‘Secure Device Identity’
Profile for TSN-IA

IEEE Plenary; March 07, 2022

Kai Fischer, Andreas Furch, Oliver Pfaff
Problem Statement

- This is a follow-up to the IEEE January plenary session ‘Secure Device Identity’ Profile for TSN-IA, 2022-01-21 (https://www.ieee802.org/1/files/public/docs2022/60802-Pfaff-et-al-Secure-Device-Identity-Profile-0122-v01.pdf)

- Recap (2022-01-21):
  - IDevID EE certificate design variance: identified the driving factors
  - Different identity models (by different bodies) for the same physical entity. sketched a co-existence model
  - Home of this information: preferred the SubjectAlternativeName extension

- Next step (2022-03-07):
  - LDevID-NETCONF* and IDevID EE certificate design variance. discuss the need for limitation
  - ‘Device’ model in IEC/IEEE 60802: discuss implications on the LDevID-NETCONF/IDevID incarnations (per ‘device’)  
  - ‘Device identity’ model in IEC/IEEE 60802. discuss implications on LDevID-NETCONF/IDevID contents

*: short-hand term for an LDevID (IEEE STD 802.1AR) that complies with the IETF RFC 7589 rules for NETCONF-over-TLS
LDevID-NETCONF/IDevID Use Cases

LDevID-NETCONF

- **Protect message exchanges** (NETCONF-over-TLS)
- **Authorize resource accesses** (NACM)

Note: solutions for these use cases are already specified in a comprehensive way:
- NETCONF-over-TLS: IETF RFC 7589
- NACM: IETF RFC 8341

IDevID

- **Prepare** for NETCONF-over-TLS i.e. set-up and manage LDevID-NETCONF credentials/trust anchors and certificate-to-name mappings
- **Verify** the original equipment manufacturer (*counterfeit protection*)
- **Check** whether an IA station is an *instance-in-class* (*system integrity/resilience*)
- …

Note: not all use cases already have solutions with comprehensive specifications
## LDevID-NETCONF/IDevID Usage Models

<table>
<thead>
<tr>
<th>LDevID-NETCONF</th>
<th>IDevID</th>
</tr>
</thead>
<tbody>
<tr>
<td>• <strong>Frequently</strong> used: see use cases above</td>
<td>• <strong>Rarely</strong> used: see use cases above</td>
</tr>
<tr>
<td>• <strong>Occasionally</strong> updated (usually according a time schedule)</td>
<td>• <strong>Not</strong> updated</td>
</tr>
<tr>
<td>• Processed by verifiers in an <strong>automated</strong> fashion i.e. without human user-attention</td>
<td>• (May be) processed by verifiers in an <strong>automated</strong> fashion i.e. without human user-attention</td>
</tr>
</tbody>
</table>

**Design variance** for LDevIDs-NETCONF and IDevIDs needs to be **limited** to facilitate automated processing in multi-vendor environments

*: IETF RFCs 7589 and 6125 do not detail IDevID contents
**LDevID-NETCONF/IDevID Verification/Consumption**

**LDevID-NETCONF**

1. **Certification path validation**: IETF RFC 5280
2. **Proof-of-Possession checking**: IETF RFC 5246
3. **Entity checking**:
   - Client identity:
     - IETF RFC 7589 (certificate-to-name mapping)
     - IETF RFC 8341 (authorization)
   - Server identity: IETF RFC 7589 (expected vs. actual based on addressing info [expected] and SAN extension content* [actual])

**IDevID**

1. **Certification path validation**: IETF RFC 5280
2. **Proof-of-Possession checking**: IETF RFC 5246**
3. **Entity checking**:
   - Client identity: n.a. in IEC/IEEE 60802
   - Server identity: IETF RFC 7589 (expected vs. actual based on addressing info [expected] and SAN extension content* [actual])
     - Owner/operator-selected strategies (tbd); upfront candidates:
       i. Any end station/bridge that can authenticate
       ii. …and that is from a dedicated manufacturer
       iii. …and that belongs to a dedicated type
       iv. …and that is a dedicated instance

*: see IETF RFC 6125 for the details
**: for IDevID usage with NETCONF-over-TLS
Represented a single NETCONF/YANG server which serves a single YANG info model that can be reached through multiple network interfaces. Also represents a single NETCONF client.

