LoRaWAN Authentication with RADIUS

draft-garcia-radext-radius-lorawan-01

Dan Garcia-Carrillo (University of Murcia)
Rafael Marin-Lopez (University of Murcia)
Arunprabhu Kandasamy (Acklio)
Alexander Pelov (Acklio)
LoRaWAN – *Long Range Wide Area Network

- Long range: upto 20km
  **(depending on environment)**
- Low Power: 25mW, 20yrs battery life
- Data Rate: Upto 50kbps
- Payload: 11-242 bytes

*Alexander Pelov, Alper Yegin Slides from Ip-wan BoF
**https://hal-institut-mines-telecom.archives-ouvertes.fr/hal-01331966
LoRaWAN – *Long Range Wide Area Network

- Long range: upto 20km
  *(depending on environment)*
- Low Power: 25mW, 20yrs battery life
- Data Rate: Upto 50kbps
- Payload: 11-242 bytes

*Alexander Pelov, Alper Yegin Slides from lp-wan BoF
**https://hal-institut-mines-telecom.archives-ouvertes.fr/hal-01331966
LoRaWAN Authentication - Motivation

• LoRaWAN does not reuse standards

• Include a standard authentication, AAA, framework in LoraWAN.

  – AAA infrastructure proven to be well known, battle-tested techs. Deployed in the wild, since years.. ex: eduroam

  – *Scalable, federation aware

* “Security in IoT” lectures from Rafa Marin Lopez, University of Murcia Spain, at Telecom Bretagne, France
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning
2. Over the Air Activation (Join Procedure)
LoRaWAN Authentication - Message Definition

• The request (join-request)

<table>
<thead>
<tr>
<th>Size(bytes)</th>
<th>8</th>
<th>8</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join Request</td>
<td>AppEUI</td>
<td>DevEUI</td>
<td>DevNonce</td>
</tr>
</tbody>
</table>

• The Response (join-accept)

<table>
<thead>
<tr>
<th>Size(bytes)</th>
<th>3</th>
<th>3</th>
<th>4</th>
<th>1</th>
<th>1</th>
<th>16(opt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join Accept</td>
<td>AppNonce</td>
<td>NetID</td>
<td>DevAddr</td>
<td>DLSettings</td>
<td>RxDelay</td>
<td>CFList</td>
</tr>
</tbody>
</table>
LoRaWAN Authentication - Keys

• AppKey
  - specific for the end-device that is assigned by the application owner to the end-device

• NwkSKey
  - to decrypt the MAC commands

• AppSKey
  - to decrypt the Application specific data
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning
LoRaWAN Authentication - LoRaWAN 1.0 Join procedure

1. Commissioning
2. Over the Air Activation (Join Procedure)
1. Commissioning
2. Over the Air Activation (Join Procedure)
3. RADIUS Extension

* MIC field empty. To be calculated by the RADIUS server.
** Optional field.
LoRaWAN Authentication with RADIUS

• New RADIUS Attributes
  – JoinRequest (containing the join-request)
  – JoinAnswer (containing the join-accept)
  – Nwkskey (containing the NwkSKey)
  – Appskey (containing the AppSKey, optional)

• Keys are transported as RADIUS attributes
  Consideration for transporting key materials securely
    • Similar to RFC 6218 (Cisco Vendor-Specific RADIUS Attributes for the Delivery of Keying Material)
    • RFC6614 (Radius over TLS)
LoRaWAN Authentication with RADIUS

• Open Issues
  – The Join Request has AppEUI indicating the Organization, but to route JR and JA* through the AAA infrastructure we need to specify a realm (e.g. um.es).

  00 11 22 33 44 55 66 77 88 99 AA um.es

  – A mechanism for matching the AppEUI to the domain name of the organization is needed.

  – Possible solution would be an inverse approach of [RFC7043] using DNS.
LoRaWAN Authentication with RADIUS

• Proof of concept Implementation
  - End Device (Nemeus)
    - usb key with Java app
  - Base station (ExpEmB)
    - Intel Atom, 2GB RAM
  - Lora Network server (Acklio)
    - implemented in golang
  - RADIUS (bronze1man)
    - implemented in golang

Next: Implementation in FreeRADIUS

• [LoRa Network Server] www.ackl.io
• [Radius] github.com/bronze1man/radius
LoRaWAN Authentication with RADIUS

• Acknowledgements
  – Thanks, to Sri Gundavelli, Yeoh Chun-Yeow, Alan DeKok, Stephen Farrell and Mark Grayson., for their valuable comments
  – This work has been possible partially by:
    • The SMARTIE project (FP7-SMARTIE-609062 EU Project)
    • The Spanish National Project CICYT EDISON (TIN2014-52099-R) granted by the Ministry of Economy and Competitiveness of Spain (including ERDF support).
Comments and Questions?

- Thanks for your attention
Backup Slides

- Key and MIC calculation

join-request
\[
\text{cmac} = \text{aes128_cmac}(\text{AppKey}, \text{MHDR} | \text{AppEUI} | \text{DevEUI} | \text{DevNonce})
\]

\[
\text{MIC} = \text{cmac}[0..3]
\]

join-accept
\[
\text{cmac} = \text{aes128_cmac}(\text{AppKey}, \text{MHDR} | \text{AppNonce} | \text{NetID} | \text{DevAddr} | \text{RFU} | \text{RxDelay} | \text{CFList})
\]

\[
\text{MIC} = \text{cmac}[0..3]
\]

\[
\text{NwkSKey} = \text{aes128_encrypt}(\text{AppKey}, 0x01 | \text{AppNonce} | \text{NetID} | \text{DevNonce} | \text{pad16})
\]

\[
\text{AppSKey} = \text{aes128_encrypt}(\text{AppKey}, 0x02 | \text{AppNonce} | \text{NetID} | \text{DevNonce} | \text{pad16})
\]

The pad16 function appends zero octets so that the length of the data is a multiple of 16.

\[
\text{MIC}*[\text{RFC4493}]
\]
Backup Slides

• **Size of RADIUS attributes**
  (incl. Type & length fields)
  Request = 25B
  Accept = 19B
  NwkSKey = 34B
  AppSKey = 34B
Backup Slides

• **Nonce**
  
The DevNonce can be extracted by issuing a sequence of RSSI measurements under the assumption that the quality of randomness fulfills the criteria of true randomness.