Security Use Cases IEC/IEEE 60802

Contributor group

Fischer, Kai <kai.fischer@siemens.com>
Furch, Andreas <andreas.furch@siemens.com>
Pfaff, Oliver <oliver.pfaff@siemens.com>
Pössler, Thomas <thomas.poessler@siemens.com>
Steindl, Günter <guenter.steindl@siemens.com>

Abstract

This document identifies security use cases that apply to IEC/IEEE 60802 'TSN Profile for Industrial Automation' and shall be addressed by its security profile.
The security use cases are complementary to the industrial use cases that apply to IEC/IEEE 60802 'TSN Profile for Industrial Automation' (see [15]). The union of the industrial use cases and security use cases are meant to be applicable to production sites/cells that are operated based on TSN-IA and that employ security.

Note: this version is a quick first write-up based on the identification of security use cases discussed in the IEC/IEEE 60802 working group on 2021-04-26. The goal is to kick-off discussion in the ad-hoc meeting 2021-04-30. ***Further elaboration (content, illustrations, form, and structure) is needed. The markup 'TODO' is used to identify open action items***

Log

v0.1	2021-04-30	Initial draft

Content

Figures

References

Listed in chronological order:

1. Stajano, F.; Anderson, R: The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks, 1999
2. IETF RFC 4949: Internet Security Glossary, Version 2, 2007
3. IETF RFC 5116: An Interface and Algorithms for Authenticated Encryption, 2008
4. IEEE 802.1X-2010: IEEE Standard for Local and Metropolitan Area Networks – Port-Based Network Access Control, 2010
5. IETF RFC 6125: Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS), 2011
6. IEC 62443-3-3: System Security Requirements and Security Levels, 2015
7. IETF RFC 7696 Guidelines for Cryptographic Algorithm Agility and Selecting Mandatory-to-Implement Algorithms, 2015
8. NIST SP 800-90A: Recommendation for Random Number Generation Using Deterministic Random Bit Generators, 2015
9. NIST SP 800-90C Recommendation for Random Bit Generator (RBG) Constructions, 2016
10. NIST SP 800-63-3: Digital Identity Guidelines, 2017
11. NIST SP 800-63A: Digital Identity Guidelines - Enrollment and Identity Proofing, 2017
12. IEEE 802.1AE-2018: IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Security – Revision D 1.3, 2018
13. IEEE 802.1AR-2018: IEEE Standard for Local and Metropolitan Area Networks–Secure Device Identity, 2018
14. IEEE 802.3-2018: IEEE Standard for Ethernet, 2018
15. IEC/IEEE 60802: Use Cases IEC/IEEE 60802 V1.3, Draft (work-in-progress), 2018
16. IETF RFC 8366: A Voucher Artifact for Bootstrapping Protocols. 2018
17. IETF RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3, 2018
18. NIST SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, 2018
19. IETF RFC 8572: Secure Zero Touch Provisioning (SZTP), 2019
20. IETF RFC 8576: Internet of Things (IoT) Security: State of the Art and Challenges, 2019
21. IEC 62443-4-2: Technical Security Requirements for IACS Components, 2019
22. IEC/IEEE 60802: Time-Sensitive Networking Profile for Industrial Automation, Draft 1.2, 2020
23. IETF BRSKI: Bootstrapping Remote Secure Key Infrastructures (BRSKI), Draft 45 (work-in-progress), 2020

Abbreviations

AEAD	Authenticated Encryption with Added Data
BRSKI	Bootstrapping Remote Security Key Infrastructure

CA	Certification Authority
CMS	Cryptographic Message Syntax
CoAP	Constrained Application Protocol
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Service
DTLS	Datagram Transport Layer Security
EE	End Entity
FW	FirmWare
HTTP	Hypertext Transfer Protocol
HW	HardWare
IA	Industrial Automation
ID	IDentifier
IDevID	Initial Device ID
IoT	Internet of Things
IT	Information Technology
LDevID	Locally significant Device ID
MASA	Manufacturer Authorized Signing Authority
NETCONF	NETwork CONFiguration
OT	Operational Technology
PRNG	Pseudo Random Number Generator
RNG	Random Number Generator
SW	SoftWare
SZTP	Secure Zero Touch Provisioning
TLS	Transport Layer Security
TOFU	Trust On First Use
TRNG	True Random Number Generator
TSN	Time-Sensitive Networking

1. Use case 01: Checking the equipment under control

1.1 General

This security use case serves to motivate and explain the checking (actual vs. expected) of IA components as prerequisite before imprinting keys/credentials for security and using the IA component.

