particular related to distribution, revocation and renewal of security keying material in the group.

The remainder of the paper is organized as follow. In Section II we show the use of group communication in commercial smart lighting and security requirements. In Section III we discuss IoT secure group communication approaches based on Internet Engineering Task Force (IETF) standardization works. Section IV we show evaluation result of Object Security for Constrained RESTful Environment (OSCORE) which is the most feasible security protocol for IoT group communication. In Section V, we discuss about research challenges. Finally, we conclude our survey in Section VI.

II. Commercial smart lighting

The Internet of Light (IoL) for commercial installation requires a more capable system architecture compared to the smart home domain. In commercial environments especially the commissioning phase and the secure operation are more challenging. In this section, we discuss the use cases for group communication in these environments and their respective security requirements.

A. Use cases

The group communication can be used for the operational functions as below.

- Discovery: To discover devices deployed in a network, a client sends a query to a resource directory (RD). The IP address of a RD can be pre-configured at the manufacturing phase, but in the case when the client does not have such an IP address, the client can discover the RD by sending multicast message with URI [coap://[MCD1]/.well-known/core?rt=core.rd*] and getting a response to the request from RD [2].
- Device management: Improving the efficiency of applications relying on resource-constrained devices is the key point to provide successful IoT services. When many devices are operated in a common application context, it can be convenient to organize such devices as a group for device management. In particular, a group can be organized and enforced according to the common logical functionality or the physical location of its members.
- Firmware update/software update: Firmware and software updates are essential for a secure operation of an commercial smart lighting environment. When a large number of devices are deployed, updating firmware one by one is not only taking a lot of time but also causing traffic jam in the constrained network. In a non-stable network environment, packets get lost easily. Then the sender

978-1-5386-9151-9/19/$31.00 ©2019 IEEE

should take care of all the devices to make sure all the packets are delivered correctly. It also incurs a big burden on the server side pushing the data when there are big number of devices.

- Operation: In the scenario where a lot of luminaries are deployed for a same functionality, the luminaries are expected to operate at the same time to be on/off without time gap between other luminaries. Multicast communication enables the devices act as a group without propagation delay in a latency sensitive use case.

B. *Security requirement*

In the group communication architecture one-to-many and many-to-many communication relationships between senders (mutlicasters) and receivers (listeners) can be present. Security for multicast group communication should be considered even more carefully than unicast communication since it can affect to more devices at once. Furthermore, different key management mechanisms from the unicast case should be provided as group members can leave or join a group dynamically.

- Group key management
  - To have confidentiality of exchanged messages, the devices should be able to share a group key in a secure way.
  - When a device is joining or leaving a group, the group key that is shared and used for previous communication should be updated accordingly to provide forward secrecy and backward secrecy.
- Authentication and Authorization
  - Only an authenticated and authorized device should join to a group.
  - Only an authorized device should be able to access a certain resource. Allowing a not authenticated and authorized device to join a group might cause data leakage and congestion in a constrained network.
- Confidentiality and integrity
  - Data transmitted between the nodes should be encrypted. Since multicast IP addresses are public, any device having the multicast IP address can receive data sent from sender. If data is not encrypted, it enables eavesdropping attack.
  - Plain text can be easily modified by attacker. With data encryption, it can provide not only data confidentiality but also integrity as an attacker who does not have a key cannot modify the data.
- Non-repudiation and source authentication
  - In case a sender transmits a group message to receivers, the receivers should be able to verify that the message came from the authenticated sender of group.
  - When the receiver responds to a sender, the sender should be able to verify which receiver sends the response.
- Availability
  - Due to limited resources of an IoT devices they are more vulnerable to distributed denial of service (DDoS) attacks. When the authenticated sender is compromised, it can create network jamming by sending a lot of messages to the all listeners that are in the same network domain. Therefore, DDoS resistance scheme should be considered.
  - When the attacker try to re-distribute group key or make the group manager generates new key for a group, the attack should be detected.