*: showing an exemplary composition by functional units as well as end stations and bridges.
Device Model: LDevID-NETCONF/IDevID Implications

LDevID-NETCONF

- **Number of credentials in an IA station**: by default a **singleton**
- This credential is used by the single NETCONF/YANG server on the IA station - independent from the IP address by which this server is addressed
- Its EE certificate shall contain #1..n instance identifiers in form of #1 SAN extension providing #1..n dNSName (or iPAddress) values

IDevID

- **Number of in an IA station**: by default a **singleton** (per orderable item)
- This credential is used by the single NETCONF/YANG server on the IA station - independent from the IP address by which this server is addressed
- Its EE certificate can not contain instance identifiers in form of dNSName (or iPAddress); these values are not known by the manufacturer (in product business)
Device Identity Model: YANG Module ietf-hardware

- This section uses IETF RFC 8348 to describe a ‘device identity’ model for IEC/IEEE 60802:
  - IETF RFC 8348 provides a “YANG data model for the management of hardware on a single server”
  - This helps to make the consideration of ‘secure device identity’ concrete. It is meant to be non-normative.
- IETF RFC 8348 defines the YANG modules iana-hardware and ietf-hardware
  - iana-hardware is a predefined and extensible enumeration of understood purposes for a piece of HW
    - Example values: backplane, chassis, cpu, power supply...
  - ietf-hardware provides a YANG information model to describe a variety of HW-based products in the interval [atomic...composite]
    - The container hardware incarnates an ietf-hardware info model. It provides a list of component items
    - A child object component describes a HW item that is classified by the iana-hardware enumeration. The management of its values is a shared concern between manufacturers and users:
      - ro items are manufacturer-assigned values that are part of the state data (can not be configured)
      - rw items are user-managed values that are part of the config data (may be configured)
    - See [here](#) for a HelloWorld example of an ietf-hardware object in JSON notation
**Device Identity Model: ro Child* Elements in component**

- **physical-index** (int32, 0..1): a manufacturer-assigned index value to refer to other component objects for Entity-MIBs
- **description** (string, 0..1) a manufacturer-assigned description
- **contains-child** (string, 0..n): a manufacturer-assigned reference to a component object in the same hardware object
- **hardware-rev** (string, 0..1): a manufacturer-assigned hardware revision for the component
- **firmware-rev** (string, 0..1): a manufacturer-assigned firmware revision for the component
- **software-rev** (string, 0..1): a manufacturer-assigned software revision for the component
- **serial-num**: (string, 0..1): a manufacturer-assigned serial number for the component (unique per mfg-name)
- **mfg-name** (string, 0..1): a manufacturer-assigned name for the component manufacturer (no registry authority resp. registration process is mentioned in IETF RFC 8348)
- **model-name** (string, 0..1): a manufacturer-assigned model name associated with the physical entity
- **is-fru** (boolean, 0..1): a manufacturer-assigned Boolean identifying if the physical entity described by this component is replaceable
- **mfg-date** (date-time, 0..1): a manufacturer-assigned date of manufacturing of the component
- **uuid**: (uuid, 0..1): a manufacturer-assigned universally unique identifier for this component

*: there are also ro grand-child elements in component. They are not considered for brevity
Device Identity Model: IA Stations

<table>
<thead>
<tr>
<th>Functional unit 1</th>
<th>Functional unit 2</th>
<th>Functional unit 3</th>
</tr>
</thead>
</table>

*Single instance of component with class chassis in single container hardware*

*Single instance of YANG module ietf-hardware*

*: showing an exemplary hardware information model*
Device Identity Model: Protection Level

- The information contained in a hardware container is not protected on application-level. This information is only protected during transit (NETCONF-over-TLS).
- Using LDevID-NETCONF credentials, clients (CNC):
  - Can verify whether a response to a hardware retrieval request was sent by an authenticated endpoint over an authenticated transport.
  - Can not verify whether the passed information object is genuine i.e. whether or not its manufacturer-supplied subset of a hardware container originates from the claimed manufacturer and was not modified.
- This creates risks such as product counterfeiting and system integrity/resilience violations resulting from manipulations such as:
  - False mfg-name values (illustrated)
  - False model-name values
Device Identity Model: LDevID-NETCONF/IDevID

