DTLS Relay for Constrained Environments

draft-kumar-dice-dtls-relay

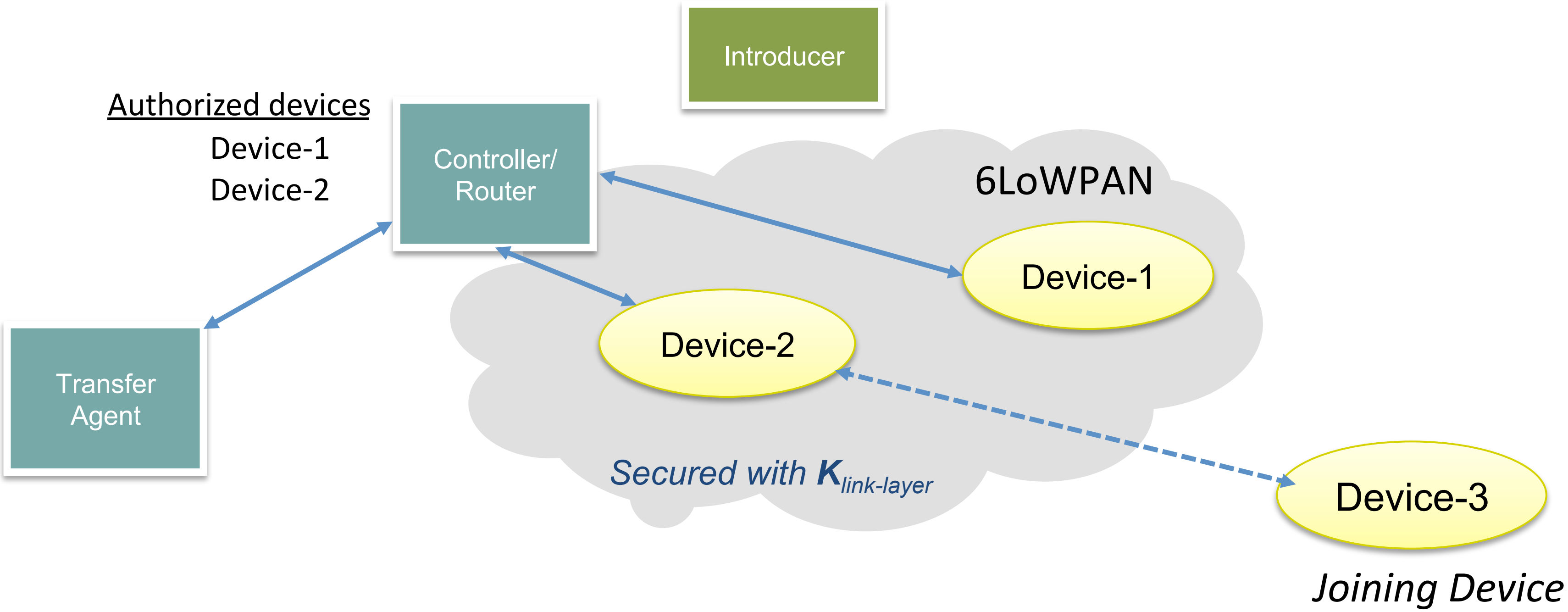
Sandeep S. Kumar, *Sye Loong Keoh, Oscar Garcia-Morchon*