III. State of the art

Security schemes for multicast communication were researched actively in the past. Most of proposals aimed Internet video transmissions, live multi-party conferencing and on-line video games as use cases [3] [4] [5], and those proposals are based on the environment that does not have network and resource limitations. Because of the different environment, these concepts cannot be applied to the IoT directly. For constrained environments, IoT alliances such as Open Connectivity Foundation (OCF), Zigbee, Thread, Fairhair and Open Mobile Alliance (OMA) are working on to meet IoT security requirement. However, those alliances do not propose new approaches but work on designing architecture for specific use cases using standard protocols. Thus the alliances are highly relying on IETF standards. In this section, we focuses on the IETF standardization works accordingly.

A. *IETF Standardization*

To provide interoperability to a heterogeneous IoT environment, Constrained RESTful Environments (CoRE) working group in IETF standardized an application layer protocol named CoAP. CoAP is running over UDP, and it has similarities to HTTP. To bind security to CoAP as HTTP over TLS, DTLS security protocol is defined as a mandatory security protocol that is allowing CoAPs scheme. For the CoAP security, depending on device resource capabilities and required security level, one of four security mode information is provided to a device during the provisioning time [1].

- *PresharedKey* mode: Literally, creating DTLS secure channel using pre-shared key that is already established on communication peers. When pre-shared key is used for group communication, it should be only used to authenticate group member
- *RawPublicKey* mode: Performing DTLS handshake using raw-public key with out of band mechanism that is defined in [6]
- *Certificate* mode: Create DTLS session using X.509 certificate defined in [7]. To use certificate mode, node should have a list of root trust anchor to verify the certificate
- *NoSec* mode: DTLS is not used in this mode. When no security mode is used, other layers of the communication should take care of security.

To apply multicast group communication using CoAP to IoT environments, [8] specifies new features such as how to process CoAP functions and the way to reuse of token values for the CoAP group communication. However, security for the CoAP multicast group communication is not defined since DTLS protocol cannot be used as a multicast security protocol. Therefore, only *NoSec* mode is used for group communication, and only messages not important are sent over multicast for now. To solve this problem several proposals are suggested in the standardization working groups, CoRE and Authentication and Authorization for Constrained Environments (ACE).

B. Authentication and Authorization for joining devices

For the group communication, only authenticated and authorized member should be allowed to get security attributes for establishing a secure communication channel. To authenticate and authorize the IoT devices, ACE working group developed ACE framework [9]. The ACE framework consists of four building blocks; that are OAuth 2.0, CoAP, Concise Object Representation (CBOR) [10] and CBOR Object Signing and Encryption (COSE) [11]. Figure 1 shows message flow of ACE framework. In the framework, it assumes that the client and resource server (RS) are registered to an authorization server (AS) in a provisioning phase. It also assumes that they share credentials and configuration parameters in a secure way during the provisioning phase.
(1) Client requests an access token (AT) to AS
(2) AS sends AT to the client
(3) Client forwards the AT to RS
(4) RS sends the AT to the AS to inspect
(5) AS sends result of the AT inspection
(6) If the AT is valid, RS sends response to the client

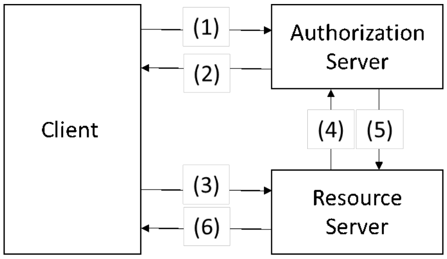

Fig. 1. ACE framework message flow

Based on the ACE framework, [12] proposed a device joining process to a group (see figure 2). In this proposal, the flow (1) and (2) are the same as in the ACE framework flow but it includes additional device information such as the group id of group the device wants to join, and the role of the joining device. If it is a sender or receiver. In the message flow (3), the device posts the token. If token is valid, client and GM create secure channel (4). After creating secure channel, the device sends a group join request (5), GM responses with keying material the device can use for group communication with certain group members (6). When the joining process is finished, the joining device is authenticated, authorized and registered at the GM. The same as ACE framework, it also assumes that joining node is registered to the AS and shares a security material with it in advance.

