Network Working Group   M. Richardson
Internet-Draft   SSW
Intended status: Informational   October 26, 2014
Expires: April 29, 2015

6tisch secure join using 6top
draft-richardson-6tisch--security-6top-03

Abstract

This document details a security architecture that permits a new 6tisch compliant node to join an 802.15.4e network. The process bootstraps the new node authenticating the node to the network, and the network to the node, and configuring the new node with the required 6tisch schedule. Any resemblance to WirelessHART/IEC62591 is entirely intentional.

Requirements Language

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A challenging part with constructing an LLN with nodes from multiple vendors is providing enough security context to each node such that the network communication can form and remain secure. Most LLNs are small and have no operator interfaces at all, and even if they have debug interfaces (such as JTAG) with personnel trained to use that, doing any kind of interaction involving electrical connections in a dirty environment such as a factory or refinery is hopeless.

It is necessary to have a way to introduce new nodes into a 6tisch network that does not involve any direct manipulation of the nodes themselves. This act has been called "zero-touch" provisioning, and it does not occur by chance, but requires coordination between the manufacturer of the node, the service operator running the LLN, and the installers actually taking the devices out of the shipping boxes.

1.1. Assumptions

For the process described in this document to work, some assumptions about available infrastructure are made. These are perhaps more than assumptions, but rather architectural requirements; the exact operation of said infrastructure to be defined in a subsequent document.

In the diagrams and text that follows entities are named (and defined in the terminology section). Unless otherwise stated these are roles, not actual machines/systems. The roles are seperated by network protocols in order that they roles can be performed by different systems, not because they have to be. Different deployments will have different scaling requirements for those entities. Smaller deployments might co-located many roles together into a single ruggedized platform, while other deployments might operate all of the roles on distinct, multiply-redundant server classes located in a fully equipped datacentre.

2. Terminology and Roles

Most terminology should be taken from [I-D.ietf-6tisch-architecture] and from [I-D.ietf-6tisch-6top-interface] and [I-D.wang-6tisch-6top-sublayer]. As well, many terms are taken from [RFC6775].

The following roles/things are defined:

PCE
: the Path Computation Engine. This entity reaches out to each of the nodes in the LLN, and configures an appropriate schedule using 6top.

Authz Server/ACE
: the Authorization Server. This offloads calculation of access control lists and other access control decisions for constrainted nodes. See [I-D.seitz-ace-problem-description]

6top
: the 6top protocol is defined abstractly in [I-D.ietf-6tisch-6top-interface] and mapped to run over CoAP in [I-D.ietf-6tisch-coap]. The 6top protocol is defined primarily to provision the 6TiSCH schedule; this document proposes to extend it for also provisioning of layer-2 security parameters.

JCE
: the Join Coordination Entity. This acronym is chosen to parallel the PCE.

joining node
: The newly unboxed constrained node that needs to join a network.

join protocol
: the protocol which secures initial communication between the joining node and the JCE

join assistant
: A constrained node near the joining node that will act as it’s first 6LR, and will relay traffic to/from the joining node.

join network
: A 802.15.4e network whose encryption and authentication key is "JOIN6TISCH".

unique join key
: a key shared between a newly joining node, and the JCE

production network
: A 802.15.4e network whose encryption/ authentication keys are determined by some algorithm. There may have network-wide group keys, or per-link keys.

production network key
: A shared L2-key known by all authorized nodes. This key can be used to derive other keys.

per-peer L2 key
: a key that results from an exchange (such as MLE) that creates a pair-wise L2 key which is known only to the two nodes involved, [I-D.piro-6tisch-security-issues] calls this a LinkKey

The following terms are used in this document and come from other documents:

DevID
: [IEEE.802.1AR] defines the secure DEVice IDentifier as a device identifier that is cryptographically bound to the device and is composed of the Secure Device Identifier Secret and the Secure Device Identifier Credential.

IDevID
: The Initial secure DEVice IDentifier (IDevID) is the Device Identifier which was installed on the device by the manufacturer.

LDevID
: A Locally significant secure DEVice IDentifiers (LDevIDs) is a Secure Device Identifier credential that is unique in the local administrative domain in which the device is used. The LDevID is usually a new certificate provisioned by some local means, such as the 6top mechanism defined in this document.

CoAP
: The CoAP protocol, defined in [RFC7252] is an HTTP-like resource access protocol. CoAP runs over UDP.

DTLS
: The datagram version of TLS, defined in [RFC6347], and which can be used to secure CoAP in the same way that TLS secures HTTP.

