DICE Working Group
Internet-Draft
Intended status: Informational
Expires: July 21, 2014

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

January 17, 2014

DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)
draft-keoh-dice-multicast-security-02

Abstract

The CoAP and 6LoWPAN standards are fast emerging as the de-facto protocols in the area of resource-constrained devices. Such IP-based systems are foreseen to be used for building and lighting control systems where wireless devices interconnect with each other, forming low-power and lossy networks (LLNs). Both multicast and its 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 present for unicast in these CoAP devices. This draft deals with the adaptation of the DTLS record layer to protect multicast group communication, assuming that all group members already have the group security association parameters in their possession. The adapted DTLS record layer provides message confidentiality, integrity and replay protection to group messages using the group keying material before sending the message via IPv6 multicast to the group.

Status of This Memo

This Internet-Draft will expire on July 21, 2014.

Copyright Notice

Table of Contents

1. Introduction

There is an increased use of wireless control networks in environmental monitoring, industrial automation, lighting controls 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 the Internet-of-Things (IoT). These connected devices are equipped with communication capability that enables them to interact with each other as well as with the wider Internet services. However, the devices in such wireless control networks are characterized by power constraints (as these are usually battery-operated), have limited computational resources (low CPU clock, small RAM and flash storage) and often, the communication bandwidth is limited and unreliable (e.g., IEEE 802.15.4 radio). 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. It is more effective in LLNs to convey messages to a group of devices without requiring the sender to perform multiple time and energy consuming unicast transmissions to reach each individual group member. For example, in a building and lighting control system, the heating, ventilation, air-conditioning and lighting devices are often grouped according to the layout of the building, and control commands are issued simultaneously to a group of devices. Group communication for LLNs is based on the Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] sent over IP-multicast [I-D.ietf-core-groupcomm].

Currently, CoAP messages are 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. 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 [RFC3830] [RFC4082] [RFC3740] [RFC4046] [RFC4535], 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. [RFC3740] and [RFC4046] provide reference architectures for multicast security. [RFC4535] describes Group Secure Association Key Management Protocol (GSAKMP), a security framework for creating and managing cryptographic groups on a network which can be reused for key management in our context with any needed adaptation for LLNs.

This draft describes an approach to use DTLS as mandated in CoAP unicast to also support multicast security. We will assume that all devices in the group already have a group security association parameters based on a key management mechanism which is outside the scope of this draft. This draft specification only focuses on the adaptation of DTLS record layer to protect multicast messages to be sent to the group, and thus providing confidentiality, integrity and replay protection to the CoAP group messages.

1.1. Terminology

This specification uses the following terminology:

- Group Controller: The entity that is responsible for creating a multicast group and establishing security associations among authorized group members. It is also responsible for renewing/ updating the multicast group keys.

- Sender: The Sender is an entity that sends data to the multicast group. In a 1-to-N multicast group only a single sender is authorized to transmit data to the group. In an M-to-N multicast group (where M and N are not necessarily the same value), M group members are authorized to be senders.

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

- Security Association (SA): A set of policy and cryptographic keys that provide security services to network traffic that matches that policy [RFC3740]. A Security Association usually contains the following attributes:

  * selectors, such as source and destination transport addresses.

  * properties, such as identities.

  * cryptographic policy, such as the algorithms, modes, key lifetimes, and key lengths used for authentication or confidentiality.

  * keying material for authentication, encryption and signing.

- Group Security Association: A bundling of security associations (SAs) that together define how a group communicates securely. [RFC3740]

- o Keying material: Data that is specified as part of the SA which is needed to establish and maintain a cryptographic security association, such as keys, key pairs, and IVs [RFC4949].

1.2. Outline

This draft is structured as follows: Section 2 motivates the proposed solution with group communication use cases in LLNs and derives a set of requirements. Section 3 provides an overview of the proposed DTLS-based multicast security assuming that all devices in the group already have a group security association parameters in their possession. In Section 4, we describe the details of the adaptation of DTLS record layer for confidentiality and integrity protection of the multicast messages. Section 5 presents the security considerations.

