Background for ‘IEEE 802.1AR Adoption by IEC/IEEE 60802’

IEEE Plenary; Nov. 8, 2021

Oliver Pfaff

SIEMENS

DevID Solution

- The systems, protocols, and/or the policies and procedures that support the use of DevID-equipped devices in a customer network.
- Notes:
  - A DevID (*Device IDentifier*) is a means by which a device (an IA-station) can make claims about its own identity and prove such claims in interactions with communication partners
  - 802.1AR describes core elements of DevID solutions especially: DevID modules, DevID signature suites, IDevIDs/LDevIDs, IDevID/LDevID EE certificates, DevID trust anchor stores
  - 802.1AR does not deliver a full DevID solution. Several actions are needed to create a DevID solution including the following:

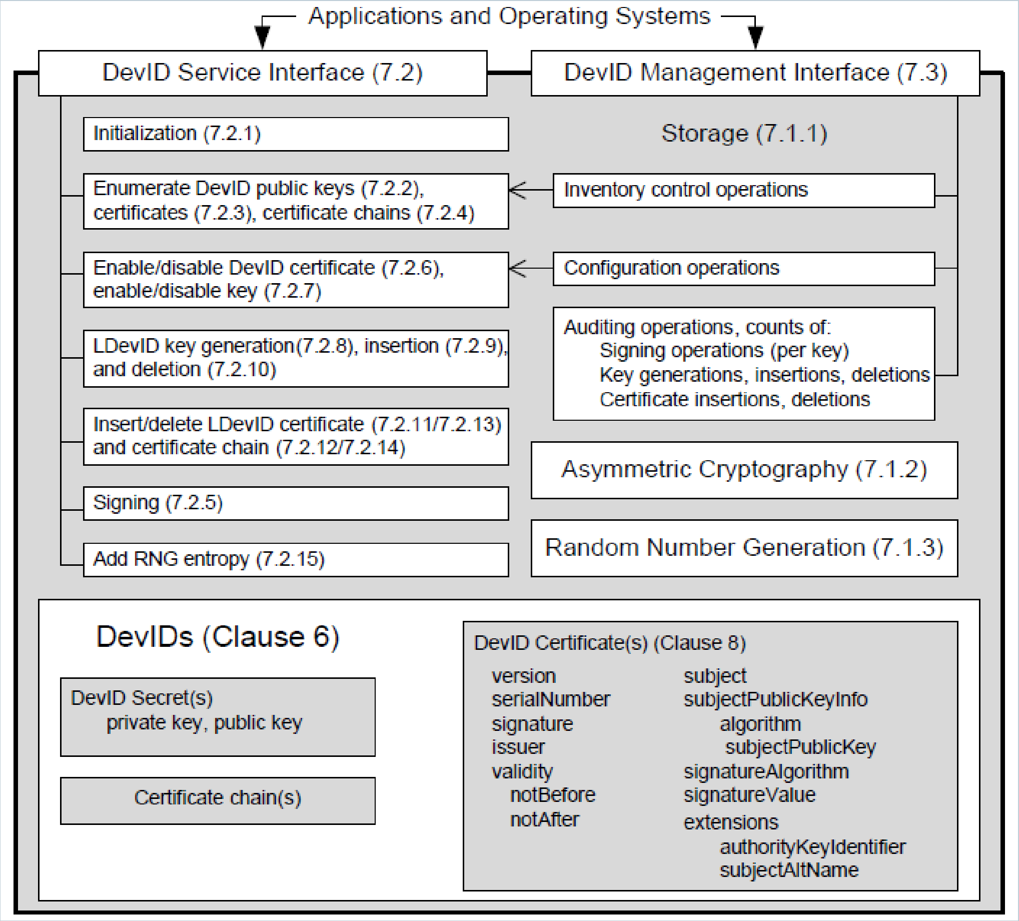
    i. Further detail 802.1AR-specified elements especially: DevID modules, DevID signature suites, IDevIDs/LDevIDs, IDevID/LDevID EE certificates

    ii. Incarnate 802.1AR-identified (but not defined) elements especially: DevID trust anchor stores

    iii. Define procedures/protocols to utilize IDevIDs/LDevIDs e.g. in NETCONF-over-TLS

    iv. Define procedures/protocols to create/manage IDevIDs/LDevIDs e.g. with NETCONF/YANG

DevID Module



Applications and Operating Systems

DevID Service Interface (7.2)

DevID Management Interface (7.3)

Initialization (7.2.1)

Enumerate DevID public keys (7.2.2), certificates (7.2.3), certificate chains (7.2.4)

Enable/disable DevID certificate (7.2.6), enable/disable key (7.2.7)