IETF88 Nov 4, 2013, Berlin

Email: sandeep.kumar AT philips.com

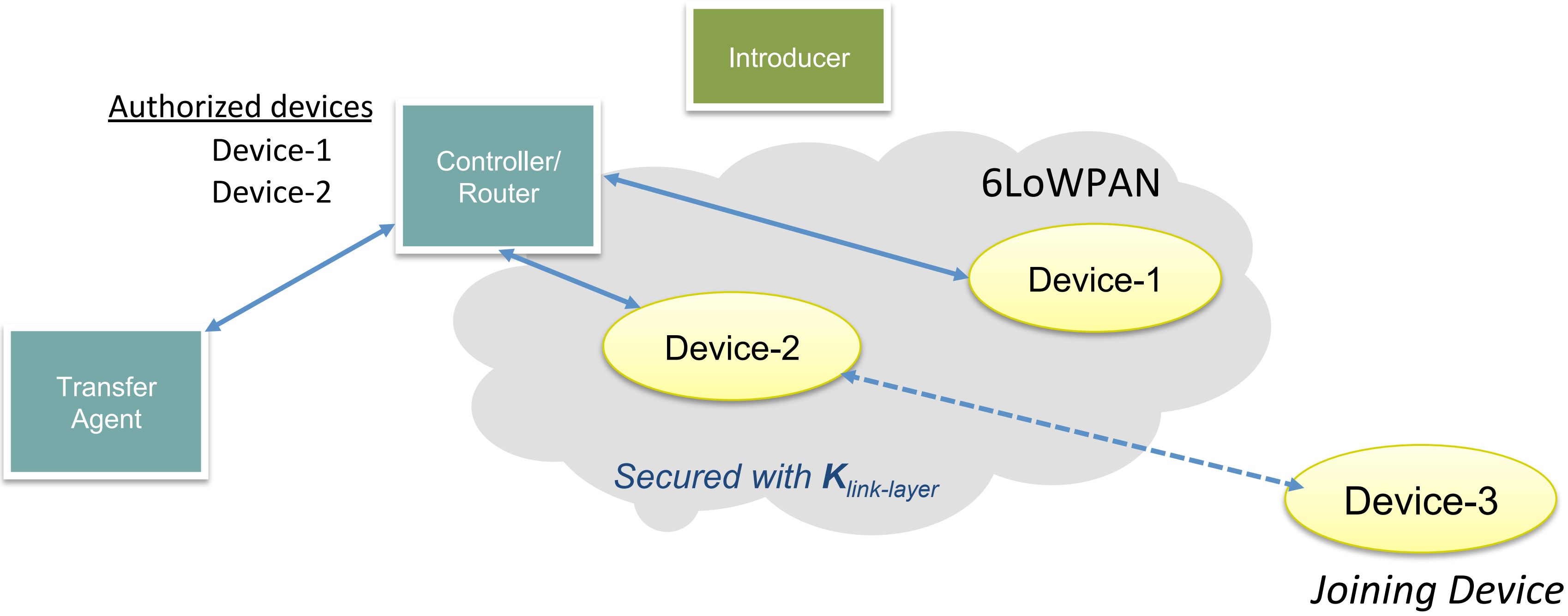
Use case: Secure Network Access

- **draft-jennings-core-transitive-trust-enrollment**
- *Joining Devices* only have initial trust (keying material) with a *Trust Agent*



- *Device-3* is multi-hop away from the controller.
  - Can only communicate with link neighbors
  - No IP routing yet for point-to-point connection with *Transfer Agent*
- Similar issues with raw-public key based pairing of controller and devices multi-hop away

Motivation



- DTLS is chosen security solution for CoAP: beneficial for constrained devices if it can be also re-used for the “joining device” authentication
- Perform DTLS authentication handshake from a non-IP routable *Joining Device* to *Transfer Agent* (or *Controller).*
- Once authenticated, keys or key management messages can pass through the secure session tunnel (**out-of-scope** for now and should be addressed in a broader context)

Proposed solution

- An already authenticated device in the network acts as a **relay** to forward messages between the *Joining Device* and the *Trust Agent/Controller*.