2. Use Cases and Requirements

This section defines the use cases for group communication in LLNs and specifies a set of security requirements for these use cases.

2.1. Group Communication Use Cases

The "Group Communication for CoAP" draft [I-D.ietf-core-groupcomm] provides the necessary background for multicast based CoAP communication in LLNs and the interested reader is encouraged to first read this document to understand the non-security related details. This document also lists a few multicast group communication uses cases with detailed descriptions and some are listed here briefly:

a. Lighting control: enabling synchronous operation of a group of 6LoWPAN [RFC4944] [RFC6282] 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 control system are updated simultaneously, avoiding an excessive load on the LLN due to unicast firmware updates.

c. Parameter update: settings of a group of similar 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.

Elaborating on one of the main use cases, Lighting control, consider a building equipped with 6LoWPAN IP-connected lighting devices, switches, and 6LoWPAN border routers; the devices are organized in groups according to their physcial location in the building, e.g., lighting devices and switches in a room/floor can be configured as a single multicast group. The switches are then used to control the lighting devices in the group by sending on/off/dimming commands to all lighting devices in 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 physical subnets in the entire building.

2.2. Security Requirements

The "Miscellaneous CoAP Group Communication Topics" draft [I-D.dijk-core-groupcomm-misc] already defines 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:

a. Multicast communication topology: We consider both 1-to-N (one sender with multiple listeners) and M-to-N (multiple senders with multiple listeners) communication topologies. The 1-to-N communication topology is the simplest group communication scenario that would serve the needs of a typical LLN. For example, in the simple lighting control use case, the switch is the only entity that is responsible for sending control commands to a group of lighting devices. In more advanced lighting control use cases, a N-to-M communication topology would be required, for example if multiple sensors (presence or day-light) are responsible to trigger events to a group of lighting devices.

b. Multicast group size: The security solutions should support the typical group sizes that "Group Communication for CoAP" draft [I-D.ietf-core-groupcomm] intends to support. Group size is the combination of the number of Senders and Listeners in a group with possible overlap (a Sender can also be a Listener but need not be always). In typical LLN usecases, the number of Senders (normally the controlling devices) is much smaller than the number of Listeners (the controlled devices). A security solution that supports 1 to 255 Senders would cover the group sizes required for most use cases that are relevant for this document. The number of Listeners can be larger in the range of 2 to 5,000 devices.

c. Establishment of a Group Security Association (GSA): A secure mechanism must be used to distribute keying materials, multicast

security policies and security parameters to members of a multicast group. A GSA must be established by the group controller (which manages the multicast group) among the group members. The 6LoWPAN border router, a device in the 6LoWPAN, or a remote server outside the 6LoWPAN could play the role of the group controller. However, GSA establishment 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 include this feature preferably based on [RFC3740], [RFC4046] and [RFC4535].

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

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

f. Multicast data group authentication and integrity: It is essential to ensure that a multicast message originated from a member of the group and that messages have not been tampered with by attackers who are not members. 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 all other group members are trusted not to tamper with the multicast message.

g. Multicast data security ciphersuite: All group members must use the same ciphersuite to protect the authenticity, integrity and confidentiality of multicast messages. The ciphersuite is part of the GSA. Typically authenticity is more important than confidentiality in LLNs. Therefore the proposed multicast data security protocol must support atleast ciphersuites with MAC only (NULL encryption) and AEAD [RFC5116] ciphersuites. Other ciphersuites that are defined for data record security in DTLS should also be preferably supported.

h. Multicast data source authentication: Source authenticity is required if the group members are assumed to be untrusted and can tamper with the multicast messages. Source authenticity is not a critical feature to be always enabled in every LLN use case. If source authenticity is required for a specific use case, then it can be typically provided using public-key cryptography in which every multicast message is additionally signed by each 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 in every use case, source authenticity is not performed by default as part of the proposed data security protocol but can be added to it using additional mechanisms which are specified in this draft. It is important to note that for use cases demanding source authenticity, additional security mechanism is needed to provide such guarantee.

i. Forward security: Devices that leave the group should not have access to any future GSAs. This ensures that a past member device cannot continue to decrypt confidential data that is sent in the group. It also ensures that this device cannot send encrypted and/or integrity protected data after it leaves the group. The GSA update mechanism has to be defined as part of the key management scheme.

j. Backward confidentiality: A new device joining the group should not have access to any old GSAs. This ensures that a new member device cannot decrypt data sent before it joins the group. The key management scheme should ensure that the GSA is updated to ensure backward confidentality.