LDevID key generation(7.2.8), insertion (7.2.9), and deletion (7.2.10)

Insert/delete LDevID certificate (7.2.11/7.2.13) and certificate chain (7.2.12/7.2.14)

Signing (7.2.5)

Add RNG entropy (7.2.15)

Storage (7.1.1)

Inventory control operations

Configuration operations

Auditing operations, counts of:
Signing operations (per key)
Key generations, insertions, deletions
Certificate insertions, deletions

Asymmetric Cryptography (7.1.2)

Random Number Generation (7.1.3)

DevIDs (Clause 6)

DevID Secret(s)
private key, public key

Certificate chain(s)

DevID Certificate(s) (Clause 8)

version	subject
serialNumber	subjectPublicKeyInfo
signature	algorithm
issuer	subjectPublicKey
validity	signatureAlgorithm
notBefore	signatureValue
notAfter	extensions
	authorityKeyIdentifier
	subjectAltName

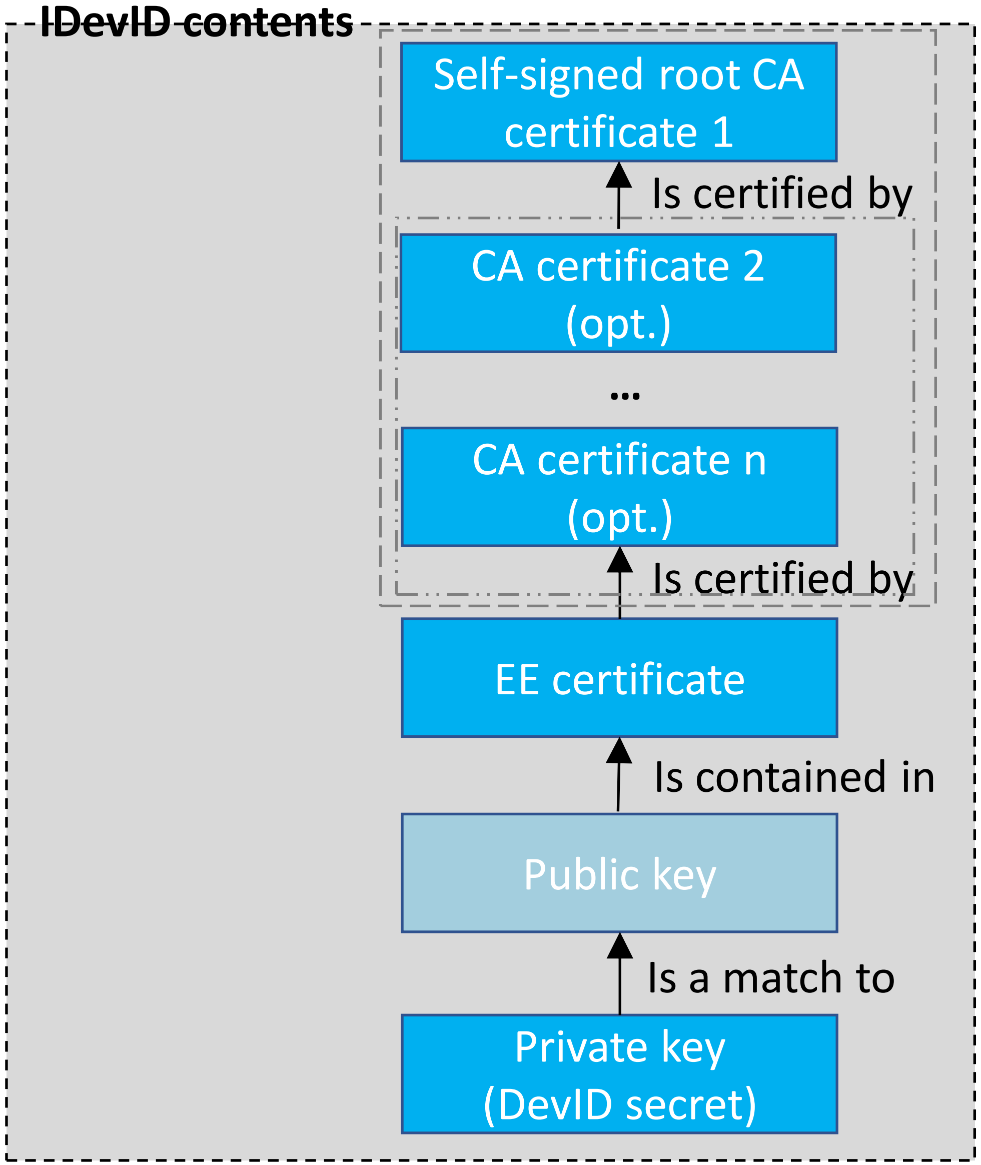
Single device

- Logical security component that:
  - Is a part of a device (an IA-station)
  - Stores and operates on the DevID(s) associated with a device
  - Presents a *cryptographic boundary:* DevID secrets (private keys) shall only be stored and used within this boundary