ARO
: [RFC6775]defines a number of new Neighbor Discovery options including the Address Registration Option

DAR/DAC
: [RFC6775]defines the Duplicate Address Request and Duplicate Address Confirmation options to turn the multicasted Duplicate Address Detection protocol into a client/server process

EARO
: [I-D.thubert-6lo-rfc6775-update-reqs]extends the ARO option to include some additional fields necessary to distinguish duplicate addresses from nodes that have moved networks when there are mulitple LLNs linked over a backbone.

3. Architectural requirements of join protocol

This section works from the ultimate goal, and goes backwards to prerequisite actions. Section 6 presents the protocol from beginning to end order.

The ultimate goal of the join protocol is to provide a new node with enough locally significant security credentials that it is able to take part in the network directly. The credentials may vary by deployment. They can be one of:

1) a network-wide shared symmetric key (this is the production network key, or MasterKey)

2) a locally significant (one-level only) 802.11AR type DevID certificate (which allows it to negotiate a pair-wise key)

One of these items is communicated by the JCE to the joining node using the 6top protocol. The authentication of this communication channel is the subject of the Join Protocol as explained below.

Given one of the the above, there are a number of possible protocols that can be used to generate layer-2 sessions keys for the node, including:

1) Mesh Link Exchange [I-D.kelsey-intarea-mesh-link-establishment] (IMPORTANT, a good option. Uses certificates from common CA)

2) work in 802.15.9 (uses certificates from common CA)

3) Security Framework and Key Management Protocol Requirements for 6TiSCH [I-D.ohba-6tisch-security] (this document provides the phase 0 required, using the network-wide shared key)

4) Layer-2 security aspects for the IEEE 802.15.4e MAC [I-D.piro-6tisch-security-issues]: the MasterKey is used to derive per-peer L2 keys

Per-peer L2 keying is critical when doing peer2peer schedule negotiation over 15.4 Information Elements. Therefore a network-wide layer-2 key is inappropriate for the self-organizing networks, and a protocol (MLE, 802.15.9) SHOULD be used to derive per-peer L2 keys.

For networks where there is a PCE present and will do all schedule computation, then the only trust relationship necessary is between the individual node and the PCE, and it MAY be acceptable to have a network-wide L2 key derived in ways such as [I-D.piro-6tisch-security-issues] describes in section ?

The intermediate goal of the join protocol is to enable a Join Coordination Entity (JCE) to reach out to the new node, and install the credentials detailed above. The JCE must authenticate itself to the joining node so that the joining node will know that it has joined the correct network, and the joining node must authenticate itself to the JCE so that the JCE will know that this node belongs in the network. This two way authentication occurs in the 6top/CoAP/DTLS session that is established between the JCE and the joining node.

[I-D.ietf-6tisch-6top-interface] presents a way to interface to a 6top information model (defined in YANG). [I-D.ietf-6tisch-coap] explains how to access that information model using CoAP. That model is to be extended to include security attributes for the network. The JCE would therefore reach out to the joining node and simply provision appropriate security properties into the joining node, much like the PCE will provision schedules.

This 6top-based secure join protocol has defined a push model for security provisioning by the JCE. This has been done for three reasons:

1) 6tisch nodes already have to have a 6top CoAP server for schedule provising

2) this permits the JCE to manage how many nodes are trying to join at the same time, and limit how much bandwidth/energy is used for the join operation, and also for the JCE to prioritize the join order for nodes.

3) making the JCE initiate the DTLS connection significantly simplies the certificate chains that must be exchanged as the most constrained side (the joining node) provides it’s

credentials first, and lets the less constrained JCE figure out what kind of certificate chain will be required to authenticate the JCE to the joining node. In EAP-TLS/802.1x situations, the TLS channel is created in the opposite direction, and it would have to complete in a tentative way, and then further authorization occur in-band.

In order for a 6top/DTLS/CoAP connection to occur between the JCE and the joining node, there needs to be end-to-end IPv6 connectivity between those two entities. The joining node will not participate in the route-over RPL mesh, but rather will be seen by the network as being a 6lowpan only leaf-node.

There are some alternatives to having full end to end connectivity which are discussed in the security considerations section.

