YANG Data Model for RFC 7210 Key Table

draft-chen-rtgwg-key-table-yang-00
Goals

• YANG data model for configuring cryptographic keys for routing protocols
  – Based on key table defined in RFC 7210
    • Conceptual key database
    • Accommodates different key management implementations
    • Accommodates different routing protocols
    • Accommodates different security protocols

• Inter-operable key management solution that uses NETCONF and key-table YANG model
RFC 7210 Key Table

- A database of keys
- Heterogeneous deployments

Properties of a key: allow for different types of key, e.g. with HMAC-SHA-1 KDF, without KDF, uses AES-128-CMAC.
**OSPF Authentication**  
(RFC 2328 Appendix D.3)

### Router ID 1.1.1.1

<table>
<thead>
<tr>
<th>Admin Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface(s)</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KDF</th>
<th>AlgID</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetime Start</th>
<th>Send Lifetime End</th>
<th>Accept Lifetime Start</th>
<th>Accept Lifetime End</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>5</td>
<td>5</td>
<td>2.2.2.2</td>
<td>all</td>
<td>ospf</td>
<td>NA</td>
<td>none</td>
<td>hmac</td>
<td>0x0..</td>
<td>both</td>
<td>T1</td>
<td>T2</td>
<td>T1 + 1</td>
<td>T2 - 1</td>
</tr>
<tr>
<td>k2</td>
<td>7</td>
<td>7</td>
<td>2.2.2.2</td>
<td>all</td>
<td>ospf</td>
<td>NA</td>
<td>none</td>
<td>hmac</td>
<td>0x1..</td>
<td>both</td>
<td>T5</td>
<td>T6</td>
<td>T5 + 1</td>
<td>T6 - 1</td>
</tr>
</tbody>
</table>

### Router ID 2.2.2.2

<table>
<thead>
<tr>
<th>Admin Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface(s)</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KDF</th>
<th>AlgID</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetime Start</th>
<th>Send Lifetime End</th>
<th>Accept Lifetime Start</th>
<th>Accept Lifetime End</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>5</td>
<td>5</td>
<td>1.1.1.1</td>
<td>all</td>
<td>ospf</td>
<td>NA</td>
<td>none</td>
<td>hmac</td>
<td>0x0..</td>
<td>both</td>
<td>T1</td>
<td>T2</td>
<td>T1 + 1</td>
<td>T2 - 1</td>
</tr>
<tr>
<td>L2</td>
<td>7</td>
<td>7</td>
<td>1.1.1.1</td>
<td>all</td>
<td>ospf</td>
<td>NA</td>
<td>none</td>
<td>hmac</td>
<td>0x1..</td>
<td>both</td>
<td>T5</td>
<td>T6</td>
<td>T5 + 1</td>
<td>T6 - 1</td>
</tr>
</tbody>
</table>
**RSVP Authentication (RFC 2747)**

**Router ID 1.1.1.1**

<table>
<thead>
<tr>
<th>Admin Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KDF</th>
<th>AlgID</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetime Start</th>
<th>Send Lifetime End</th>
<th>Accept Lifetime Start</th>
<th>Accept Lifetime End</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>15</td>
<td>2.2.2.2</td>
<td>all</td>
<td>rsvp</td>
<td>NA</td>
<td>none</td>
<td>aes</td>
<td>...</td>
<td>0x0..</td>
<td>in</td>
<td>T1</td>
<td>T2</td>
<td>T1 + 1</td>
<td>T2 - 1</td>
</tr>
<tr>
<td>A2</td>
<td>17</td>
<td>2.2.2.2</td>
<td>all</td>
<td>rsvp</td>
<td>NA</td>
<td>none</td>
<td>aes</td>
<td>...</td>
<td>0x1..</td>
<td>in</td>
<td>T5</td>
<td>T6</td>
<td>T5 + 1</td>
<td>T6 - 1</td>
</tr>
</tbody>
</table>