Implications

<table>
<thead>
<tr>
<th>LDevID-NETCONF</th>
<th>IDevID</th>
</tr>
</thead>
<tbody>
<tr>
<td>• EE certificate contents for NETCONF/YANG servers:</td>
<td>• EE certificate contents for NETCONF/YANG servers:</td>
</tr>
<tr>
<td>• SAN extension with dNSName and/or iPAddress (IETF RFC 7589, also see RFC 6125)</td>
<td>• Needs to bind identification information to safeguard manufacturer-specific information in the ietf-hardware info model (see below for details)</td>
</tr>
</tbody>
</table>
Secure Device Identity: ‘Device Identity’ Protection Demand

LDevID-NETCONF

- The protection of manufacturer-supplied content in hardware containers is not necessary
- Rationale: manufacturer-supplied content in hardware containers is checked initially based on an IDevID (and might by re-checked later on this basis); no recurring checks are needed whenever LDevID-NETCONF credentials are used

IDevID

- The protection of manufacturer-supplied content in hardware containers is needed to address use cases such as:
  - Verify the original equipment manufacturer (counterfeit protection)
  - Check whether an IA station is an instance-in-class (system integrity/resilience)
Secure Device Identity: Protection Model

This protection can be provided by-reference i.e. an otherName in the IDevID EE certificate carrying:

- a reference to an instance of ietf-hardware information object or a part thereof (e.g. component)
- an identification of pre-digesting transform(s) e.g. extraction of manufacturer-assigned information
- an identification of the digesting algorithm e.g. SHA2-256 or SHA2-512
- the digest-value of the referred and transformed information in an ietf-hardware instance

Notes:
- ‘Secure device identity’ additions are shown in bold
- The by-reference binding requires verifiers to obtain 2 objects (IDevID certification path via TLS and hardware container via NETCONF-over-TLS) and to assess these 2 objects in conjunction
Secure Device Identity: Alternative Binding Approaches

- **By-value**: the IDevID EE certificate becomes self-contained (no 2nd object needed). But this duplicates a potentially large set of information items (hardware with n>1 component child objects), requires YANG-to-ASN.1 syntax transformation for a complex information model (ietf-hardware) esp. its linking feature ➔ not preferred
- **By-URL** (pointing to a protected object): requires verifiers to obtain 3 or more objects (IDevID certification path via TLS and hardware container plus a new object providing e.g. a detached signature for manufacturer-specific contents in hardware container via NETCONF-over-TLS) and to assess these objects in conjunction ➔ not preferred
- **By-URL** (pointing to an unprotected object): not smart ➔ not viable
- **By-context** (Issuer field in IDevID EE certificate vs. mfg-name in ietf-hardware instance): does not protect other manufacturer-specific values in an ietf-hardware instance e.g. model-name. Introduces syntax-rooted issues: the Issuer field is an X.500 name (DN) e.g. CN=TSN-IA Product CA,OU=IT Department,DC=Alice US,DC=COM. The mfg-name could be a friendly name e.g. “Alice Inc.” ➔ not preferred
Secure Device Identity: Subject-Specific LDevID-NETCONF/IDevID EE Certificate Contents

**LDevID-NETCONF**
- EE certificate contents for NETCONF/YANG servers:
  - SAN extension with `dNSName` and/or `iPAddress` (IETF RFC 7589, also see IETF RFC 6125)

**IDevID**
- EE certificate contents for NETCONF/YANG servers:
  - SAN extension with `otherName` carrying:
    - a **reference** to an instance of an `ietf-hardware` information object or a part thereof
    - an identification of **pre-digesting transform(s)**
    - an identification of the **digesting algorithm**
    - the **digest-value** of the referred and transformed information in an `ietf-hardware` instance
Summary, Follow-Ups

• Reminder: IDevID credentials are required to perform the security setup of an IA-station in factory default according the IETF-defined security model for NETCONF/YANG.

