DICE Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 8, 2014

S. Keoh
University of Glasgow
S.S. Kumar, Ed.
O. Garcia-Morchon
E. Dijk
Philips Research

November 4, 2013

DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs) draft-keoh-dice-multicast-security-01

Abstract

Wireless IP-based systems will be increasingly used for building control systems in the future where wireless devices interconnect with each other, forming low-power and lossy networks (LLNs). The CoAP and 6LoWPAN standards are emerging as the de-facto protocols in this area for resource-constrained devices. Both multicast and security are key needs in these networks. This draft presents a method for securing IPv6 multicast communication in LLNs based on the DTLS which is already available in CoAP devices. This draft deals with the adaptation of the DTLS record layer to protect CoAP group communication, assuming that all group member devices are already configured with the group security association. The DTLS record layer is used to provide authentication and encrypt multicast messages using the group keying material before sending the message via IPv6 multicast to the group.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 May 8, 2014.

Table of Contents

1. Introduction

There is an increased use of wireless control networks in city infrastructure, environmental monitoring, industrial automation, and building management systems. This is mainly driven by the fact that the independence from physical control wires allows for freedom of placement, portability and for reducing the cost of installation as less cable placement and drilling are required. Consequently, there is an ever growing number of electronic devices, sensors and actuators that have become Internet connected, thus creating a trend towards Internet of Things (IoT). These connected devices are equipped with communication capability that enables them to interact with each other as well as with Internet services at anytime and anyplace. However, the devices in such wireless control networks are usually battery-operated or powered by scavenged energy, they have limited computational resources (low CPU clock, small RAM and flash storage) and often, the communication bandwidth is limited (e.g., IEEE 802.15.4 radio), and also the transmission is unreliable. Hence, such wireless control networks are also known as Low-power and Lossy Networks (LLNs).

In addition to the usual device-to-device unicast communication that would allow devices to interact with each other, group communication is an important feature in LLNs that can be effectively used to convey messages to a group of devices without requiring the sender to perform time- and energy-consuming multiple unicast transmissions to reach group members. For example, in a building control management system, Heating, Ventilation and Air-Conditioning (HVAC) and lighting devices can be grouped according to the layout of the building, and control commands can be issued to a group of devices. Group communication for LLNs has been made possible using the Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] based on IP-multicast.

Currently, CoAP can be protected using Datagram Transport Layer Security (DTLS) [RFC6347]. However, DTLS is mainly used to secure a connection between two endpoints and it cannot be used to protect multicast group communication. We believe that group communication in LLNs is equally important and should be secured as it is also vulnerable to the usual attacks over the air (eavesdropping, tampering, message forgery, replay, etc). Although there have been a lot of efforts in IETF to standardize mechanisms to secure multicast communication, they are not necessarily suitable for LLNs which have much more limited bandwidth and resources. For example, the MIKEY Architecture [RFC3830] is mainly designed to facilitate multimedia distribution, while TESLA [RFC4082] is proposed as a protocol for broadcast authentication of the source and not for protecting the confidentiality of multicast messages.

This draft describes an approach to use DTLS as mandated in CoAP to support multicast security. It assumes that all devices in the group share a security parameters and keying material, for e.g., it can be distributed by a controller in the network through a DTLS unicast secure channel to each device in the group. This draft focuses only on the use of DTLS record layer to protect multicast messages to be sent to the group, and thus providing integrity, confidentiality and authenticity to the IP multicast messages in the LLN.

1.1. Terminology

This specification defines the following terminology:

Controller: The entity that is responsible for creating a multicast group, adding members, and distributing keying material to members of the group. It is also responsible for renewing/updating the multicast group keying material. It is not necessarily the sender in the multicast group.

Sender: The entity that sends multicast messages to the multicast group.

Listener: The entity that receives multicast messages when listening to a multicast IP address.

1.2. Outline

This draft is structured as follows: Section 2 motivates the proposed solution with multicast use cases in LLNs and derives a set of requirements. Section 3 provides an overview of the DTLS-based multicast security. In Section 4, we describe the use of DTLS record layer to encrypt and integrity protect multicast messages assuming that all devices in the group already have a security parameters and group keying material in possession. Section 5 and Section 6 describe Security and IANA considerations.

2. Use Cases and Requirements

This section defines the use cases for multicast and specifies a set of security requirements for these use cases.