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 with minimal adaptation. 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

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. Figure 1 depicts a 1-to-N multicast communication and the roles of the nodes. 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, when new listeners join 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.

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

Figure 1: The roles of nodes in a 1-to-N multicast communication topology

3.2. Securing Multicast in LLNs

A group controller in an LLN creates a multicast group. The group 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 group 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 group controller communicates with individual device to add them to the new group. Additionally it distributes the Group Security Association (GSA) consisting of keying material, security policies security

parameters and ciphersuites using a standardized key management for LLN which is out of scope of this draft. 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 examples of ciphersuites (based on the security requirements) 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 [FIPS.197.2001], key length is 128-bit, and the authentication function is CCM [RFC6655] with a Message Authentication Code (MAC) length of 8 octets. Similar to [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 [FIPS.180-2.2002]. Depending on the future needs, other ciphersuites with different cipher algorithms and MAC length may be supported.

Senders in the group can encrypt and authenticate the CoAP group messages from the application using the keying material into the DTLS record. The authenticated encrypted message is passed down to the lower layer of the IPv6 protocol stack for transmission to the multicast address as depicted in Figure 2. The listeners when receiving the message, use the multicast IPv6 destination address and port (i.e., Multicast identifier) to look up the GSA needed for that group connection. The received message is then decrypted and the authenticity is verified using the keying material for that connection.

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

Figure 2: Sending a multicast message protected using DTLS Record Layer

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 group controller, and all member devices in the group have the GSA.

4.1. SecurityParameter derivation

The GSA is used to derive the same "SecurityParameters" structure as defined in [RFC5246] for all devices.

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 keying materials:

```
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 also set to the same value for all devices as part of the GSA 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 GSA could directly include the above six keying material when being configured in all group devices.

The current read and write states are instantiated for all group members based on the keying material and according to their roles: 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.

4.2. Record layer adaptation

In this section, we describe in detail the adaptation of the DTLS Record layer to enable multiple senders in the group to securely send information using a common group key, while preserving the confidentiality, integrity and freshness of the messages.

The following Figure 3 illustrates the structure of the DTLS record layer header, the epoch and seq_number are used to ensure message freshness and to detect message replays.

```
+---------+---------+--------+--------+--------+------------+-------+
| 1 Byte  | 2 Byte  | 2 Byte | 6 Byte | 2 Byte |            |       |
+---------+---------+--------+--------+--------+------------+-------+
| Content | Version | epoch  |  seq_  | Length | Ciphertext |  MAC  |
|   Type  | Ma | Mi |        | number |        |   (Enc)    | (Enc) |
+---------+---------+--------+--------+--------+------------+-------+
```

Figure 3: The DTLS record layer header and optionally encrypted payload and MAC

The epoch is fixed by the DTLS handshake and the seq_number is initialized to 0. The seq_number is increased by one whenever a sender sends a new record message. This is the mechanism of DTLS to detect message replay. Finally, the message is protected (encrypted and authenticated with a MAC) using the session keys in the "server write" parameters.

One of the problems with supporting multiple senders is that, the seq_number used by senders need to be syncronized to avoid their reuse, otherwise packets sent by different senders may get discarded as replayed packets. Further, the bigger problem is using a single key in a multiple sender scenario leads to nonce reuse in AEAD cipher suites like AES-CCM [RFC6655] and AES-GCM [RFC5288] as defined in DTLS. 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 the 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 of seperating the sequence number spaces by embedding a unique sender identifier in the sequence number as suggested in [RFC5288].

Thus in addition to configuring each device in the group with the GSA, the controller needs to assign a unique SenderID to each device which has the sender role in the group. The size of the SenderID is 1-octet based on the requirement for the supported group size mentioned in Section 2.2. The list of SenderIDs are then distributed to all the group members by the controller.

The existing DTLS record layer header is adapted such that the 6-octet seq_number field is split into a 1-octet SenderID field and a 5-octet "truncated" trunc_seq_number field. Figure 4 illustrates the adapted DTLS record layer header.