**Router ID 2.2.2.2**

<table>
<thead>
<tr>
<th>Admin Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KDF</th>
<th>AlgID</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetime Start</th>
<th>Send Lifetime End</th>
<th>Accept Lifetime Start</th>
<th>Accept Lifetime End</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>19</td>
<td>1.1.1.1</td>
<td>all</td>
<td>rsvp</td>
<td>NA</td>
<td>none</td>
<td>aes</td>
<td>...</td>
<td>0x2..</td>
<td>in</td>
<td>T1</td>
<td>T2</td>
<td>T1 + 1</td>
<td>T2 - 1</td>
</tr>
<tr>
<td>p2</td>
<td>21</td>
<td>1.1.1.1</td>
<td>all</td>
<td>rsvp</td>
<td>NA</td>
<td>none</td>
<td>aes</td>
<td>...</td>
<td>0x3..</td>
<td>in</td>
<td>T5</td>
<td>T6</td>
<td>T5 + 1</td>
<td>T6 - 1</td>
</tr>
</tbody>
</table>
Key Table YANG Model

<table>
<thead>
<tr>
<th>Admin Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KDF</th>
<th>AlgId</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetime Start</th>
<th>Send Lifetime End</th>
<th>Accept Lifetime Start</th>
<th>Accept Lifetime End</th>
</tr>
</thead>
</table>

```Yang
+--rw security-association-entry* [admin-key-name]
   +--rw admin-key-name string
   +--rw local-key-name string
   +--rw peer-key-name string
   +--rw peers
      +--rw interfaces
         |   +--rw (interface-options)
         |   +--:(all-interfaces)
         |   |   +--rw all? Empty
         |   +--:(interface-list)
         |   |   +--rw interface* if:interface-ref
         +--rw protocol identityref
         +--rw protocol-specific-info
         +--rw kdf key-derivation-function-type
         +--rw alg-id cryptographic-algorithm-type
         +--rw key yang:hex-string
         +--rw direction enumeration
         +--rw send-lifetime-start lifetime-type
         +--rw send-lifetime-end lifetime-type
         +--rw accept-lifetime-start lifetime-type
         +--rw accept-lifetime-end lifetime-type
```

Defined as containers (i.e. YANG placeholder) and left for routing protocols to augment.
Relationship with Other Modules

- An independent tree
  - Does not augment from key-chain module
- Links to ietf-interfaces
- Routing protocols link to this module

```
ietf-key-table
  +--rw key-table
    +--rw security-association-entry* [admin-key-name]
      +--rw admin-key-name
      +--rw ...
    +--rw interfaces
      |  +--rw (interface-options)
      |  +--:(all-interfaces)
      |   |  +--rw all? Empty
      |  +--:(interface-list)
      |   |  +--rw interface* if:interface-ref
      +--rw ...

ietf-interfaces
  +--rw interfaces
    |  +--rw interface* [name]
    |   |  +--rw name
    |   +--rw ...
    +--ro interface-state
      +--ro interface* [name]
      +--ro name
      +--ro ...

foo-routing-protocol
  module: foo-routing-protocol
  augment /rt:routing/rt:routing-instance/rt:routing-protocols/rt:routing-protocol:
    +--rw foo
      +--rw interface* [name]
        +--rw name              if:interface-ref
        +--rw authentication
          +--rw out-key
            |  +--rw peer-identifier? string
            +--rw in-key
              +--rw key-identifier? local-key-ref
              +--rw peer-identifier? string

Need to update draft
```
Comparison: Configuration