2.1. Use Cases

As stated in the Group Communication for CoAP Internet Draft [I-D.ietf-core-groupcomm] in the IETF CoRE WG, multicast is essential in several application use cases. Consider a building equipped with 6LoWPAN [RFC4944] IP-connected lighting devices, switches, and 6LoWPAN border routers; the devices are organized as groups according

to their location in the building, e.g., lighting devices and switches in a room/floor can be configured as a multicast group, the switches are then used to control the lighting devices in the group by sending on/off/dimming commands to the group. 6LoWPAN border routers that are connected to an IPv6 network backbone (which is also multicast enabled) are used to interconnect 6LoWPANs in the building. Consequently, this would also enable multicast groups to be formed across different subnets in the entire building. The following lists a few multicast group communication uses cases in a building management system; a detailed description of each use case can be found in Group Communication for CoAP Internet Draft [I-D.ietf-core-groupcomm].

a. Lighting control: enabling synchronous operation of a group of 6LoWPAN connected lights in a room/floor/building. This ensures that the light preset of a large group of luminaries are changed at the same time, hence providing a visual synchronicity of light effects to the user.

b. Firmware update: firmware of devices in a building or a campus control application are updated simultaneously, avoiding an excessive load on the LLN due to unicast firmware updates.

c. Parameter update: settings of devices are updated simultaneously and efficiently.

d. Commissioning of above systems: information about the devices in the local network and their capabilities can be queried and requested, e.g. by a commissioning device.

2.2. Security Requirements

The Miscellaneous CoAP Group Communication Topics Internet Draft [I-D.dijk-core-groupcomm-misc] has defined a set of security requirements for group communication in LLNs. We re-iterate and further describe those security requirements in this section with respect to the use cases as presented in Section 2.1:

a. Multicast communication topology: We consider both one-to-many and many-to-many communication topologies in this draft. The one-to-many communication topology is the simplest group communication scenario that would serve the needs of a typical LLN. For example, in the lighting control use case, the switch is the only entity that is responsible for sending control commands to a group of lighting devices. These lighting devices are actuators that do not issue commands to each other. In other use cases, a many-to-many multicast communication topology would be required, in particular multiple sensors and actuators are

part of a multicast group and these sensors will trigger events to the group in order to notify the interested parties. Devices in the group could also send commands in order to trigger some actions on other devices in the group.

b. Establishment of a Group Security Association (GSA) [RFC3740]: A secure channel must be used to distribute keying material, multicast security policy and security parameters to members of a multicast group. A GSA must be established between the controller (which manages the multicast group and may be a different device than the sender) and the group members. The 6LoWPAN border router, a device in the 6LoWPAN, or a remote server outside the 6LoWPAN could play the role of controller for distributing keying materials. Since the keying material is used to derive subsequent group keys to protect multicast messages, it is important that it is encrypted, integrity protected and authenticated when it is distributed. However, this is out of scope of this draft, and it is anticipated that an activity in IETF dedicated to the design of a generic key management scheme for the LLN will be started in the future.

c. Multicast security policy: All group members must use the same ciphersuite to protect the authenticity, integrity and confidentiality of multicast messages. The ciphersuite can either be negotiated or set by the controller and then distributed to the group members. It is generally very complex and difficult to require all devices to negotiate and agree with each other on the ciphersuite to be used, it is therefore more effective that the multicast security policy is set by the controller.

d. Multicast data group authentication: It is essential to ensure that a multicast message is originated from a member of the group. The multicast group key which is known to all group members is used to provide authenticity to the multicast messages (e.g., using a Message Authentication Code, MAC). This assumes that only the sender of the multicast group is sending the message, and that all other group members are trusted not to tamper with the multicast message.

e. Multicast data source authentication: Source authenticity is optional. It can typically be provided using public-key cryptography in which every multicast message is signed by the sender. This requires much higher computational resources on both the sender and the receivers, thus incurring too much overhead and computational requirements on devices in LLNs. Alternatively, a lightweight broadcast authentication, i.e., TESLA [RFC4082] can be deployed, however it requires devices in

the multicast group to have a trusted clock and have the ability to loosely synchronize their clocks with the sender. Consequently, given that the targeted devices have limited resources, and the need for source authenticity is not critical, it is advocated that source authenticity is made optional.