DevID Signature Suite

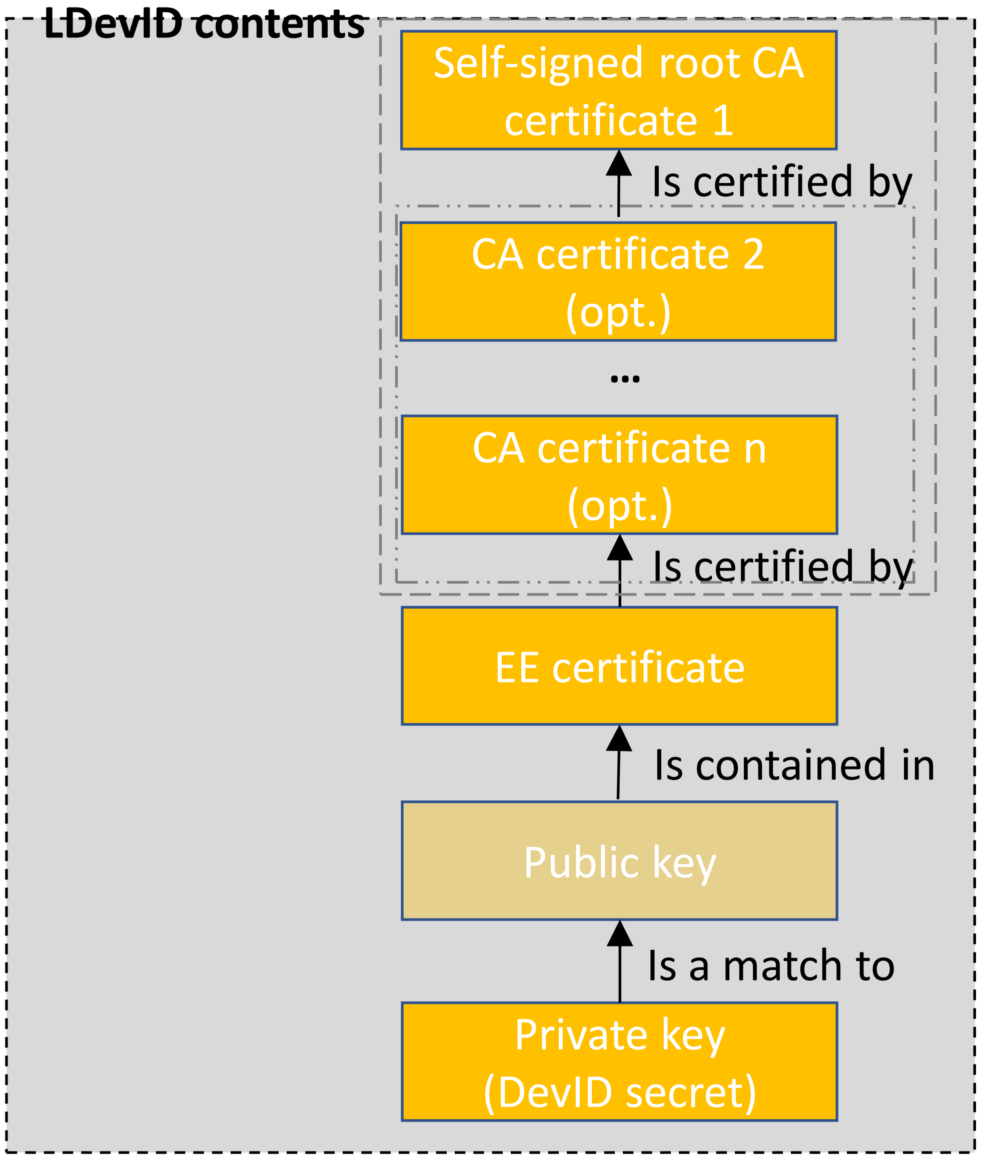
- Interoperable specification of cryptographic algorithms used for signing, parameter values used by those algorithms and representation in DER encoded certificate fields, in particular:
  - `signatureAlgorithm`: a struct to identify a signature algorithm and opt. provide parameters
  - `subjectPublicKeyInfo`: a struct to represent a public key value
  - `signatureValue`: a struct to represent a signature value (here: certificate signing)
- Following DevID signature suites are specified by 802.1AR:
  - `RSA-2048/SHA-256`
  - `ECDSA P-256/SHA-256`
  - `ECDSA P-384/SHA-384`
- *False friend*: DevID signature suites and TLS cipher suites are neither equal nor disjoint

