

6TiSCH Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 29, 2017

M. Richardson
Sandelman Software Works
February 25, 2017

**Minimal Security rekeying mechanism for 6TiSCH
draft-richardson-6tisch-minimal-rekey-01**

Abstract

This draft describes a mechanism to rekey the networks used by 6TiSCH nodes. It leverages the security association created during an enrollment protocol. The rekey mechanism permits incremental deployment of new sets of keys, followed by a rollover to a new key.

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 August 29, 2017.

Copyright Notice

Copyright (c) 2017 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	2
2.	Terminology	2
3.	An approach to rekeying	3
4.	YANG model for keys	4
5.	YANG model for short-address	6
6.	Security of CoMI link	8
7.	Rekey of master connection	8
8.	Privacy Considerations	8
9.	Security Considerations	8
10.	IANA Considerations	9
11.	Acknowledgments	9
12.	References	9
12.1.	Normative References	9
12.2.	Informative References	9
Appendix A.	Example	10
Author's Address		10

[1.](#) Introduction

6TiSCH networks of nodes often use a pair of keys, K1/K2 to authenticate beacons (K1), encrypt broadcast traffic (K1) and encrypt unicast traffic (K2). These keys need to occasionally be refreshed for a number of reasons:

- o cryptographic hygiene: the keys must be replaced before the ASN roles over or there could be repeated use of the same key.
- o to remove nodes from the group: replacing the keys excludes any nodes that are suspect, or which are known to have left the network
- o to recover short-addresses: if the JRC is running out of short (2-byte) addresses, it can rekey the network in order to garbage collect the set of addresses.

This protocol uses the CoMI [[I-D.ietf-core-comi](#)] to present the set of 127 key pairs.

In addition to providing for rekey, this protocol includes access to the allocated short-address.

[2.](#) Terminology

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 [[RFC2119](#)]. These

Richardson

Expires August 29, 2017

[Page 2]

words may also appear in this document in lowercase, absent their normative meanings.

The reader is expected to be familiar with the terms and concepts defined in [[I-D.ietf-6tisch-terminology](#)], [[RFC7252](#)], [[I-D.ietf-core-object-security](#)], and [[I-D.ietf-anima-bootstrapping-keyinfra](#)].

3. An approach to rekeying

Rekeying of the network requires that all nodes be updated with the new keys. This can take time as the network is constrained, and this management traffic is not highest priority.

The JRC must reach out to all nodes that it is aware of. As the JRC has originally provided the keys via either zero-touch [[I-D.ietf-6tisch-dtsecurity-secure-join](#)] or [[I-D.ietf-6tisch-minimal-security](#)] protocol, and in each case, the JRC assigned the short-address to the node, so it knows about all the nodes.

The data model presented in this document provides for up to 127 K1/K2 keys, as each key requires a secKeyId, which is allocated from a 255-element palette provides by [[IEEE8021542015](#)]. Keys are to be updated in pairs, and the pairs are associated in the following way: the K1 key is always the odd numbered key (1,3,5), and the K2 key is the even numbered key that follows (2,4,6). A secKeyId value of 0 is invalid, and the secKeyId value of 255 is unused in this process.

Nodes MAY support up to all 127 key pair slots, but MUST support a minimum of 6 keys (3 slot-pairs). When fewer than 127 are supported, the node MUST support secKeyId values from 1 to 254 in a sparse array fashion.

A particular key slot-pair is considered active, and this model provides a mechanism to query and also to explicitly set the active pair.

Nodes decrypt any packets for which they have keys, but MUST continue to send using only the keypair which is considered active. Receipt of a packet which is encrypted (or authenticated in the case of a broadcast) with a secKeyId larger (taking consideration that secKeyId wraps at 254) than the active slot-pair causes the node to change active slot pairs.

This mechanism permits the JRC to provision new keys into all the nodes while the network continues to use the existing keys. When the JRC is certain that all (or enough) nodes have been provisioned with

the new keys, then the JRC causes a packet to be sent using the new key. This can be the JRC sending the next Enhanced Beacon or unicast traffic using the new key if the JRC is also a regular member of the LLN. In the likely case that the JRC has no direct connection to the LLN, then the JRC updates the active key to the new key pair using a CoMI message.

The frame goes out with the new keys, and upon receipt (and decryption) of the new frame all receiving nodes will switch to the new active key. Beacons or unicast traffic leaving those nodes will then update additional peers, and the network will switch over in a flood-fill fashion.