• The automation of this security setup demands a common design for IDevID EE certificates

• With some additions more use cases can be facilitated by the IDevID credentials. This includes counterfeiting protection and system integrity/resilience.

• Required follow-ups:

• Further elaborate on the ‘secure device identity’ model for IEC/IEEE 60802:
  • By-reference details esp. reference mechanism
  • Pre-digesting transform(s) details
  • Details for assessing IDevID EE certificates and ietf-hardware information bound to it; supporting several assessment strategies

• Further elaborate on the ‘device identity’ model for IEC/IEEE 60802 (note: this is at the boundary but not inside the security turf)
### Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASN.1</td>
<td>Abstract Syntax Notation no. 1</td>
<td>LDevID</td>
<td>Locally significant Device ID</td>
</tr>
<tr>
<td>CA</td>
<td>Certification Authority</td>
<td>MAC</td>
<td>Media Access Control</td>
</tr>
<tr>
<td>Cert</td>
<td>Certificate</td>
<td>MQTT</td>
<td>Message Queuing Telemetry Transport</td>
</tr>
<tr>
<td>CNC</td>
<td>Centralized Network Configuration</td>
<td>NETCONF</td>
<td>NETwork CONFiguration</td>
</tr>
<tr>
<td>CRL</td>
<td>Certificate Revocation List</td>
<td>OID</td>
<td>Object ID</td>
</tr>
<tr>
<td>CUC</td>
<td>Centralized User Configuration</td>
<td>PLC</td>
<td>Programmable Logic Controller</td>
</tr>
<tr>
<td>DHCP</td>
<td>Dynamic Host Configuration Protocol</td>
<td>PoP</td>
<td>Proof-of-Possession</td>
</tr>
<tr>
<td>DIY</td>
<td>Do It Yourself</td>
<td>RA</td>
<td>Registration Authority</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name Service</td>
<td>ro</td>
<td>read only</td>
</tr>
<tr>
<td>EE</td>
<td>End Entity</td>
<td>rw</td>
<td>read write</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transfer Protocol</td>
<td>SAN</td>
<td>Subject Alternative Name</td>
</tr>
<tr>
<td>IA</td>
<td>Industrial Automation</td>
<td>SHA</td>
<td>Secure Hash Standard</td>
</tr>
<tr>
<td>ID</td>
<td>IDentifier</td>
<td>STD</td>
<td>STandarD</td>
</tr>
<tr>
<td>IDevID</td>
<td>Initial Device ID</td>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>IO</td>
<td>Input Output</td>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>IOC</td>
<td>IO Controller</td>
<td>TSN</td>
<td>Time-Sensitive Networking</td>
</tr>
<tr>
<td>IOD</td>
<td>IO Device</td>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
<td>UUID</td>
<td>Uniform Resource ID</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
<td>YANG</td>
<td>Yet Another Next Generation</td>
</tr>
</tbody>
</table>

Unrestricted | Siemens 2022 | 2022-03-07
Contacts

Kai Fischer, Siemens AG, T CST SES-DE, kai.fischer@siemens.com

Andreas Furch, Siemens AG, T CST SES-DE, andreas.furch@siemens.com

Oliver Pfaff, Siemens AG, DI FA CTR ICO PO, oliver.pfaff@siemens.com
Device Model: End Stations in IEC/IEEE 60802

Application*
- PLC program
- IO module
- Edge client
- Asset dashboard
- Engineering/aplication

Middleware*
- IEC 61158
- OPC-UA
- MQTT
- HTTP
- DIY

Network integration (TSN-IA)
- TCP/IP
- Ethernet layer
- IP layer
- 802.1AS
- 802.1AB
- 802.1CB
- 802.3
- 802.1Q

*: showing an exemplary/random suite of middleware/application modules
Device Model: Bridges in IEC/IEEE 60802

Network integration (TSN-IA)

Single bridge

Network interface

TCP/IP

Ethernet layer

802.1AS  802.1AB  802.1CB

802.3  802.1Q

IP layer

YANG

NETCONF server

Network configuration

DHCP

DNS

Addressing

Unrestricted | Siemens 2022 | 2022-03-07
Security Setup Sequence (1)