f. Multicast data integrity: A group level integrity is required to ensure that messages have not been tampered with by attackers who are not members of the multicast group.

g. Multicast data confidentiality: Multicast message may be encrypted, as some control commands when sent in the clear could pose privacy risks to the users.

h. Multicast data replay protection: It must not be possible to replay a multicast message as this would disrupt the operation of the group communication.

i. Multicast key management: Group keys used to protect the multicast communication must be renewed periodically. When members have left the multicast group, the group keys might be leaked; and when a device is detected to have been compromised, this also implies that the group keys could have been compromised too. In these situations, the controller must perform a re-key protocol to renew the group keys. This work will be addressed as part of the key management for LLN in the future based on [RFC3740] and [RFC4046].

3. Overview of DTLS-based Secure Multicast

The goal of this draft is to secure COAP group communication over 6LoWPAN networks, by extending the use of the DTLS security protocol to allow for the use of DTLS record layer to provide protection to multicast messages. The IETF CoRE WG has selected DTLS [RFC6347] as the default must-implement security protocol for securing CoAP, therefore it is conceivable that DTLS can be extended to facilitate CoAP-based group communication. Reusing DTLS for different purposes while guaranteeing the required security properties can avoid the need to implement multiple security protocols and this is especially beneficial when the target deployment consists of resource-constrained embedded devices. This section first describes group communication based on IP multicast, and subsequently sketches a solution for securing group communication using DTLS.

3.1. IP Multicast in LLN

Devices in the LLN are categorized into two roles, (1) sender and (2)

listener. Any node in the LLN may have one of these roles, or both roles. The application(s) running on a device basically determine these roles by the function calls they execute on the IP stack of the device.

In principle, a sender or listener does not require any prior access procedures or authentication to send or listen to a multicast message [RFC5374]. A sender to an IPv6 multicast group sets the destination of the packet to an IPv6 address that has been allocated for IPv6 multicast. A device becomes a listener by "joining" to the specific IPv6 multicast group by registering with a network routing device, signaling its intent to receive packets sent to that particular IPv6 multicast group. Any device can in principle decide to listen to any IPv6 multicast address. This also means applications on the other devices do not know, or do not get notified, of new senders or listeners in the LLN. More details on the IPv6 multicast and CoAP group communication can be found in [I-D.ietf-core-groupcomm]. This draft does not intend to modify any of the underlying group communication or multicast routing protocols.

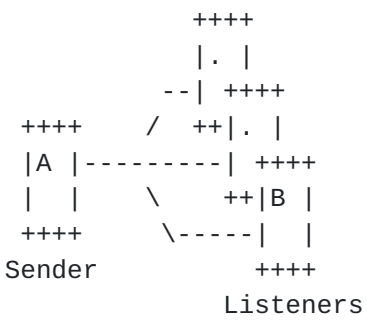

Figure 3.1: The roles of nodes in a one-to-many multicast communication topology

3.2. Securing Multicast in LLNs

A controller in an LLN creates a multicast group. The controller may be hosted by a remote server, or a border router that creates a new group over the network. In some cases, devices may be configured using a commissioning tool that mediates the communication between the devices and the controller. The controller in the network can be discovered by the devices using various methods defined in [I-D.vanderstok-core-dna] such as DNS-SD [RFC6763] and Resource Directory [I-D.ietf-core-resource-directory]. The controller communicates with individual device to add them to the new group. Additionally, the controller can distribute a Group Security Association (GSA) consisting of keying material, security policies and security parameters to use, to all the member devices in the group, e.g., by establishing a secure DTLS channel with each device.

As mentioned previously, a standardized way of performing key management for LLN is out of scope of this draft, and we assumes that each device in the group has been configured with a GSA using a.

Senders in the group can encrypt and authenticate application messages using the keying material in the DTLS record layer before it is sent using IP multicast. For example, a CoAP message addressed to a multicast group is protected using DTLS record layer and then sent to a multicast group. The listeners when receiving the message, use the multicast IP destination address (i.e., Multicast identifier) to look up the GSA needed for that connection. The received message is decrypted and the authenticity is verified using the keying material for that connection.