((EDNOTE: do we need an example?))

4. YANG model for keys

```
module ietf-6tisch-symmetric-keying {
  yang-version 1.1;

  namespace
    "urn:ietf:params:xml:ns:yang:6tisch-keys";
  prefix "ietf6keys";

  //import ietf-yang-types { prefix yang; }
  //import ietf-inet-types { prefix inet; }

  organization
    "IETF 6tisch Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/6tisch/>
    WG List:  <mailto:6tisch@ietf.org>
    Author:   Michael Richardson
              <mailto:mcr+ietf@sandelman.ca>;

  description
    "This module defines the format for a set of network-wide 802.15.4
    keys used in 6tisch networks.  There are 128 sets of key pairs,
    with one keypair (K1) used to authenticate (and sometimes encrypt)
    multicast traffic, and another keypair (K2) used to encrypt unicast
    traffic.  The 128 key pairs are numbered by the (lower) odd
    keyindex, which otherwise is a 0-255 value.  Keyindex 0 is
    not valid.  This module is a partial expression of the tables in
    https://mentor.ieee.org/802.15/dcn/15/15-15-0106-07-0mag-security-section-
    pictures.pdf";

  revision "2017-03-01" {
    description
```

Richardson

Expires August 29, 2017

[Page 4]

```
    "Initial version";
    reference
        "RFC XXXX: 6tisch minimal security";
}

// top-level container
container ietf6tischkeys {
    config false;
    description
        "A voucher that can be used to assign one or more
        devices to an owner.";

    // secKeyIdMode is always 1, do not describe it here.
    leaf secKeyIndex {
        type uint8;
        description
            "The keyIndex for this keySet. A number between 1 and 255.";

        reference
            "IEEE802.15.4";
    }

    container secKeyUsage {
        leaf txPacketsSent {
            type uint32;
            description "Number of packets sent with this key.";
        }
        leaf rxPacketsSuccess {
            type uint32;
            description "Number of packets received with this key that were
                successfully decrypted and authenticated.";
        }
        leaf rxPacketsReceived {
            type uint32;
            description "Number of packets received with this key, both
                successfully received, and unsuccessfully.";
        }
    }
}

container secKeyDescriptor {
    description
        "This container describes the details of a specific cipher key";
    leaf secKey {
        type binary;
        description "The actual encryption key.
            This value is write only, and is not returned in a
            read, or returns all zeroes.";
```


Richardson

Expires August 29, 2017

[Page 5]

```
    }  
  }  
}  
}
```

5. YANG model for short-address

```
module ietf-6tisch-symmetric-keying {  
  yang-version 1.1;  
  
  namespace  
    "urn:ietf:params:xml:ns:yang:6tisch-keys";  
  prefix "ietf6keys";  
  
  //import ietf-yang-types { prefix yang; }  
  //import ietf-inet-types { prefix inet; }  
  
  organization  
    "IETF 6tisch Working Group";  
  
  contact  
    "WG Web:   <http://tools.ietf.org/wg/6tisch/>  
    WG List:  <mailto:6tisch@ietf.org>  
    Author:   Michael Richardson  
              <mailto:mcr+ietf@sandelman.ca>";  
  
  description  
    "This module defines the format for a set of network-wide  
    802.15.4 keys used in 6tisch networks. There are 128  
    sets of key pairs, with one keypair (K1) used to  
    authenticate (and sometimes encrypt) multicast traffic,  
    and another keypair (K2) used to encrypt unicast traffic.  
    The 128 key pairs are numbered by the (lower) odd  
    keyindex, which otherwise is a 0-255 value.  
    Keyindex 0 is not valid. This module is a partial  
    expression of the tables in  
https://mentor.ieee.org/802.15/dcn/15/15-15-0106-07-0mag-security-section-  
pictures.pdf";  
  
  revision "2017-03-01" {  
    description  
      "Initial version";  
    reference  
      "RFC XXXX: 6tisch minimal security";  
  }  
  
  // top-level container  
  container ietf6tischkeys {  
    config false;
```