The specific mechanism to enable end to end connectivity with the JCE are still open but will consist of one of:

(1) IPIP tunnel between Join Assistant and JCE (least preferred)

(2) using straight RPL routing: the Join Assistant sends a DAO (moderate preference)

(3) using a separate RPL DODAG for join traffic (could be a non-storing, best practice)

(4) establishing a specific multi-hop 6tisch track for join traffic for each Join Assistant (not always practical)

Of these mechanisms, the only one which does not require additional state on the Join Assistant (which is also a constrained device) is (1) and (2). Mechanism (2) additionally requires no specific state on the Join Assistant. Mechanism (2), in a non-storing DODAG requires additional state on the DODAG root (6LBR) only; while mechanism (1) requires a similar amount of state on the JCE. For deployments where the JCE is part of the 6LBR, the amount of state is similar, but in any case, the 6LBR is assumed to be a non-constrained node.

As long as the Join Assistant does not do any kind of stateful firewalling, the IPIP tunnel and the DAO (2) method can be done by the Join Assistant statelessly. Upward traffic from the Join Network must be restricted to a 6tisch slotframe(s) to which join traffic is welcome, no tunnelling is necessary as the upwards routes are all in place. A destination address ACL on traffic from the Join Network restricts the Joining Nodes to sending traffic only to the address of the JCE. (If JCE and 6LBR are colocated, then this is the address in the ABRO, if they are not colocated, then this address needs to have been provisioning in the Join Assistant when it joined, or could be carried in a new RA option)

When using option (2), networks that have storing mode DODAGs will consume routing resources on all intermediate nodes between the Join Assistant and the DODAG root. This resource will be depleted without any authentication, and this threat is detailed below.

Continuing to work backwards, in order the JCE reach out to provision the Joining Node, it needs to know that the new node is present. This is done by taking advantage of the 6lowPAN Address Resolution Option (ARO) (section 4.1 [RFC6775]). The ARO causes the new address to also be sent up to the 6LBR for duplicate detection using the DAR/DAC mechanism. The 6LBR simply needs to tell the JCE about this using a protocol that needs to be defined, but could be either DAR or NS.

In addition to needing to know the joining devices address from the DAR/NS, the JCE also needs to know the joining node' IDevID. If the serialNumber attribute of the IDevID is less than 64 bits, then it is possible that it could be placed into the EUI-64 option of the ARO, or the OUI of the [I-D.thubert-6lo-rfc6775-update-reqs] EARO. The JCE needs to know the joining node's serialNumber to know if this is device that it should even attempt to provision; and if so, it may need to retrieve an appropriate certificate chain (see [I-D.richardson-6tisch-idevid-cert]) from the Factory in order for the JCE to prove it is the legitimate owner of the joining node.

Neither 802.1AR nor [RFC5280] provide any structure for the serialNumber, except that they are positive integers of up-to 20 octets in size (numbers up to 2^160). This specification would require that the serialNumber encoded in the IDevID is the same as the EUI-64 used by the device. Some consideration needs to be given as to whether there are privacy considerations to doing this: any observer that can see the join traffic, can also see the source MAC address of the node as well.

Prior to being able to announce itself in a NS, the joining node needs to find the Join Network. This is done by listening to an extended beacon which are broadcast in designated slotframes by Join Assistants. The Extended Beacon provides a way for the Joining Node to synchronize itself to the overall timeslot schedule and provides an Aloha period in which the Joining Node can send a Router Soliticiation, and receive an appropriate Router Advertisement giving the Joining Node a prefix and default route to which to send join traffic.

It may be possible to eliminate a message exchange if space for a Router Advertisement can be found as part of the Join Network Extended Beacon. This Enhanced Beacon would be distinct to the Join Network, and would be encrypted with the well-known Join Network key.

3.1. prefixes to use for join traffic

What prefix would the joining node for communication? There are three options:

(1) just use link-local addresses (requires all traffic be tunneled)

(2) use a prefix specifically for join traffic (may be easier with a join-only DODAG)

(3) use the same prefix as the rest of the traffic (may require more complex ACLs, and leaks information to attackers)

4. security requirements

4.1. threat model

There are three kinds of threats that a join process must deal with: threats to the joining node, threats to the resources of the network, and threats to other joining nodes.

4.1.1. threats to the joining node

A node may be taken out of it’s box by a malicious entity and powered on. This could happen during shipping, while being stored in a warehouse. The device may be subject to physical theft, or the goal of the attacker may be to turn the device into a trojan horse of some kind. Physical protection of the device is out of scope for this document; this document will henceforth assume that the device is sealed in some tamper-evident way and this document deals with attacks over the network.

An attacker may attempt to convince the joining node that it is the legitimate Production Network; this is done by putting up a legitimate looking Join Network, and following the protocol as described in this document. The Joining Node can not know if it has the corrrect Production Network until steps 11-13, when it attempts to validate the ClientCertificate provided by the JCE.