IDevID



- A DevID that has 1..n incarnations per 1 device and where each incarnation
  - Is installed by the supplier of the device i.e. these identifiers are present in factory default state
  - Comprises a private key (called DevID secret), a corresponding EE certificate (X.509v3) and its certificate chain
  - Serves the identification/authentication of the device against others. Terms and conditions apply:
    - Information contained in IDevID EE certificates is limited to information that is known by the time of IDevID creation; this does not comprise deployment-specific items e.g. names/properties and addresses
    - This identifier does not cover the checking of identification/authentication by others

LDevID



- A DevID that has 0..n incarnations per 1 device and where each incarnation
  - Is installed by the user of the device i.e. these identifiers are not present in factory default state
  - Has the same information model (left) as an IDevID - but with other value/object instances
  - Serves the same purpose as an IDevID – but with a major change in terms and conditions:
    - ~~Information contained in IDevID EE certificates is limited to information that is known by the time of IDevID creation; this does not comprise deployment-specific items e.g. names/properties and addresses~~ 🡐 *this limitation is resolved by LDevIDs (which is fundamental for operational use in e.g. NETCONF-over-TLS)*
    - This identifier does not cover the checking of identification/authentication by others

IDevID EE Certificate

X.509 certificate fields	802.1AR usage
`version`	Fixed value (v3)
`serialNumber`	Arbitrary positive integer (at most 20 octets)
`signature`	Single value according DevID signature suites
`issuer`	Arbitrary X.500 name, must match subject name in the CA certificate containing the public key corresponding to the private key used to sign the certificate
`validity`	Should be eternal (notAfter=99991231235959Z); notBefore (time of DevID generation) is arbitrary
`subject`	Arbitrary X.500 name, must contain serialNumber attribute with a device serial number (value must uniquely identify a device in the issuer‘s domain of significance)
`subjectPublicKeyinfo`	Arbitrary value according DevID signature suites
`extensions`	• authorityKeyIdentifier: arbitrary value identifying the public key corresponding to the private key used to sign the certificate • subjectAltName (opt): should include a HardwareModuleName (RFC 4108) that provides additional information about the device • keyUsage (opt): shall include digitalSignature, may include keyEncipherment
`signatureAlgorithm`	Single value according DevID signature suites
`signatureValue`	Arbitrary value according `signatureAlgorithm`

LDevID EE Certificate

X.509 certificate fields	802.1AR usage
`version`	Fixed value (v3)
`serialNumber`	Arbitrary positive integer (at most 20 octets)
`signature`	Single value according DevID signature suites
`issuer`	Arbitrary X.500 name, must match subject name in the CA certificate containing the public key corresponding to the private key used to sign the certificate
`validity`	Arbitrary value tuple with notBefore (time of DevID generation) < notAfter
`subject`	Arbitrary X.500 name (can be empty)
`subjectPublicKeyinfo`	Arbitrary value according DevID signature suites
`extensions`	• authorityKeyIdentifier: arbitrary value identifying the public key corresponding to the private key used to sign the certificate • subjectAltName (opt): should include a HardwareModuleName (RFC 4108) that provides additional information about the device
`signatureAlgorithm`	Single value according DevID signature suites
`signatureValue`	Arbitrary value according `signatureAlgorithm`

DevID Trust Anchor Store

- The database of trust anchor information for IDevIDs and LDevIDs that is stored and used by a DevID solution
  - IDevID trust anchor(s):
    - Serve the checking of IDevID-based identification/authentication by other devices/components
    - Are installed by the supplier of the device i.e. these trust anchors are present in factory default state
  - LDevID trust anchor(s):
    - Serves the checking of LDevID-based identification/authentication by other devices/components
    - Are installed by the user of the device i.e. these trust anchors are not present in factory default state
- Notes:
  - Trust anchor (IEEE 802.1AR): A CA that is trusted and for which the trusting party holds information, usually in the form of a self-signed certificate issued by this CA
  - Trust anchor (RFC 5280): A CA certificate that serves as a trust anchor for the certification path validation

Abbreviations

CA	Certificate Authority
DevID	Device Identifier
DER	Distinguished Encoding Rules
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve DSA
EE	End Entity
IDevID	Initial DevID
LDevID	Locally significant DevID
NETCONF	NETwork CONFiguration
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
YANG	Yet Another Next Generation

Contact

Oliver Pfaff, Siemens AG, DI FA CTR ICO PO, oliver.pfaff@siemens.com