4. Multicast Data Security

This section describes in detail the use of DTLS record layer to secure multicast messages. This assumes that group membership has been configured by the controller, and all devices in the group have been configured with the GSA. Since the exact details of the group key management are outside the scope of this draft, we assume that the GSA can be used to derive the same SecurityParameters structure as defined in [RFC5246] for all devices. Additional ciphersuites may need to be defined to convey the bulk cipher algorithm, MAC algorithm and key lengths within the key management protocol. We provide two such examples of ciphersuites that could be defined as part of a future key management mechanism:

```
Ciphersuite MTS_WITH_AES_128_CCM_8 = {TBD1, TBD2}
Ciphersuite MTS_WITH_NULL_SHA256   = {TBD3, TBD4}
```

Ciphersuite MTS_WITH_AES_128_CCM_8 is used to provide confidentiality, integrity and authenticity to the multicast messages where the encryption algorithm is AES [AES], key length is 128-bit, and the authentication function is CCM [RFC6655] with a Message Authentication Code (MAC) length of 8 bytes. Similar to RFC4785 [RFC4785], the ciphersuite MTS_WITH_NULL_SHA is used when confidentiality of multicast messages is not required, it only provides integrity and authenticity protection to the multicast message. When this ciphersuite is used, the message is not encrypted but the MAC must be included in which it is computed using a HMAC [RFC2104] that is based on Secure Hash Function SHA256 [SHA]. Depending on the future needs, other ciphersuites with different cipher algorithms and MAC length may be supported.

Apart from existing ciphersuites defined for (D)TLS, new ciphersuites based on AERO[ID.mcgrew-aero] which are particularly designed to support multiple senders, may be more suitable. More work needs to be

done in future to identify the usage of AERO with the DTLS record layer protection for group communication specified in this draft .

The SecurityParameters.ConnectionEnd should be set to "server" for senders and "client" for listeners. The current read and write states can be derived from SecurityParameters by generating the six key material items:

client write MAC key
server write MAC key
client write encryption key
server write encryption key
client write IV
server write IV

This requires that the client_random and server_random within the SecurityParameters are set same for all devices as part of the key management protocol to derive the same keying material for all devices in the group with the PRF function defined in Section 6.3 of [RFC5246] . Alternatively, the key management protocol could directly provide the above six key material to all group devices as part of the GSA.

The current read and write states are instantiated for all group members based on the keying material; senders use "server write" parameters for the write state and listeners use "server write" parameters for the read state. Additionally each connection state contains the sequence number which is incremented for each record sent; the first record sent has the sequence number 0.

For the optional multicast data source authentication, the sender can sign the message using public key cryptography at the application layer and send it as the multicast message in the DTLS record payload. This option is independent of the DTLS layer and outside the scope of this draft.

4.1. Sending Secure Multicast Messages

All messages addressed to the multicast group must be secured using "server write" parameters. Using the DTLS record layer, multicast messages are encrypted and protected using a Message Authentication Code (MAC) according to the chosen ciphersuite. The authenticated encrypted message is passed down to the lower layer of the IP protocol stack for transmission to the multicast address.

As described in the previous section, the example ciphersuite MTS_WITH_AES_128_CCM_8 defines that the multicast message must be encrypted using AES with a 128-bit "server write encryption key". Since the CCM mode of operation is used for authenticated encryption,

the same key is used to compute the MAC. As for the ciphersuite example MTS_WITH_NULL_SHA, the multicast message must not be encrypted, but a MAC must be computed using the "server write MAC key".