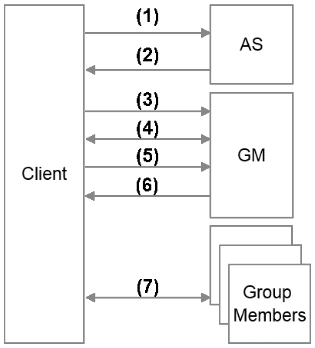

Fig. 2. Joining process message flow

As another Internet Draft (I-D) for secure group communication regarding authorization and key distribution, [13] introduced two different schemes; asymmetric key based scheme and symmetric key based authentication and communication. It also assumes certificate and corresponding private key is configured in a provisioning phase, and it is used when device communicates with AS. The proposed architecture has a key distribution center (KDC) instead of a GM introduced in [12]. When a device is authenticated and authorized by KDC, the device gets another access token for resources that the device wants to access. This proposal [13] is submitted to ACE working group but it does not use the ACE framework for authorization and authentication.

C. Secure group communication

Link layer
IEEE 802.15.4 is a standard specifying the way to operate low-rate wireless personal area networks (LR-WPAN). Since the standard mainly focuses on low cost and low power network between constrained devices, it is considered as a fundamental network protocol for the IoT stacks in Zigbee, WirelessHART and Thread. IEEE 802.15.4 provides data integrity relying on 128bit AES. However the way to exchange the security key between devices is not part of the specification. The key management, device authentication and security schemes are supposed to be provided from upper layers of the communication protocol. In addition, the security scheme defined by the standard cannot fully protect network and devices from a DoS, replay or spoofed ACK attack [14]. On the link layer level, the encryption scheme only can be used for member authentication joined a certain network with a network key. It cannot be used for secure multicast communication. Having only link layer security is therefore not a viable option within an IoT environment.

Network layer
IPsec is a network layer security protocol providing end-to-end security between devices, gateways or a device and a gateway by using two different protocols; Authentication Header (AH) and Encapsulating Security Payload (ESP) with two mode of operations [15]. Each protocol can use either tunnel mode that whole IP packet including IP address is protected or transport mode that only payload of the IP packet is protected depending

on security requirement. To support IPsec security protocol for multicast communication, [16] introduces an extension. This extension supports both Any-Source Multicast (ASM) and Source-Specific Multicast (SSM). It is only activated when the packet is sent as IP multicast packets. In the extension, it describes new header construction semantics for tunnel mode with address preservation. To fully support multicast communication, it also describes Group Key Management Subsystem (GKMS) and extended IPsec databases such as group security policy database (GSPD) and Group Peer Authorization Database (GPAD). IPsec extension can be used as an alternative security protocol for multicast communication. However in constrained IoT network environment composed of devices running over IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), the extension cannot be supported. [17] proposed header compression mechanisms for using IPsec in constrained environment, but it was not adopted as a standard scheme. For adapting [16] extension to 6LoWPAN, further research is required. Even if IoT multicast device uses IPsec with the extension, it is only meaningful when all devices that are having the same multicast IP address having the same Security Association (SA) since this extension is not covering more than one SA for one IP multicast group.

Transport layer
As we discuss in Section II, binding DTLS security protocol is mandatory for the CoAP communication. Thus, we can assume that all IoT devices using CoAP have DTLS support. However, it is only designed for the unicast communication. To get benefit by reusing the protocol for the multicast communication, [18] is proposed in the DICE working group concluded in IETF. The proposal assumes that sender and receiver are in the same group and have shared DTLS keying material.

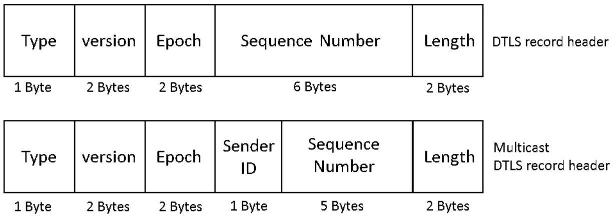

Fig. 3. DTLS record protocol header

It modified 6 bytes of DTLS record header to specify sequence number and sender ID (see Figure 3). By reusing the DTLS protocol devices, it can reduce the required memory size. However, this proposal is not standardized in the working group and without further progress, the working group is concluded. Since the proposal changes the DTLS standard protocol header, if it is not adopted as a standard protocol, it has interoperability limitation.

