TLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2013

S. Keoh
O. Garcia-Morchon
S. Kumar
E. Dijk
Philips Research

October 15, 2012

DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)
draft-keoh-tls-multicast-security-00

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/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 multicast communication in LLNs based on the DTLS security protocol which is already present in CoAP devices. This is achieved by using unicast DTLS-protected communication channel to distribute keying material and security parameters to group members. Group keys consisting of a Traffic Encryption Key (TEK) and a Traffic Authentication Key (TAK) are generated by group members based on the keying material received. A group member uses its DTLS record layer implementation to encrypt a multicast message and provide message authentication using the group keys before sending the message via IP 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 April 18, 2013.

Copyright Notice

Copyright (c) 2012 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

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) [RFC4347]. 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. The secure channel established with DTLS is used to distribute keying material (including a TEK Generation Key (TGK), security parameters, multicast security policy) to members of a multicast group, which then allows a group member to securely generate group keys, known as Traffic Encryption Key (TEK) for multicast encryption/decryption and Traffic Authentication Key (TAK) for multicast authentication. Multicast messages are protected using the DTLS record layer in order to provide integrity, confidentiality and authenticity to the IP multicast messages in the LLN.

1.1. Terminology

This specification defines the following terminology:

Crypto Session ID (CS_ID): Unique identifier for a secure multicast session.

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 keys. 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.

Group Security Association (GSA): A bi-directional secure channel between the controller and the member device that guarantees the confidentiality, integrity and authenticity of the messages exchanged between them.

TEK Generation Key (TGK): A bit string generated randomly and then distributed by the controller to all members of a multicast group. From the TGK, the multicast group keys (Traffic Encryption Key and Traffic Authentication Key) can then be generated.

Traffic Encryption Key (TEK): The key used to encrypt the multicast message.

Traffic Authentication Key (TAK): The key used to compute the Message Authentication Code (MAC) of the multicast message.

PRF(k,x): A keyed pseudo-random function.

||: Denotes concatenation of two bit strings.

XOR: Exclusive OR

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 creation of a group security association (GSA) using DTLS to distribute keying materials, and the generation of group keys based on the MIKEY Architecture [RFC3830]. Section 5 proposes the use of DTLS record layer to encrypt and integrity protect multicast messages, while Section 6 discusses the group key renewal. Section 7 and Section 8 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] [RFC6282] 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 luminaires 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 only consider a one-to-many communication topology in this draft where there is only one sender device sending multicast messages to the group. This 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. Although in other use cases, a many-to-many multicast communication topology would be required, it is much more complex and it poses greater security challenges, therefore considered as out of scope in this draft.

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.

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 send nor to tamper with the multicast message. In a one-to-many communication topology, the lighting devices that serve as actuators only receive control commands from an authorized switch and do not issue commands to other lighting devices in the group.

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 should 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.

3. Overview of DTLS-based Secure Multicast

The goal of this draft is to secure IP multicast operations as used in 6LoWPAN networks, by extending the use of the DTLS security protocol to allow for group keys distribution, and using the DTLS record layer to provide protection to multicast messages, specifically CoAP group communication. The IETF CoRE WG has selected DTLS [RFC4347] 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 handshake 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

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 does not require any prior access procedures or authentication to send a multicast message, a sender with a valid multicast group key can essentially send a secure multicast message to the group. A device becomes a listener to a specific IP multicast group by listening to the associated IP multicast address. Any device can in principle decide to listen to any IP multicast address, and can use the associated valid group key to authenticate and decrypt the multicast messages. This also means that no prior access procedure is required to be a listener nor do applications on the other devices know, or get notified, of new listeners in the LLN.

```
                                       ++++
                                       |. |
                                     --| ++++
                            ++++    /  ++|. |
                            |A |---------| ++++
                            |  |    \    ++|B |
                            ++++     \-----|  |
                           Sender          ++++
                                         Listeners
```

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 [I-D.cheshire-dnsext-dns-sd] and Resource Directory [I-D.shelby-core-resource-directory]. The controller communicates with individual device to add them to the new group. The controller establishes a GSA with each member device by performing a DTLS handshake protocol. The estabished DTLS secure channel (DTLS session) is then used by the controller to securely distribute over the network:

a. Keying material (known as the TEK Generation Key, TGK), used for deriving multicast group keys.

b. Multicast identifier, a unique identifier for the multicast group. This is typically the multicast IP address.

c. Multicast security policy, which defines the ciphersuite for multicast encryption and authentication.

d. Security parameters, used for generating group keys.