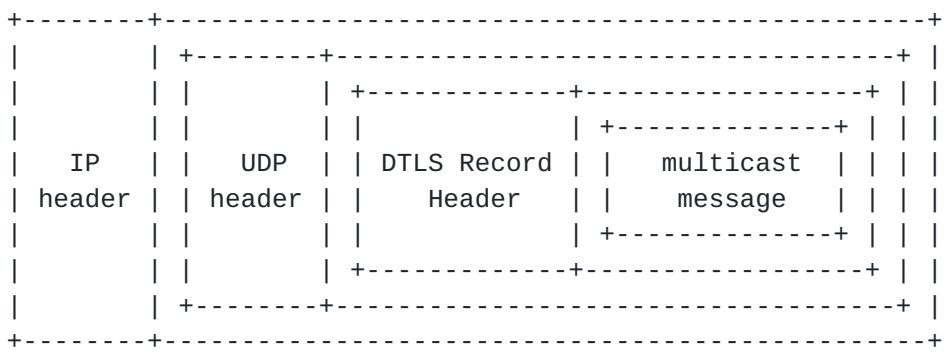

Figure 4.1: Sending a multicast message protected using DTLS Record Layer

4.1.1. One Sender, Multiple Listeners Multicast Group

This section describes the use of DTLS record layer to protect a one-sender, multiple-listeners multicast group communication. In this setting, it is the responsibility of the controller which configures the group membership to ensure that there is only one sender in a multicast group and other devices never send multicast messages to the same group in order to ensure the security properties of the multicast messages. This is especially a concern in AEAD cipher suites if multiple senders reuse the same nonce for encryption as described in Section 5.1.1 in [RFC5116].

The following illustrates the structure of the DTLS record layer header, the epoch and sequence number are used to ensure message freshness and to detect message replays. As there is only one sender in the multicast group, the sender is responsible for maintaining and manipulating the epoch and sequence number when sending multicast messages. The receivers in the group are "trusted" not to tamper with these parameters.

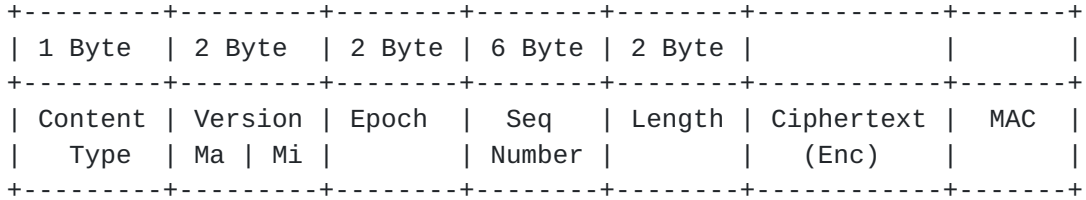

Figure 4.2: The DTLS record layer header and optionally encrypted payload and MAC

The sequence number is initialized to 0, and it is increased by one whenever the sender sends a new multicast record message. This is the standard behavior of the current DTLS in order to detect message replay. The sender or the controller can increase the epoch number by sending a ChangeCipherSpec message whenever the sequence number has been exhausted, or whenever the ciphersuite has been changed in order to reset the sequence number. Finally, the multicast message is protected (encrypted if needed, and authenticated with a MAC) using the "server write" parameters.

4.1.2. Multiple Senders, Multiple Listeners Multicast Group

There is a need to support multi-senders in group communication. In particular, in a lighting network there are multiple presence sensors that would be assigned the sender role as they are responsible for multicasting the presence information to the luminaries in the group. In this section, we outline an approach to enable all senders in the group to securely send information using a common group key, while preserving the freshness and integrity of the messages.

One of the main problems with supporting multiple senders using a single key is that it leads to nonce reuse AEAD cipher suites like AES-CCM[RFC6655] and AES-GCM[RFC5288]. Nonce reuse can completely break the security of these cipher suites. According to the AES-CCM for TLS Section 3 [RFC6655], the CCMNonce is a combination of a salt value and the sequence number.

```
struct {
        opaque salt[4];
        opaque nonce_explicit[8];
} CCMNonce;
```

The salt is the "client write IV" (when the client is sending) or the "server write IV" (when the server is sending) as defined in the "SecurityParameters". Further [RFC6655] requires that the value of the nonce_explicit MUST be distinct for each distinct invocation of the CCM encrypt function for any fixed key. When the nonce_explicit is equal to the sequence number of the TLS packets, the CCMNonce has the structure as below:

```
struct {
        uint32 client_write_IV; // low order 32-bits
        uint64 seq_num;         // TLS sequence number
} CCMClientNonce.

struct {
        uint32 server_write_IV; // low order 32-bits
        uint64 seq_num; // TLS sequence number
} CCMServerNonce.
```

In DTLS, the 64-bit sequence number is the 16-bit epoch concatenated with the 48-bit seq_number.

Therefore to prevent that CCMNonce is reused, either all senders need to synchronize or seperate non-overlapping sequence number spaces need to be created for each sender. Synchronization between senders is especially hard in LLN and therefore we go for the second approach by creating different sequence number spaces by embedding unique sender identifiers in the sequence number as suggested in [RFC5288].