When the joining node determines that this is the incorrect network, it must remember the PANID of the network that it has attempted to join, and then look for another network to try. It SHOULD have some limit as the number of times it will try before going back to sleep, or shutting down, and it SHOULD take care not to consume more than some specified percentage of any battery it might have.

Should a malicious production network be present at the same time/ place as the legitimate production network, a the malicious agent could intercept and replay various packets from the proper join network, but ultimately this either results in a jamming-like denial of service, and/or the the ClientCertificate will not validate.

It is a legitimate situation for there to be multiple possible join networks, and the joining node may have to try each one before it finds the network that it the right one for it. The incorrect, but non-malicious networks will not attempt the 6top provisioning step, and SHOULD return a negative result in steps 8/9, refusing the node's NS. Those incorrect networks will be recognize that the node does not belong to them, because they will be able to see the Joining Node's IDevID in the ARO of step 4.

4.1.2. threats to the resources of the network

The production network has two important resources that may be attacked by malicious Joining nodes: 1) energy/bandwidth, 2) memory for routing entries.

A malicious joining node could send many NS messages to the Join Assistant (from many made up addresses), which would send many NS/DAR messages to the 6LBR, and this would consume bandwidth, and therefore energy from the members of the mesh along the path to the 6LBR. This can be mitigated by limited the total bandwidth available for joining.

A malicious joining node could send many NS messages, and if the 6LBR agreed to accept the new node (by IDevID), then the Join Assistant would MAY inject routing information into mesh for the Joining node. Non-storing DODAGs store are routing information in the DODAG Root (probably the 6LBR), which is generally not a constrained node. Storing DODAGs store routing entries at all nodes up to the DODAG, and those are constrained nodes. Using a separate Join DODAG, and having that DODAG be non-storing will reduce any impact on intermediate nodes, but it does cause resources to be used for the second DODAG, and it may have a code impact if the nodes otherwise would not implement non-storing RPL.

4.1.3. threats to other joining nodes

A joining node (or the nodes of a malicious network, co-located near the legitimate production network) may mount attacks on legitimate nodes which have not yet joined.

The malicious nodes may attempt to perform 6top operations against the joining node to keep it from being able to respond to the legitimate 6top session from the legitimate JCE. During the Join phase, the Joining node MUST have all other resources and protocols turned off, even if they would normally be accessible as read-only unauthenticated CoAP resources.

Malicious nodes could use the Join Network to mount various DTLS based attacks against the joining node, such as sending very long certificate chains to validate. One might think to limit the length of such chains, but as shown in [I-D.richardson-6tisch-idevid-cert] the chain may as a long as the supplier chain, plus may include additional certificates due to resales of plants/equipment/etc. Validating from a trusted certificate down to the specific certificate which proves ownership would eliminate random certificate chains, but the attacker could just feed the joining node legitimate chains that it observed (and replayed) from the legitimate JCE. This does no good; the Joining node finds that the DTLS connection is invalid, but it may significantly run batteries down.

4.2. implementation cost

(storage of security material, computational cost)

4.3. denial of service

other communication impacts of security protocol mechanics

5. protocol requirements/constraints/assumptions

5.1. inline/offline

dependencies on centralized or external functionality, inline and offline

6. time sequence diagram

```
   +-----+   +------+            +----------+             +-----------+
   |     |   |      |            |  JOIN    |             |  Joining  |
   | JCE |   | 6LBR |            | Assistant|             |   Node    |
   +-----+   +------+            |  (proxy) |             |           |
      |         |                +----------+             +-----------+
      |         |                     |                            |
      |         |                     |-------BEACON (1)---------->|
      |         |                     |                            |
      |         |                     |<----Router Solicitation----|
      |         |                     |---Router Advertisement---->|
      |         |                     |                            |
      |         |                     |<------CERT CACHE-----------|
      |         |                     |         LOAD    (2)        |
      |         |                     |                            |
      |         |                     |                            |
      |         |                     |-------CERT CACHE---------->|
      |         |                     |        RESPONSE (multiple) |
      |         |                     |                 (packets)  |
      |         |                     |                            |
      |         |                     |<-----JOIN REQUEST (4) -----|
      |         |                     |      (NS w/ARO )           |
      |         |                     |                            |
      |         |<---NS (DAR) (5)-----|                            |
      |<--??(6)-|                     |                            |
      |         |                     |                            |
      |--??(7)->|                     |                            |
      |         |                     |                            |
      |         |----NS (DAC)-(8)---->|                            |
      |         |     +------+        |                            |
      |         |<DAO-| mesh |<--DAO--|                            |
      |         |-DAO-| node |--DACK->|                            |
      |         | ACK +------+        |                            |
      |         |                     |-------JOIN ACK (9)-------->|
      |         |                     |                            |
      |         |                     |                            |
      |================(10)==========>|----------6top----(11)----->|
      |         |                     |          DTLS              |
      |<===============(13)===========|<---------CoAP----(12)------|
      |         |                     |        (many packets)      |
```