These parameters must be the same for all members of the group. Based on the TGK and the security parameters received, each member generates a multicast Traffic Encryption Key (TEK), and a Traffic Authentication Key (TAK) to be used for the multicast session. Each member also creates a Crypto Session (CS) to store security information (e.g., TGK, TEK, TAK, multicast identifier, ciphersuite, etc) relevant to the multicast session.

A designated sender in the group can encrypt application messages using the TEK and signs the message using the TAK. The message is then encapsulated using 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 address (i.e., Multicast identifier) to look up the corresponding crypto session to obtain the TEK and TAK. The received message is decrypted using the TEK, and the authenticity is verified using the TAK.

The TEK and TAK can be renewed and updated using a re-key protocol. The controller sends new security parameters for renewing TEK and TAK over the DTLS unicast channel it has established with each group member. Using the secure unicast channels provides better reliability and security as members can individually acknowledge receipts of the new security parameters, and secondly the security parameters are protected with each member's DTLS unicast session key. One of the reasons to renew the multicast group key is that the current TEK and TAK could have been compromised, hence it defeats the purpose of the re-keying process if the controller were to distribute the new security parameters via multicast. The controller has a re-key schedule and in general the controller should update the group keys when the group membership changes.

4. Multicast Group Keys Generation and Distribution

This section describes the usage of DTLS handshake protocol to establish a GSA with all group members in order to facilitate group key distribution and management. Participating devices shall have been pre-configured with a Pre-Shared Key (PSK), raw public-key [I-D.ietf-tls-oob-pubkey] or public-key certificate, preferably individual per device. When PSK and raw public key are used, they shall also be known to the controller (through an out-of-band communication channel), so that the controller is able to authenticate and establish a secure channel with each participating device.

4.1. DTLS based Group Security Association (GSA)

The controller is commissioned to set up a multicast group. The controller performs the standard DTLS handshake protocol with each participating device in order to establish a pairwise DTLS session key. Similar to the use of DTLS in CoAP [I-D.ietf-core-coap], the DTLS handshake protocol can be performed based on PSK mode, raw public key mode or public key certificate mode. In the end, the controller establishes a DTLS security channel with each member of

the multicast group in the sense that each session is distinct from the other. The DTLS handshake protocol is shown as below:

```
      Client                                               Server

                                   <--------       HelloRequest*
      ClientHello                  -------->
                                   <--------  HelloVerifyRequest*
      ClientHello (Cookie)         -------->
                                                     ServerHello
                                                    Certificate*
                                              ServerKeyExchange*
                                             CertificateRequest*
                                   <--------     ServerHelloDone
      Certificate*
      ClientKeyExchange
      CertificateVerify*
      [ChangeCipherSpec]
      Finished                     -------->
                                              [ChangeCipherSpec]
                                   <--------            Finished
```

Figure 4.1: DTLS handshake protocol

* indicates optional messages in DTLS. When PSK is used, the ServerKeyExchange message may contain a PSK Identity hint, and the ClientKeyExchange contains a PSK identity.

Depending on the implementation, both the controller and the device may be implemented as a DTLS Client or a DTLS Server. Regardless of their roles, it is advocated that the controller initiates the DTLS handshake. When the controller implements the DTLS Client, it sends a ClientHello message to the device, otherwise it sends a HelloRequest message to initiate the DTLS handshake protocol.

The established DTLS secure channel must provide both confidentiality and integrity of the messages exchanged between the controller and the member device. Through this secure channel, the controller distributes a TEK Generation Key (TGK), a multicast security policy and security parameters to the member device over the DTLS secure channel. The TGK is generated using a pseudorandom function, and it SHALL serve as the 'master' key to derive the TEK and TAK for securing multicast communication. The TGK SHALL be at least 128-bit in length. The security parameters consist of a Multicast Identifier (Mul_ID), a Crypto Session identifier (CS_ID), and a random number (RAND). In this context, the Mul_ID is the multicast address of the group, the CS_ID is a unique identifier for the crypto session and the RAND MUST be a (at least) 128-bit pseudo-random bit string.