Therefore in addition to configuring each device in the group with the GSA, the controller needs to assign a unique SenderID (represented as two octets) to each device which has the sender role in the group. The list of SenderIDs are then distributed to all the group members by the controller. Alternatively, this setup procedure can be eliminated by allowing senders to derive their SenderIDs themselves based on the device's IPv6 or MAC address, or even randomly. The specific method to be used is not defined here, except care should be taken that it would lead to a high probability of unique SenderIDs for all senders within the specific multicast group. To overcome potential clash in SenderIDs, a back-off mechanism is defined in the Security Considerations section.

The existing DTLS record layer header is adapted such that the 6-byte sequence number field is split into a 2-byte SenderID field and a 4-byte "truncated" sequence number field. Each sender in the group uses its own unique SenderID in the DTLS record layer header when sending a multicast message to the group. It also manages its own epoch and "truncated" sequence number in the "server write" connection state, hence they do not need to synchronize them with other senders in the group. Figure 4.3 illustrates the adapted DTLS record layer header.

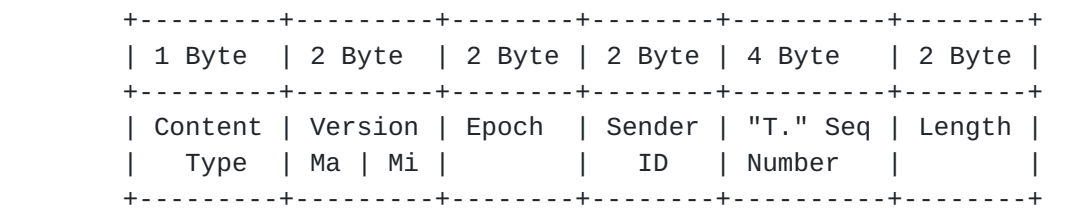

Figure 4.3: The adapted DTLS record layer header

4.2. Receiving Secure Multicast Messages

4.2.1. One Sender, Multiple Listeners Multicast Group

When a listeners receives a protected multicast message from the

sender, it looks up the corresponding "client read" connection state based on the multicast IP destination of the packet. This is fundamentally different from standard DTLS logic in that the current "client read" connection state is bound to the source IP address. However, given that this is a one sender- multiple listeners communication topology, it is possible to bind the current "client read" connection state to the source IP address if it is already known to all listeners. Therefore a lookup based on the source IP address is also possible in this case.

The listeners authenticate and decrypt the multicast message using the "server write" keys. The verification of MAC ensures that the payload and the DTLS Record Layer header have not been tampered with. As there is only one sender, and all other group members are "trusted", only the sender is able to manipulate the epoch and the sequence number, hence once the DTLS header has been authenticated, the epoch and the sequence number can be sufficiently trusted to detect any message replay.

4.2.2. Multiple Senders, Multiple Listeners Multicast Group

Listener devices in a multi-senders multicast group, need to store multiple "client read" connection states for the different senders linked to the SenderIDs. The keying material is same for all senders however the epoch and the "truncated" sequence number of the last received packets needs to be kept different for different senders. The listeners first perform a "server write" keys lookup by using the multicast IP destination address of the packet. By knowing the keys, the listeners decrypt and check the MAC of the message. This guarantees that no one has spoofed the SenderID, as it is protected by the MAC. Subsequently, by authenticating the SenderID field, the listeners retrieve the "client read" connection state which contains the last stored epoch and "truncated" sequence number of the sender, which is used to check the freshness of the message received. The listeners must ensure that the epoch is the same and "truncated" sequence number in the message received is higher than the stored value, otherwise the message is discarded. As each sender manages its own epoch and sequence number, receivers are confident that these values are reliable. Once the authenticity and freshness of the message have been checked, the listeners can pass the message to the higher layer protocols. The epoch and the sequence number in the corresponding "client read" connection state are updated as well.

Listeners who are late joiners to a multicast group, do not know the current epoch and sequence number being used by different senders. When they receive a packet from a sender with a random sequence numbered in it, it is impossible for the listener to verify if the packet is fresh and has not been replayed by an attacker. To overcome

this late joiner security issue, we can use the techniques similar AERO [ID.mcgrew-aero] where the late joining listener on receiving the first packet from a particular sender, initialize its last seen epoch and sequence number in the "client read" state, however does not pass it to the application and drops this packet. This provides a reference point to identify if future packets are fresher than the last seen packet.