Application layer
Providing security on the application layer scheme named OSCORE [19] is proposed in CoRE working group. The protocol is designed to provide security where end-to-end security is broken because of the use of a proxy. OSCORE protects not only CoAP payload but also CoAP options by using COSE object. CoAP options are separated to three classes as described in Table 1. Class I and U options are remained in the CoAP header, and class U option is encrypted with payload.

TABLE I
CoAP option classes of OSCORE

Option location	Classes	Information
Inner option	Class E	Encryption
Outer option	Class I	Integrity protection
	Class U	non-protection

Figure 4 shows how OSCORE message is structed from a CoAP message to provide end-to-end security. The option E, and CoAP code and payload are placed in the plain text. The plain text is encrypted with COSE object which includes the option I. After the encryption, the OSCORE set code to POST or FETCH and add OSCORE option in the option U field. The cipher text is placed in payload field. Since it is an application layers security protocol, OSCORE can also be used for the secure multicast group communication.

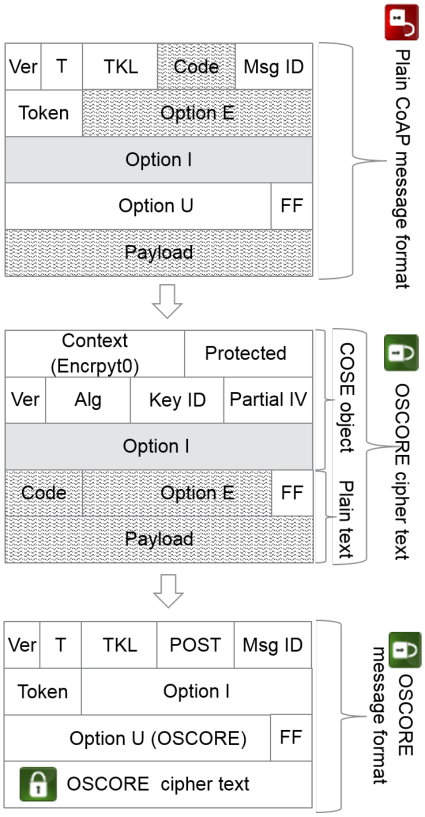

Fig. 4. OSCORE Message format

[20] defines how to use OSCORE for multicast communication. The proposal is fully aligned with OSCORE, but additional features such as having a group ID in COSE header and

using counter signature algorithm for source authentication are additionally specified. For the source authentication, Ed25519 counter signature algorithm which is defined in [19] is used. Considering the case where a receiver does not have enough computing resources to verify the signature of sender, it suggested pure receiver that does not need to verify the signature and send a response to the sender. However, in this proposal, there is no option that allows not generating signature on a sender when it is constrained device. In addition, most of the secure cryptographic processors are not supporting Edwards-curve Digital Signature Algorithm (EdDSA) so it is obstacle to use it for latency sensitive application scenarios.

IV. Evaluation

Group OSCORE is the indisputable protocol to be adopted as a standard protocol for secure IoT group communication in IETF. The alliances such as Fairhair, OCF are considering adding the group OSCORE to the specification. Especially OMA adopted OSCORE object model in their LWM2M architecture to support the protocol [23] [24] . Therefore, we evaluated group OSCORE to verify applicability for commercial lighting system. To evaluate elapsed time and CPU time, we modified Contiki OS based OSCORE implementation to group OSCORE. From the implementation, we removed Contiki OS dependencies and added to our Open Architecture for Intelligent Solid State Lighting System (OpenAIS). Note that the implementation is not based on the latest version of I-D [20] but the previous I-D version [25]. For the evaluation, we used a system on a chip (SoC) connected to and controlling actual luminaries. Table 2 shows detailed hardware specification.