These parameters must be the same for all members of the multicast group. This draft defines a multicast security policy which consists of only two ciphersuites to protect multicast messages. All member devices must support the following ciphersuites:

```
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], the ciphersuite MTS_WITH_NULL_SHA is used when confidentiality of multicast messages is not required, it only provides integrity and authentictiy 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.

The GSA (i.e., the DTLS secure channel) established is kept to facilitate group key renewals, thus allowing the controller to distribute new security parameters to members of the multicast group to update the group keys. This is further described in Section 6.

4.2. Generation of Group Keys

Once the member device has received the security parameters, multicast security policy and the TGK from the controller, the device generates the Traffic Encryption Key (TEK) and Traffic Authentication Key (TAK) using the Pseudo Random Function (PRF) as defined in Section 4.1 in MIKEY [RFC3830]. The TEK is used as the common group key known to all members of the group to encrypt multicast messages, while the TAK is used to create a MAC for the message. The DTLS record layer advocates the use of different key for encryption and authentication.

Similar to MIKEY [RFC3830], the following input parameters are defined:

```
inkey       : the input key to the key generation function.
inkey_len   : the length in bits of the input key.
label       : a specific label, dependent on the type of the key to be
              generated, the random number, and the session IDs.
outkey_len : desired length in bits of the output key.
```

The key generation function has the following output:

outkey: the output key of desired length.

The following defines the input parameters to the group keys generation function. These input parameters are distributed by the controller and used by the devices in a multicast group to generate group keys.

inkey   : TGK
inkey_len   : bit length of TGK
label   : constant || mul_id || cs_id || RAND
outkey_len : bit length of the output key.

As defined in MIKEY [RFC3830], the constant part of label depends on the type of key that is to be generated. The constant 0x2AD01C64 is used to generate a TEK from TGK, while the the constant 0x1B5C7973 is used to generate a TAK. The outkey_len SHALL be set to 128 bit. A crypto session should be created to store information about the multicast session, providing a mapping of the multicast identifier to the TEK, TAK, the security parameters and the multicast security policy as well as the information about the controller that is associated with the multicast session.

The following re-iterates the key generation procedure as described in MIKEY [RFC3830] with the difference that SHA256 is used instead of SHA-1.

The PRF(inkey,label) that is based on the P-function in MIKEY [RFC3830] is applied to compute the output keys (TEK and TAK):

- Let n = inkey_len / 256, rounded up to the nearest integer if not already an integer

- Split the inkey into n blocks, inkey = s_1 || ... || s_n, where all s_i, except possibly s_n, are 256 bits each

- Let m = outkey_len / 256, rounded up to the nearest integer if not already an integer

(The values "256" equal half the input block-size and full output hash size of the SHA256 as part of the P-function.)

Then, the output key, outkey, is obtained as the outkey_len most significant bits of

PRF(inkey, label) = P(s_1, label, m) XOR P(s_2, label, m) XOR ... XOR P(s_n, label, m).

5. Multicast Data Security

This section describes the use of DTLS record layer to secure multicast messages.

5.1. Sending Secure Multicast Messages

All messages addressed to the multicast group must be secured using the TEK and TAK. Using the DTLS record layer, multicast messages are encrypted using the TEK and a Message Authentication Code (MAC) is generated using the TAK according to the ciphersuite defined in the multicast security policy. The MAC is appended to the encrypted message before it is passed down to the lower layer of the IP protocol stack for transmission to the multicast address.

As described in Section 4.1, the ciphersuite MTS_WITH_AES_128_CCM_8 defines that the multicast message must be encrypted using AES with a 128-bit TEK. Since the CCM mode of operation is used for authenticated encryption, the same TEK is used to compute the MAC and the TAK is not used. As for the ciphersuite MTS_WITH_NULL_SHA, the multicast message must not be encrypted, but a MAC must be computed using the TAK key.