Own MAC and IP address
Own IDevID credential/trust anchor (not needed here)
Further IDevID trust anchors
Own LDevID-NETCONF credential/trust anchor
CA (CA, sub-CA, RA or proxy) and certificate repository feature

Unrestricted | Siemens 2022 | 2022-03-07

NETCONF-over-TLS setup: TO-BE-DONE

New IA station (without LDevID-NETCONF credentials) event*
MAC/IP address(es) of IA station

Happens once

New LDevID-NETCONF event

TLS: (provisional accept: own LDevID-NETCONF [CNC] vs. own IDevID [IA station])
NETCONF-over-TLS: supply LDevID-NETCONF trust anchor

NETCONF-over-TLS: get ietf-hardware info
Assess IDevID EE server cert and ietf-hardware incarnation: matching, expected?

Process in provisional accept mode with TOFU
Re-validate, terminate provisional accept mode with TOFU
LDelID-NETCONF trust anchor.

NETCONF/YANG server is defunct**

*: Not detailed herein
**: Does not comply to NETCONF/YANG security rules set forth by IETF RFCs 6241, 7589
Security Setup Sequence (2)

**CNC**
- Own MAC and IP address
- Own IDevID credential/trust anchor (not needed here)
- Further IDevID trust anchors
- Own LDevID-NETCONF credential/trust anchor

**IA station**
- Own MAC and IP address(es)*
- Own IDevID credential/trust anchor
- NACM objects (role-based)
- Own ietf-hardware incarnation
- LDevID-NETCONF trust anchor.

NETCONF/YANG server is defunct**

**NETCONF-over-TLS setup: TO-BE-DONE**

Happens once

TLS (almost plain vanilla: own LDevID-NETCONF [CNC] vs. own IDevID [IA station])

NETCONF-over-TLS: supply LDevID-NETCONF credential (misc. key-pair generation options)

Own LDevID-NETCONF credential

TLS (plain vanilla: own LDevID-NETCONF [CNC] vs. own LDevID-NETCONF [IA station])

NETCONF-over-TLS: supply LDevID-NETCONF certificate-to-name mapping

Certificate to name mapping

*: Not detailed herein

**: Does not comply to NETCONF/YANG security rules set forth by IETF RFCs 6241, 7589

CA (CA, sub-CA, RA or proxy) and certificate repository feature

TLS (plain vanilla: own LDevID-NETCONF [CNC] vs. own LDevID-NETCONF [IA station])

NETCONF-over-TLS: supply LDevID-NETCONF certificate-to-name mapping

Certificate to name mapping
Security Setup Sequence (3)

CNC
- Own MAC and IP address
- Own IDevID credential/trust anchor (not needed here)
- Further IDevID trust anchors
- Own LDevID-NETCONF credential/trust anchor

IA station
- Own MAC and IP address(es)*
- Own IDevID credential/trust anchor
- NACM objects (role-based)
- Own ietf-hardware incarnation
- LDevID-NETCONF trust anchor
- Own LDevID-NETCONF credential/trust anchor
- Certificate to name mapping

NETCONF/over-TLS setup: DONE

- New network configuration event
- TLS (plain vanilla: own LDevID-NETCONF [CNC] vs. own LDevID-NETCONF [IA station])
- NETCONF-over-TLS: supply/update/remove configuration data***

확안: Not detailed herein
**: Complies to NETCONF/YANG security rules set forth by IETF RFCs 6241, 7589
***: May encompass LDevID credentials/trust anchors (and further security settings) for middleware/applications

Unrestricted | Siemens 2022 | 2022-03-07
GeneralName Choices (IETF RFC 5280)

[0] otherName (asn1:OtherName)
[1] rfc822Name (asn1:IA5String)
[2] dDNSName (asn1:IA5String)
[3] x400Address (asn1:ORAddress)
[4] directoryName (asn1:Name)
[5] ediPartyName (asn1:EDIPartyName)
[6] uniformResourceIdentifier (asn1:IA5String)
[7] IPAddress (asn1:OCTET STRING)
[8] registeredID (asn1:OBJECT IDENTIFIER)