TABLE II
Hardware specification

Item	Contents
Module	MT7688AN
CPU	580 MHz MIPS
Flash	32 MB
RAM	128 MB DDR2
Network	Wi-Fi 802.11 b/g/n
Battery	USB

We deployed two different type of luminaries, one is a multicaster (M) having a passive infrared (PIR) sensor and the other one is a pure Listener (L). M is triggered to send a message to the L when the sensor detect motions. It also act as a L at the same time to control the light itself. We evaluated CPU running time, with and without the security protocol. Figure 5 shows CPU time for the computation. On the sender side, we evaluated the time for preparing the multicast message without generating signature. On the listener side, the result includes the time for getting a sender information from the message header, security context generation and data decryption. We evaluated elapsed time as well including propagation time and operation idle time since elapsed time would be the time that service users experience. To get average elapsed time, we repeated evaluation process forty times. From M to L, it took avg. 26 milliseconds with group OSCORE. Without the security protocol, it took avg. 17 milliseconds. In this evaluation, signature function was omitted as well.

V. Research Challenge

In contrast with existing the internet that powerful devices are connected, in heterogeneous IoT network environment, each group member may have different resource capacity even though the devices belong to the same group for the same purpose.

- Non-constrained sender and listeners
- Non-constrained sender and constrained listeners
- Non-constrained sender and mixed listeners
- Constrained sender and non-constrained listeners
- Constrained sender and mixed listeners
- Constrained sender and listeners

Depending on the resource capabilities of the group members, security requirements and security level should be considered accordingly. In [20] proposal, it provides an option to the listeners so that listeners could discard signature verification when it has constrained computing resources. This scenario fits to the case where sender has enough resource, and listeners do not have. However, the option is not allowed to the sender. Thus the sender always has to generate signature when it sends messages to listeners. When sender is constrained and listeners are not, and when listeners send response with signature, the proposal is not feasible to apply to the case. Defining security level depends on device capabilities and giving a flexibility is required.

For the data encryption, nodes that are joined to a common multicast group need a group key. In proposal [12], it includes the way to share group key when node is joining. However, the method to provide forward secrecy and backward secrecy is not covered in current proposal. Furthermore, in current multicast architecture considered in proposals [12], [20] are based on 1:M, one sender with multiple listeners, and the proposals are only considering one device is joining one group. However, in real service scenario, sender could be more than one in a group, and a device could join more than one group

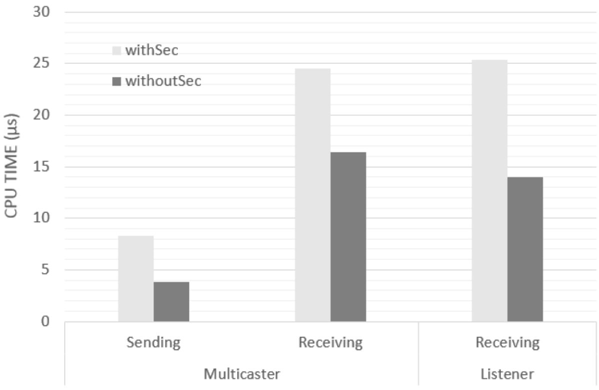

Fig. 5. CPU time comparison on multicaster and listener

at the same time. Therefore, considering the scalable and dynamic IoT architecture for group communication, the way to update group key when devices are joining or leaving a group based on current standardization architecture should be researched.

In case a device is deployed in a physically different network domain from other group members having link-local multicast IP address is registered as a member of group, the logical group member could have unicast IP address. Most of the IETF standardization works [8], [12], [13], [20] are considering group communication based on IP multicast, therefore the members having different IP address cannot be managed as group entities. To extend group communication over IP multicast to application level group communication, group managers roles should be extended and need to be researched for maintaining group keys.

VI. Conclusion

In this paper a overview of the security requirements of commercial smart lighting installations is given and how multicast group communication can be used in such architectures. One use case is how to authenticate authorize a device that wants to join a group. The other use case is how to provide data confidentiality for secure communication based on each TCP/IP layer. In section V, we discussed existing proposals considering different architectures where nodes having different computing resources that are joining and leaving dynamically. Nevertheless there is high demand for security solution for group communication, the solution that meets security requirements from industry service scenarios is not fully fulfilled so far. This paper provides an overview about current status of standardization work for secure IoT group communication, a practical evaluation of the current OSCORE draft and applicability in commercial smart lighting.