```
+--------+-------------------------------------------------+
|        | +--------+------------------------------------+ |
|        | |        | +-------------+------------------+ | |
|        | |        | |             | +--------------+ | | |
|   IP   | |  UDP   | | DTLS Record | |   multicast  | | | |
| header | | header | |    Header   | |    message   | | | |
|        | |        | |             | +--------------+ | | |
|        | |        | +-------------+------------------+ | |
|        | +--------+------------------------------------+ |
+--------+-------------------------------------------------+
```

Figure 5.1: Sending a multicast message protected using DTLS Record Layer

The DTLS record layer header contains a 48-bit sequence number that is used for (1) allowing the recipient to correctly verify the DTLS MAC, (2) preventing message replay. The current use of the sequence number is adequate in a one-to-many multicast communication topology. The sequence number is generated by the sender as specified in DTLS. The sequence number field in the DTLS record layer header is incremented whenever the sender sends a multicast message. This requires all member devices to keep track of the sequence number received, so that the message freshness can be verified.

5.2. Receiving Secure Multicast Messages

Member devices receiving the multicast message, look up the crypto session to find the corresponding TEK and TAK to decrypt and verify the MAC of the multicast message. The destination multicast IP address which serves as the Multicast identifier (Mul_ID) can be used to locate the crypto session in order to obtain the TEK and TAK. The crypto session must also contain the last received message's epoch and sequence number, enabling the member devices to detect message replay. Multicast messages received with a sequence number less than or equal to the value stored in the crypto session must be dropped. The epoch number in the received message must also match the epoch number stored in the corresponding crypto session. As a consequence of this mechanism, a message that arrives out-of-order (i.e. with a sequence number less than the value stored in the crypto session) will be ignored.

This replay detection mechanism only applies to one-to-many communication topology, where member devices are assumed to be trusted not to tamper with the messages.

6. Group Keys Renewal

The controller can initiate re-key of the TEK and TAK according to a key renewal schedule and when the group membership changes. It is important that the group keys, i.e., TEK and TAK are renewed periodically to prevent potential attacks and cryptanalysis. When performing re-key, the controller generates a new Random number (RAND), and a new crypto session ID (CS_ID), and subsequently sends this information through the unicast DTLS secure channel established with each member. The new TEK and TAK are then generated by each member based on the algorithm described in Section 4.2, using the new RAND and CS_ID received from the controller. The TGK which serves as the 'master' group key does not change. When the TEK and TAK have been updated, the epoch number maintained in the multicast crypto session must be incremented.

7. IANA Considerations

tbd

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

8. Security Considerations

tbd

9. Acknowledgements

The authors greatly acknowledge discussion, comments and feedback from Dee Denteneer, Peter van der Stok and Zach Shelby. We also appreciate prototyping and implementation efforts by Pedro Moreno Sanchez who works as an intern at Philips Research.

10. References

10.1. Normative References

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

[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.

[RFC3740] Hardjono, T. and B. Weis, "The Multicast Group Security Architecture", RFC 3740, March 2004.

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

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

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

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

10.2. Informative References

[I-D.cheshire-dnsext-dns-sd]
           Cheshire, S. and M. Krochmal, "DNS-Based Service

Discovery", draft-cheshire-dnsext-dns-sd-11 (work in progress), December 2011.

[I-D.dijk-core-groupcomm-misc]
Dijk, E. and A. Rahman, "Miscellaneous CoAP Group Communication Topics", draft-dijk-core-groupcomm-misc-01 (work in progress), July 2012.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-12 (work in progress), October 2012.

[I-D.ietf-core-groupcomm]
Rahman, A. and E. Dijk, "Group Communication for CoAP", draft-ietf-core-groupcomm-02 (work in progress), July 2012.

[I-D.ietf-tls-oob-pubkey]
Wouters, P., Gilmore, J., Weiler, S., Kivinen, T., and H. Tschofenig, "Out-of-Band Public Key Validation for Transport Layer Security", draft-ietf-tls-oob-pubkey-04 (work in progress), July 2012.

[I-D.shelby-core-resource-directory]
Shelby, Z., Krco, S., and C. Bormann, "CoRE Resource Directory", draft-shelby-core-resource-directory-04 (work in progress), July 2012.

[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.

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6

Datagrams over IEEE 802.15.4-Based Networks", RFC 6282,
September 2011.

Authors' Addresses

Sye Loong Keoh
Philips Research
High Tech Campus 34
Eindhoven 5656 AE
NL

Email: sye.loong.keoh@philips.com

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

Email: oscar.garcia@philips.com

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

Email: sandeep.kumar@philips.com

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

Email: esko.dijk@philips.com