- Two cases:
  - Address is known by the *Joining Device.* E.g. during manufacturing
  - Address is not known by the *Joining Device* but known to the *relay.* E.g. in raw public-key case
- Relay can either maintain state or can be stateless

Stateful Relay with unknown address

IP_C:p_C = IP (non-routable) and port of Client
IP_S:5684 = IP and coaps port of end Server
IP_Ra:5684 = IP (link-local) and coaps port of Relay
IP_Rb:p_Rb = IP and the port of Relay

Relay stores table
{IP_C, p_C, p_Rb}

```
+---------------+---------------+----------------+----------------------------+
|  DTLS Client  |   DTLS Relay  |   DTLS Server  |          Message           |
|     (C)       |       (R)     |       (S)      | Src_IP:port | Dst_IP:port  |
+---------------+---------------+----------------+-------------+--------------+
|      --ClientHello-->                          |   IP_C:p_C  | IP_Ra:5684   |
|                          --ClientHello-->      |   IP_Rb:p_Rb| IP_S:5684    |
|                                                |             |              |
|                          <--ServerHello--      |   IP_S:5684 | IP_Rb:p_Rb   |
|                                  :             |             |              |
|            <--ServerHello--                    |   IP_Ra:5684| IP_C:p_C     |
|                    :                           |             |              |
|                        ::                      |       :     |    :         |
|                        ::                      |       :     |    :         |
|        --Finished-->                           |   IP_C:p_C  | IP_Ra:5684   |
|                            --Finished-->       |   IP_Rb:p_Rb| IP_S:5684    |
|                                                |             |              |
|                            <--Finished--       |   IP_S:5684 | IP_Rb:p_Rb   |
|        <--Finished--                           |   IP_Ra:5684| IP_C:p_C     |
|                       ::                       |      :      |     :        |
+------------------------------------------------+-------------+--------------+
```

Stateless Relay with unknown address

IP_C:p_C   = IP (non-routable) and port of Client
IP_S:5684   = IP and coaps port of Server
IP_Ra:5684 = IP (link-local) and coaps port of Relay
IP_Rb:p_Rb = IP and the port of Relay

DRY(Header,Content) – DTLS RelaY message for encapsulation

```
+----------------+---------------------+---------------------+---------------------------+
|  DTLS Client   |     DTLS Relay      |     DTLS Server     |          Message          |
|     (C)        |         (R)         |         (S)         | Src_IP:port | Dst_IP:port |
+----------------+---------------------+---------------------+-------------+-------------+
|         --ClientHello-->                                   |   IP_C:p_C  | IP_Ra:5684  |
|                      --DRY[H(IP_C:p_C),C(ClientHello)]-->  |   IP_Rb:p_Rb| IP_S:5684   |
|                                                            |             |             |
|                      <--DRY[H(IP_C:p_C),C(ServerHello)]--  |   IP_S:5684 | IP_Rb:p_Rb  |
|                                           :                |             |             |
|         <--ServerHello--                                   |   IP_Ra:5684| IP_C:p_C    |
|                 :                                          |             |             |
|                              ::                            |      :      |    :        |
|                              ::                            |      :      |    :        |
|          --Finished-->                                     |   IP_C:p_C  | IP_Ra:5684  |
|                         --DRY[H(IP_C:p_C),C(Finished)]-->  |   IP_Rb:p_Rb| IP_S:5684   |
|                                                            |             |             |
|                         <--DRY[H(IP_C:p_C),C(Finished)]--  |   IP_S:5684 | IP_Rb:p_Rb  |
|          <--Finished--                                     |   IP_Ra:5684| IP_C:p_C    |
|                              ::                            |     :       |     :       |
+------------------------------------------------------------+-------------+-------------+
```

Other issues

- Need to prevent Denial-of-Service attacks from malicious unauthenticated nodes
  - Policies in the Relay to (dis)allow relaying of such messages
  - Policies can be sent by the controller to all devices
- Should DRY be DTLS message or in another layer?
  - Should the DRY headers be secured?

Summary

- DTLS Relay mechanism in nodes to enable *end-to-end* DTLS session for *Joining nodes*
- Enables re-use of existing security protocols on constrained devices in LLNs to also enable network access.
- Further define security mitigation for DoS and DRY headers

Questions?