Richardson

Expires August 29, 2017

[Page 6]

```
description
  "A voucher that can be used to assign one or more
  devices to an owner.";

// secKeyIdMode is always 1, do not describe it here.
leaf secKeyIndex {
  type uint8;
  description
    "The keyIndex for this keySet. A number between
    1 and 255.";

  reference
    "IEEE802.15.4";
}

container secKeyUsage {
  leaf txPacketsSent {
    type uint32;
    description "Number of packets sent with this key.";
  }
  leaf rxPacketsSuccess {
    type uint32;
    description "Number of packets received with this key
    that were successfully decrypted and authenticated.";
  }
  leaf rxPacketsReceived {
    type uint32;
    description "Number of packets received with this key,
    both successfully received, and unsuccessfully.";
  }
}

container secKeyDescriptor {
  description
    "This container describes the details of a specific
    cipher key";
  leaf secKey {
    type binary;
    description "The actual encryption key";
  }
}
}
```


6. Security of CoMI link

The CoMI resources presented here are protected by OSCOAP ([[I-D.ietf-core-object-security](#)]), secured using the EDHOC connection used for joining. A unique application key is generated using an additional key generation process with the unique label "6tisch-rekey".

7. Rekey of master connection

Should the OSCOAP connection need to be rekeyed, a new EDHOC process will be necessary. This will need access to trusted authentication keys, either the PSK used from a one-touch process, or the locally significant domain certificates installed during a zero-touch process.

8. Privacy Considerations

The rekey protocol itself runs over a network encrypted with the K2 key. The end to end protocol from JRC to node is also encrypted using OSCOAP, so the keys are not visible, nor is the keying traffic distinguished in anyway to an observer.

As the secKeyId is not confidential in the underlying 802.15.4 frames, an observer can determine what sets of keys are in use, and when a rekey is activated by observing the change in the secKeyId.

The absolute value of the monitonically increasing secKeyId could provide some information as to the age of the network.

9. Security Considerations

This protocol permits the underlying network keys to be set. Access to all of the portions of this interface MUST be restricted to an ultimately trusted peer, such as the JRC.

An implementation SHOULD not permit reading the network keys. Those fields should be write-only.

The OSCOAP security for this interface is initialized by a join mechanism, and so depends upon the initial credentials provided to the node. The initial network keys would have been provided during the join process; this protocol permits them to be updated.

10. IANA Considerations

This document allocates a SID number for the YANG model. There is no IANA action required for this document.

11. Acknowledgments

12. References

12.1. Normative References

- [I-D.ietf-core-comi]
Stok, P., Bierman, A., Veillette, M., and A. Pelov, "CoAP Management Interface", [draft-ietf-core-comi-00](#) (work in progress), January 2017.
- [I-D.ietf-core-object-security]
Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security of CoAP (OSCOAP)", [draft-ietf-core-object-security-01](#) (work in progress), December 2016.
- [I-D.ietf-cose-msg]
Schaad, J., "CBOR Object Signing and Encryption (COSE)", [draft-ietf-cose-msg-24](#) (work in progress), November 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [RFC 7049](#), DOI 10.17487/RFC7049, October 2013, <<http://www.rfc-editor.org/info/rfc7049>>.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<http://www.rfc-editor.org/info/rfc7252>>.

12.2. Informative References

- [I-D.ietf-6tisch-6top-protocol]
Wang, Q. and X. Vilajosana, "6top Protocol (6P)", [draft-ietf-6tisch-6top-protocol-03](#) (work in progress), October 2016.

- [I-D.ietf-6tisch-dtsecurity-secure-join]
Richardson, M., "6tisch Secure Join protocol", [draft-ietf-6tisch-dtsecurity-secure-join-00](#) (work in progress), December 2016.
- [I-D.ietf-6tisch-minimal-security]
Vucinic, M., Simon, J., and K. Pister, "Minimal Security Framework for 6TiSCH", [draft-ietf-6tisch-minimal-security-01](#) (work in progress), February 2017.
- [I-D.ietf-6tisch-terminology]
Palattella, M., Thubert, P., Watteyne, T., and Q. Wang, "Terminology in IPv6 over the TSCH mode of IEEE 802.15.4e", [draft-ietf-6tisch-terminology-08](#) (work in progress), December 2016.
- [I-D.ietf-anima-bootstrapping-keyinfra]
Pritikin, M., Richardson, M., Behringer, M., Bjarnason, S., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructures (BRSKI)", [draft-ietf-anima-bootstrapping-keyinfra-04](#) (work in progress), October 2016.
- [IEEE8021542015]
IEEE standard for Information Technology, ., "IEEE Std 802.15.4-2015 Standard for Low-Rate Wireless Personal Area Networks (WPANs)", 2015.

[Appendix A](#). Example

Example COMI requests/responses.

Author's Address

Michael Richardson
Sandelman Software Works

Email: mcr+ietf@sandelman.ca