key-table

+--rw key-table
  +--rw security-association-entry* [admin-key-name]
    +--rw admin-key-name              string
    +--rw local-key-name              string
    +--rw peer-key-name               string
    +--rw peers
    +--rw interfaces
      |  +--rw (interface-options)
      |  +--:(all-interfaces)
      |  |  +--rw all?               Empty
      |  +--:(interface-list)
      |  |  +--rw interface*         if:interface-ref
    +--rw protocol                 identityref
    +--rw protocol-specific-info
    +--rw kdf                      key-derivation-function-type
    +--rw alg-id                   cryptographic-algorithm-type
    +--rw key                      yang:hex-string
    +--rw direction                enumeration
    +--rw send-lifetime-start      lifetime-type
    +--rw send-lifetime-end        lifetime-type
    +--rw accept-lifetime-start    lifetime-type
    +--rw accept-lifetime-end      lifetime-type
  +--rw key-chains
    +--rw key-chain-list* [name]
      +--rw name                                 string
      +--rw accept-tolerance {accept-tolerance}?
        |  +--rw duration?                         uint32
      +--rw key-chain-entry* [key-id]
        +--rw key-id                            uint64
        +--rw key-string
          |  +--rw (key-string-style)?
          |     +--:(hexadecimal) {hex-key-string}?  
          |     +--rw hexadecimal-string?   yang:hex-string
        +--rw lifetime
          |  +--rw (lifetime)?
          |     +--:(send-and-accept-lifetime)
          |     |  +--rw ...
          |     +--:(independent-send-accept-lifetime)
          |     |  +--rw independent-send-accept-lifetime
          |     |     |  +--rw send-lifetime
          |     |     |     |  +--rw ...
          |     |     +--rw accept-lifetime
          |     |     |  +--rw ...
          |     +--rw crypto-algorithm
          |     +--rw (algorithm)?
            |  +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
            |     +--rw hmac-sha1-12?         empty
            |  +--:(md5)
            |     +--rw md5?                  empty
            |  +--:(sha-1)
            |     +--rw sha-1?                empty
            |  +--:(hmac-sha-1)
            |     +--rw hmac-sha-1?           empty
            |  +--:(hmac-sha-256)
            |     +--rw hmac-sha-256?         empty
            |  +--:(hmac-sha-384)
            |     +--rw hmac-sha-384?         empty
            |  +--:(hmac-sha-512)
            |     +--rw hmac-sha-512?         empty

key-chain

Extra layer in key-chain

Missing from key-chain
# Mapping

<table>
<thead>
<tr>
<th>KeyTable</th>
<th>OSPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin Key Name</td>
<td>N/A</td>
</tr>
<tr>
<td>LocalKeyName</td>
<td>OSPF KeyID</td>
</tr>
<tr>
<td>PeerKeyName</td>
<td>N/A, SHOULD equal LocalKeyName</td>
</tr>
<tr>
<td>Peers</td>
<td>KeyChainName or empty</td>
</tr>
<tr>
<td>Interfaces</td>
<td>For nonempty Peers, MUST equal “all”</td>
</tr>
<tr>
<td></td>
<td>For empty Peers, specifies interfaces</td>
</tr>
<tr>
<td>Protocol</td>
<td>OSPF (register with IANA) [used only for lookup]</td>
</tr>
<tr>
<td>ProtocolSpecificInfo</td>
<td>N/A, empty</td>
</tr>
<tr>
<td>KDF</td>
<td>MUST be “None”</td>
</tr>
<tr>
<td>AlgID</td>
<td>{register with IANA}</td>
</tr>
<tr>
<td>Key</td>
<td>Key</td>
</tr>
<tr>
<td>Direction</td>
<td>MUST be “both”</td>
</tr>
<tr>
<td>SendLifetimeStart</td>
<td>Use as start value</td>
</tr>
<tr>
<td>SendLifetimeEnd</td>
<td>Use as end value</td>
</tr>
<tr>
<td>AcceptLifetimeStart</td>
<td>For systems with a single “accept tolerance” value, N/A</td>
</tr>
<tr>
<td></td>
<td>For systems with two “accept tolerance” values, set tolerance to difference(SendLifetimeEnd,AcceptLifetimeEnd)</td>
</tr>
<tr>
<td>AcceptLifetimeEnd</td>
<td>For systems with a single “accept tolerance” value, set tolerance to difference(Send lifetimeEnd,AcceptLifetimeEnd)</td>
</tr>
<tr>
<td></td>
<td>For systems with two “accept tolerance” values, set tolerance to difference(Send Lifetime End,Accept Lifetime End)</td>
</tr>
</tbody>
</table>
## Comparison Summary