Figure 1: Message sequence for JOIN message

6.1. explanation of each step

6.1.1. step (1): enhanced beacon

A 6tisch join/synchronization beacon is broadcast periodically, and is authenticated with a symmetric "beacon key":

well known JOIN key, such "JOIN6TISCH"

another key, provisioned in advance (OOB)

a shared symmetric key derived from public part of top level certificate (a closely held "secret")

The purpose of this key is not to provide a high level of assurance, but rather to filter out 6tisch traffic from another random traffic that may be sharing the same radio frequencies.

These beacons are used for JOIN purpose only, and are not related to the Enhanced Beacons used in the rest of 6tisch.

6.1.2. step (1B): send router solicitation

The joining node sends a router solicitation during the Aloha period of the beacon.

6.1.3. step (1C): receive router advertisement

The joining node receives a router advertisement from the Join Assistant. It could include 6CO options to help compress packets, and should contain a prefix appropriate for join traffic.

6.1.4. step (2): certificate cache load

At step 10, the JCE will need to present a certificate chain anchored at a trusted CA built into the joining node. It has been speculated that a significant amount of traffic could be avoided at step (10) if the common parts of the certificate chains could be cached in the join assistant.

This optional step involves the joining node asking for certificates from the join assistant.

6.1.5. step (3): receive certificate cache

the proxy neighbour sends requested cached certificates to the joining node

6.1.6. step (4): join request

A regular Neighbour Solicitation is sent. This should contain an ARO (or EARO) option containing the Joining Nodes' IDevID. The ARO/EARO will be proxied by the Join Assistant as part of normal 6LowPAN processing for leaf nodes (non-RPL nodes) upwards to the 6LBR

6.1.7. step (5): NS duplicate address request (DAR)

6.1.8. step (7): 6LBR informs JCE of new node

6.1.9. step (8): JCE informs/acks to 6LBR of new node

The JCE could reply in the negative, and this would cause a DAC failure, TBD

6.1.10. step (9): NS duplicate address confirmation (DAC)

6.1.11. step (10): JCE initiates connection to joining node

The double lines indicate that an IPIP tunnel operation may be required. If a straight DAO or seperate Join DODAG is used, then this is just a straight forwarding root to leaf node forwarding operation, and involves either using source routes (non-storing), or just forwarding for storing DODAGs.

A specific bandwidth allocation would be used for this join traffic

The production network encryption keys would be used for the join traffic

6.1.12. step (11): Join Assistant forwards packet to joining node

The JOIN Assistant would forward traffic to the Joining Node. Recognizing that this traffic the JOIN Network, the JOIN Assistant would use the JOIN Network key.

6.1.13. step (12): Joining node replies

The joining node replies, using JOIN Network key.

6.1.14. step (13): Join Assistant forwards reply to JCE

The JOIN Assistant, recognizing that the traffic came from the JOIN Network, restricts the destination that can be reached to the the JCE only. It can do this in a stateless way, and it does NOT need to track the traffic at (10) to open pinhole, etc.

Recognizing that the traffic came from the JOIN Network, the traffic would be placed into a bandwidth allocation (track?) that allows such traffic.

6.2. size of each packet

and number of frames needed to contain it.

7. resulting security properties obtained from this process

An end to end IPv6 CoAP/DTLS connection is created between the JCE and the Joining Node. This connection carries 6top commands to update security parameters. This results in either deployment of a single-level, locally relevant certificate (LDevID), or deployment of a network-wide symmetric "Master Key"

8. deployment scenarios underlying protocol requirements

9. device identification

The JCE authenticates the joining node using a certificate chain provided inline during the DTLS negotiation. The certificate chain is rooted in a vendor certificate that the JCE must have preloaded, and is a statement as to the node’s 802.1AR IDevID. The joining node authenticates the

9.1. PCE/Proxy vs Node identification

9.2. Time source authentication / time validation

Note: RPL Root authentication is a chartered item

9.3. description of certificate contents

9.4. privacy aspects

The EUI-64 of the Joining node is transmitted using a Well Known layer-2 encryption key. Within the ARO/EARO of the Neighbour Solicitation is an OUI, which may be identical to the EUI-64 of the Joining node, or it might be an unrelated IDevID.