```
         +---------+---------+--------+--------+-----------+--------+
         | 1 Byte  | 2 Byte  | 2 Byte | 1 Byte | 5 Byte    | 2 Byte |
         +---------+---------+--------+--------+-----------+--------+
         | Content | Version | Epoch  | Sender | trunc_seq_| Length |
         |   Type  | Ma | Mi |        |   ID   | number    |        |
         +---------+---------+--------+--------+-----------+--------+
```

Figure 4: The adapted DTLS record layer header

4.3. Sending Secure Multicast Messages

Senders in the multicast group when sending a CoAP group message from the application, create the adapted DTLS record payload based on the "server write" parameters. Each sender in the group uses its own unique SenderID in the DTLS record layer header. It also manages its own epoch and trunc_seq_number in the "server write" connection state; the first record sent has the trunc_seq_number 0. After creating the DTLS record, the trun_seq_number is incremented in the "server write" connection state. The adapted DTLS record is then passed down to UDP and IPv6 layer for transmission on the multicast IPv6 destination address and port.

4.4. Receiving Secure Multicast Messages

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 and port 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 and port.

Listener devices in a multiple 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 trunc_seq_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 IPv6 destination address and port 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 trunc_seq_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 trunc_seq_number in the message received is higher than the stored

value, otherwise the message is discarded. Alternatively a windowing mechanism can be used to accept genuine out-of-order packets. 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 trunc_seq_number in the corresponding "client read" connection state are updated as well.

5. Security Considerations

Some of the security issues that should be taken into consideration are discussed below.

5.1. Late joiners

Listeners who are late joiners to a multicast group, do not know the current epoch and trun_seq_number being used by different senders. When they receive a packet from a sender with a random trunc_seq_number 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 to AERO [I-D.mcgrew-aero] where the late joining listener on receiving the first packet from a particular sender, initialize its last seen epoch and trunc_seq_number in the "client read" state for that sender, however does not pass this packet to the application layer and instead drops it. This provides a reference point to identify if future packets are fresher than the last seen packet. Alternatively, the group controller which can act as a listener in the multicast group can maintain the epoch and trunc_seq_number of each sender. When late joiners send a request to the group controller to join the multicast group, the group controller can send the list of epoch and trunc_seq_numbers as part of the GSA.

5.2. Uniqueness of SenderIDs

It is important 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 become aware of a clash. 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.

5.3. Reduced sequence number space

The DTLS record layer seq_number is truncated from 6 octets to 5 octets. This reduction of the seq_number space should be taken into account to ensure that epoch is incremented before the trunc_seq_number wraps over. The sender or the controller can increase the epoch number by sending a ChangeCipherSpec message whenever the trunc_seq_number has been exhausted. This should be done as part of the key management mechanism which is not defined in this draft.

6. Acknowledgements

The authors greatly acknowledge discussion, comments and feedback from Dee Denteneer, Peter van der Stok, Zach Shelby and Michael StJohns. 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.

7. References

7.1. Normative References

[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-18 (work in progress), December 2013.

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

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

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

7.2. Informative References

[FIPS.180-2.2002]
National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-2, August 2002, <http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf>.

[FIPS.197.2001]
National Institute of Standards and Technology, "Advanced Encryption Standard (AES)", FIPS PUB 197, November 2001, <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.

[I-D.dijk-core-groupcomm-misc]
Dijk, E. and A. Rahman, "Miscellaneous CoAP Group Communication Topics", draft-dijk-core-groupcomm-misc-05 (work in progress), December 2013.

[I-D.ietf-core-resource-directory]
Shelby, Z., Bormann, C., and S. Krco, "CoRE Resource Directory", draft-ietf-core-resource-directory-01 (work in progress), December 2013.

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

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

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

[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.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[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.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

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

[RFC6282] Hui, J. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, September 2011.

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

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 (editor)
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