5. IANA Considerations

tbd

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

This document discusses various design aspects for multicast security in LLNs. As such this document, in entirety, concerns security.

Section 4.1.2 with multiple senders require that SenderIDs are unique to maintain the security properties of the DTLS record layer messages. However in the event that two or more senders are configured with the same SenderID, a mechanism needs to be present to avoid a security weakness and recover from the situation. One such mechanism is that all senders of the mutlicast group are also listeners. This allows a sender which receives a packet from a different device with its own SenderID in the DTLS header to be aware of a clash in SenderID. Once aware, the sender can inform the controller on a secure channel about the clash along with the source IP address. The controller can then provide a different SenderID to either device or both.

Section 4.1.2 additionally truncates the sequence number from 6 octets to 4 octets. This reduction of the sequence number space should be taken into account to ensure that epoch is incremented before the "truncated" sequence number wraps over. This should be done with an appropriate key management mechanism which is not defined in this draft.

7. Acknowledgements

The authors greatly acknowledge discussion, comments and feedback from Dee Denteneer, Peter van der Stok and Zach Shelby. Additionally thank David McGrew for suggesting options for recovering from a SenderID clash, and John Foley for the extensive review and pointing us to the AERO draft. We also appreciate prototyping and implementation efforts by Pedro Moreno Sanchez who worked as an intern at Philips Research.

8. References

8.1. Normative References

[AES] National Institute of Standards and Technology, , "Specification for the Advanced Encryption Statndard (AES)", FIPS 197, Nov 2001.

[SHA] National Institute of Standards and Technology, , "Secure Hash Standard", FIPS 180-2, Aug 2002.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, July 2012.

[RFC5288] Salowey, J., Choudhury, A., and D. McGrew, "AES Galois Counter Mode (GCM) Cipher Suites for TLS", RFC 5288, August 2008.

[RFC5116] McGrew, D., "An Interface and Algorithms for Authenticated Encryption", RFC 5116, January 2008.

8.2. Informative References

[I-D.mcgrew-aero]
McGrew, D. and Foley, J., "Authenticated Encryption with Replay prOtection (AERO)", draft-mcgrew-aero-00 (work in progress), October 2013.

[I-D.dijk-core-groupcomm-misc]
Dijk, E. and A. Rahman, "Miscellaneous CoAP Group Communication Topics", draft-dijk-core-groupcomm-misc-04 (work in progress), June 2013.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.ietf-core-groupcomm]
Rahman, A. and E. Dijk, "Group Communication for CoAP", draft-ietf-core-groupcomm-16 (work in progress), October 2013.

[I-D.ietf-tls-oob-pubkey]
Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., and T. Kivinen, "Out-of-Band Public Key Validation for Transport Layer Security (TLS)", draft-ietf-tls-oob-pubkey-09 (work in progress), July 2013.

[I-D.ietf-core-resource-directory]
Shelby, Z., Krco, S., and C. Bormann, "CoRE Resource Directory", draft-ietf-core-resource-directory-00 (work in progress), June 2013.

[I-D.vanderstok-core-dna]
Stok, P., Lynn, K., and A. Brandt, "CoRE Discovery, Naming, and Addressing", draft-vanderstok-core-dna-02 (work in progress), July 2012.

[RFC4082] Perrig, A., Song, D., Canetti, R., Tygar, J., and B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", RFC 4082, June 2005.

[RFC4785] Blumenthal, U. and P. Goel, "Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)", RFC 4785, January 2007.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, September 2007.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security

Architecture", RFC 3740, March 2004.

[RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, November 2008.

[RFC4046] Baugher, M., Canetti, R., Dondeti, L., and F. Lindholm, "Multicast Security (MSEC) Group Key Management Architecture", RFC 4046, April 2005.

Authors' Addresses

Sye Loong Keoh
University of Glasgow Singapore
Republic PolyTechnic, 9 Woodlands Ave 9
Singapore 838964
SG

Email: SyeLoong.Keoh@glasgow.ac.uk

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

Email: sandeep.kumar@philips.com

Oscar Garcia-Morchon
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: oscar.garcia@philips.com

Esko Dijk
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: esko.dijk@philips.com