An eavesdropper can therefore learn something about the manufacturer of every device as it joins.

10. slotframes to be used during join

how is this communicated in the (extended) beacon.

11. configuration aspects

(allocation of slotframes after join, network statistics, neighboetc.)

12. authorization aspects

lifecycle (key management, trust management)

12.1. how to determine a proxy/PCE from a end node

12.2. security considerations

what prevents a node from transmitting when it is not their turn (part one: jamming)

can a node successfully communicate with a peer at a time when not supposed to, may be tied to link layer security, or will it be policed by receiver?

13. security architecture

security architecture and fit of e.g. join protocol and provisioning into this

14. Posture Maintenance

(SACM related work)

15. Security Considerations

16. Other Related Protocols

17. IANA Considerations

18. Acknowledgements

19. References

19.1. Normative references

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

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

[IEEE.802.1AR] Institute of Electrical and Electronics Engineers, "Secure Device Identity", IEEE 802.1AR, 2009, <http://www.ieee802.org/1/pages/802.1ar.html>.

[I-D.ietf-6tisch-coap] Sudhaakar, R. and P. Zand, "6TiSCH Resource Management and Interaction using CoAP", draft-ietf-6tisch-coap-00 (work in progress), May 2014.

[I-D.ietf-6tisch-6top-interface] Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top) Interface", draft-ietf-6tisch-6top-interface-00 (work in progress), March 2014.

[I-D.ietf-6tisch-architecture] Thubert, P., Watteyne, T., and R. Assimiti, "An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4e", draft-ietf-6tisch-architecture-01 (work in progress), February 2014.

[I-D.irtf-nmrg-autonomic-network-definitions] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking - Definitions and Design Goals", draft-irtf-nmrg-autonomic-network-definitions-00 (work in progress), December 2013.

[I-D.seitz-ace-problem-description] Seitz, L. and G. Selander, "Problem Description for Authorization in Constrained Environments", draft-seitz-ace-problem-description-00 (work in progress), May 2014.

[I-D.richardson-6tisch-idevid-cert] Richardson, M., "X509.v3 certificate extension for authorization of device ownership", draft-richardson-6tisch-idevid-cert-00 (work in progress), May 2014.

[I-D.wang-6tisch-6top-sublayer]
Wang, Q., Vilajosana, X., and T. Watteyne, "6TiSCH Operation Sublayer (6top)", draft-wang-6tisch-6top-sublayer-00 (work in progress), February 2014.

[I-D.thubert-6lo-rfc6775-update-reqs]
Thubert, P., "Requirements for an update to 6LoWPAN ND", draft-thubert-6lo-rfc6775-update-reqs-04 (work in progress), August 2014.

19.2. Informative references

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014.

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

[I-D.thubert-6lowpan-backbone-router]
Thubert, P., "LoWPAN Backbone Router", draft-thubert-6lowpan-backbone-router-00 (work in progress), March 2008.

[I-D.ietf-netconf-zerotouch]
Watsen, K., Hanna, S., Clarke, J., and M. Abrahamsson, "Zero Touch Provisioning for NETCONF Call Home (ZeroTouch)", draft-ietf-netconf-zerotouch-00 (work in progress), July 2014.

[I-D.kelsey-intarea-mesh-link-establishment]
Kelsey, R., "Mesh Link Establishment", draft-kelsey-intarea-mesh-link-establishment-05 (work in progress), February 2013.

[I-D.ohba-6tisch-security]
Chasko, S., Das, S., Lopez, R., Ohba, Y., Thubert, P., and A. Yegin, "Security Framework and Key Management Protocol Requirements for 6TiSCH", draft-ohba-6tisch-security-01 (work in progress), March 2014.

[I-D.piro-6tisch-security-issues]
Piro, G., Boggia, G., and L. Grieco, "Layer-2 security aspects for the IEEE 802.15.4e MAC", draft-piro-6tisch-security-issues-02 (work in progress), June 2014.

Author's Address

Michael C. Richardson
Sandelman Software Works
470 Dawson Avenue
Ottawa, ON K1Z 5V7
CA

Email: mcr+ietf@sandelman.ca
URI: http://www.sandelman.ca/