<table>
<thead>
<tr>
<th>key-table</th>
<th>key-chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>• An conceptual database of security associations (keys)</td>
<td>• An abstraction of an implementation</td>
</tr>
<tr>
<td>• Defines all attributes included in RFC 7210</td>
<td>• Defines a subset of attributes in RFC 7210</td>
</tr>
<tr>
<td>• Supports multiple security deployments</td>
<td>• Supports a particular security deployment</td>
</tr>
<tr>
<td>• Does not have operational state yet</td>
<td>• Replicates some configuration data</td>
</tr>
<tr>
<td>- Can be added</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- Introduce a key-table YANG model
  - Based on RFC 7210
  - Conceptual database of keys
  - Map to different implementations
  - Support different routing protocols
  - Support different security protocols

- Introduce an inter-operable solution to manage keys
  - NETCONF
  - key-table YANG model
Next Steps

• What does WG want to standardize?
  – Overlapping topics
    • draft-chen-rtgwg-key-table-yang
    • draft-acee-rtg-key-chain-yang
  – Tangential
    • draft-tran-ipecme-yang-ipsec
    • draft-wang-ipsec-ipsec-yang
    • draft-wang-ipsec-ike-yang
Questions/Comments
OSPF YANG Model

```yang
+-rw authentication
 | +--rw (auth-type-selection)?
 |     +--:(auth-ipsec) {ospfv3-authentication-ipsec}?
 |         +--rw sa?                string
 |     +--:(auth-trailer-key-chain)
 |         +--rw key-chain?        key-chain:key-chain-ref
 |     +--:(auth-trailer-key)
 |         +--rw key?              string
 |         +--rw crypto-algorithm
 |         +--rw (algorithm)?
 |             +--:(hmac-sha-1-12) {crypto-hmac-sha-1-12}?
 |                 +--rw hmac-sha-1-12?   empty
 |             +--:(md5)
 |                 +--rw md5?            empty
 |             +--:(sha-1)
 |                 +--rw sha-1?          empty
 |             +--:(hmac-sha-1)
 |                 +--rw hmac-sha-1?   empty
 |             +--:(hmac-sha-256)
 |                 +--rw hmac-sha-256?   empty
 |             +--:(hmac-sha-384)
 |                 +--rw hmac-sha-384?   empty
 |             +--:(hmac-sha-512)
 |                 +--rw hmac-sha-512?   empty
```
ISIS YANG model

```plaintext
+---rw (authentication-type)?
  |   +--:(key-chain) {key-chain}?
  |   |   +--rw key-chain?      key-chain:key-chain-ref
  |   |   +--:(password)
  |   |   +--rw key?            string
  |   |   +--rw (algorithm)?
  |   |   |   +--:(hmac-sha1-12)
  |   |   |   |   ...          ...
  |   |   |   +--:(hmac-sha1-20)
  |   |   |   |   ...          ...
  |   |   |   +--:(md5)
  |   |   |   |   ...          ...
  |   |   |   +--:(sha-1)
  |   |   |   |   ...          ...
  |   |   |   +--:(hmac-sha-1)
  |   |   |   |   ...          ...
  |   |   |   +--:(hmac-sha-256)
  |   |   |   |   ...          ...
  |   |   |   +--:(hmac-sha-384)
  |   |   |   |   ...          ...
  |   |   |   +--:(hmac-sha-512)
  |   |   |   ...          ...
  |   |   ...
  |   ...
  ...
```
RFC 7210 Key Table

- A single database
- Heterogeneous deployment

<table>
<thead>
<tr>
<th>Adm Key Name</th>
<th>Local Key Name</th>
<th>Peer Key Name</th>
<th>Peers</th>
<th>Interface</th>
<th>Protocol</th>
<th>Protocol Specific Info</th>
<th>KD F</th>
<th>Alg D</th>
<th>Key</th>
<th>Direction</th>
<th>Send Lifetim e Start</th>
<th>Send Lifetim e End</th>
<th>Accept Lifetim e Start</th>
<th>Accept Lifetim e End</th>
</tr>
</thead>
</table>

- For smooth key rollover
- For different receiving and sending keys
- Specified by the protocol
- Together allow for different sending and receiving keys

Properties of a key: allow for different types of key, e.g. with HMAC-SHA-1 KDF, without KDF, uses AES-128-CMAC.