References

[1] Z. Shelby, K. Hartke, C. Bormann, “The Constrained Application Protocol,” RFC 7959, 2014.

[2] Z. Shelby, M. Koster, C. Bormann, P. van der Stok, C. Amsuess, “CoRE Resource Directory,” draft-ietf-core-resource-directory-14, 2018.

[3] P. S. Kruus, ”A survey of multicast security issues and architectures,” NAVAL RESEARCH LAB WASHINGTON DC, 1998.

[4] T. Hardjono, G. Rsudik, “IP multicast security : issues and directions,” In Annales des tlcommunications, vol. 55, no. 7-8, Springer-Verlag, 2000, pp. 324-340.

[5] M.Moyer, J. Rao and P. Rohatgi, “A survey of security issues in multicast communications.” IEEE network 13, no. 6, 1999 pp.12-23.

[6] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, T. Kivinen, “Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS),” RFC7250, 2014.

[7] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” RFC5280, 2008.

[8] A. Rahman, E. Dijk, “Group Communication for the Constrained Application Protocol,” RFC 7390, 2014.

[9] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, H. Tschofenig, “Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth),” draft-ietf-ace-oauth-authz-11, 2018.

[10] C. Bormann, P. Hoffman, “Concise Binary Object Representation (CBOR),” RFC 7049, 2013.

[11] J. Schaad, “CBOR Object Signing and Encryption (COSE),” RFC 8152, 2017.

[12] M. Tiloca, J. Park, “Joining OSCORE groups in ACE,” draft-tiloca-ace-oscoap-joining-04, 2018.

[13] H. Tschofenig, W. Werner, “Security for Low-Latency Group Communication,” draft-tschofenig-ace-group-communication-security-00, 2017.

[14] A. Reziouk, E. Laurent, JC Demay, “Practical security overview of IEEE 802.15.4,” Engineering & MIS (ICEMIS), International Conference on, pp. 1-9, IEEE, 2016.

[15] S. Kent, K. Seo, “Security Architecture for the Internet Protocol,” RFC 4301, 2005.

[16] B. Weis, G. Gross, D. Ignjatic, “Multicast Extension to the Security Architecture for the Internet Protocol,” RFC 5374, 2008.

[17] S. Raza, S. Duquennoy, G. Selander, “Compression of IPsec AH and ESP Headers for Constrained Environments,” draft-raza-6lowpan-ipsec-01, 2013.

[18] S. Keoh, S. Kumar, O. Garcia-Morchon, A. Rahman, “DTLS-based Multicast Security in Constrained Environments,” draft-keoh-dice-multicast-security-08, 2014.

[19] G. Selander, J. Mattsson, F. Palombini, L. Seitz, “Object Security for Constrained RESTful Environments (OSCORE),” draft-ietf-core-object-security-15, 2018.

[20] M. Tiloca, G. Selander, F. Palombini, J. Park, “Secure group communication for CoAP,” draft-ietf-core-oscore-groupcomm-02, 2018.

[21] P. Savola, “Overview of the Internet Multicast Routing Architecture,” RFC 5110, 2008.

[22] B. Pronk, F. van Tuijl, “Integrating Lighting in the Internet of Things,” (accessed Sep. 2018). [Online]. Available: http://www.openais.eu/user/file/openais-integrating_lighting_in_the_internet_of_things-ledprofessionalreview67.pdf.

[23] “Lightweight Machine to Machine Technical Specification: Core,” Candidate Version :1.1 (accessed Oct. 2018). [Online]. Available: http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180612-C/OMA-TS-LightweightM2M_Core-V1_1-20180612-C.pdf

[24] “Lightweight Machine to Machine Technical Specification: Transport Bindings,” Candidate Version :1.1 (accessed Oct. 2018). [Online]. Available: http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180612-C/OMA-TS-LightweightM2M_Transport-V1_1-20180612-C.pdf

[25] M. Tiloca, G. Selander, F. Palombini, “Secure group communication for CoAP,”draft-tiloca-core-multicast-oscoap-03, 2017.