1.2. Useful resources/mechanisms

- See [10] and [11] for an elaboration of this concern for the case of human beings (not IA components)

TODO: further elaboration

2. Use case 02: Imprinting during bootstrapping/commissioning

2.1 General

This security use case serves to motivate and explain the supply production site/cell-specific keys and credentials the first time. Following aspects are considered:

- **Taking possession**: an owner/operator obtains new equipment (IA device/controller) and wants to equip it with keys/credentials that are specific for the production site or cell – using engineering (or similar) tools
- **Device replacement without engineering** (use case 35 in [15]): an owner/operator wants to (ad-hoc) replace a broken IA device and needs to equip the replacement IA device with keys/credentials that are specific for the production site or cell – without using engineering (or similar) tools
- **Modular machine assembly** (use case 19 in [15]): an owner/operator wants to (ad-hoc) re-use a priorly deployed IA device in another machine and needs to equip the re-used IA device with keys/credentials that are specific for the new production cell – without using engineering (or similar) tools

2.2 Constraints

- IA components without periphery/interfaces for human user interaction (display, keyboard…)
- IA components with/without IDevID
- TODO: briefly introduce IDevID/LDevID (probably in an Annex)

2.3 Actor concerns

- *Manufacturers*: design of IA components, selection of sub-modules as well as underlying standards, supply of IDevIDs
- *Distributors*: originality checking (e.g., using IDevID) of random sample
- *Machine builder*: equipment originality checking (e.g., using IDevID)
- *System integrator*: equipment originality checking (e.g., using IDevID), supply of production site/cell-specific keys and credentials (e.g., LDevID)
- *Owner/operator*: originality checking (e.g., IDevID or LDevID), supply of production site/cell-specific keys and credentials (e.g., LDevID*)

2.4 Lifecycle concerns – IA component

- *Manufacturing*: optionally equip IA component with IDevID(s)
- *Bootstrapping*: supply LDevIDs or other production site/cell-specific keys/credentials by protected credential management tasks
- *Operating*: establish/update[1] long/short-lived security association keys, use them for protecting actual payload exchanges e.g., network configuration, stream establishment
- *Maintaining*: update[2] LDevIDs or other production site/cell-specific keys/credentials, use protection for these key and credential management tasks
- *Terminating*: erase[3] LDevIDs or other production site/cell-specific keys/credentials (but keep IDevID), use protection for these key and credential management tasks

2.5 Useful resources/mechanisms

See Annex A for a description of useful mechanisms.

2.6 Functional aspects

2.6.1 Taking possession

The fitness assessment for the described mechanisms is:

- IETF BRSKI: if LDevIDs (CA certificates and EE certificates) shall be supplied and the IA components and production site/cell meet the above identified properties
- IETF SZTP: if LDevIDs CA certificates shall be supplied and the IA components and production site/cell meet the above identified properties

2.6.2 Device replacement without engineering

The fitness assessment for the described mechanisms is:

- IETF BRSKI: as for taking possession above and modulo: an additional trigger for security bootstrapping than factory reset is needed

---

[1] Updating is part of another use case and mentioned here to reflect an overall perspective

[2] Updating is part of another use case and mentioned here to reflect an overall perspective

[3] Erasing is part of another use case and mentioned here to reflect an overall perspective

- IETF SZTP: as for taking possession above and modulo: an additional trigger for security bootstrapping than factory reset is needed

2.6.3 Modular machine assembly

The fitness assessment for the described mechanisms is:

- IETF BRSKI: as for taking possession above and modulo: an additional trigger for security bootstrapping than factory reset is needed
- IETF SZTP: as for taking possession above and modulo: an additional trigger for security bootstrapping than factory reset is needed

3. Use case 03: Instructing equipment about security

3.1 General

This security use case serves to motivate/explain the instructions about security that apply to IA components.

3.2 Functional aspects

- Per owner/operator entity (all equipment in a site/cell): security-only by default
- Per individual IA component: security always-on or on/off, enabled cryptographic algorithms…
- Per application/communication relation between IA components: security on/off, authentication-only/authenticated encryption, instance of protection algorithm

TODO: further elaboration

4. Use case 04: Peer entity authentication

4.1 General

This security use case serves to motivate/explain (peer) entity authentication and its inherited features, e.g., authenticated key agreement or authorization. The consideration of (peer) entity authentication also includes checking actual vs. expected.

4.2 Useful resources/mechanisms

- See [5] for an elaboration of checking actual vs. expected (as part of peer entity authentication) in IT

TODO: further elaboration

5. Use case 05: Message exchange protection

This security use case serves to motivate/explain the protection of communications between stations including its prerequisites, in particular peer entity authentication (including identification).
Note: this security use case overlaps with use case 30 in [15] (“Security”). It is proposed to serve as a candidate to transfer the content of this use case from the industrial use case document [15] to an emerging security use case document.

TODO: further elaboration

6. Use case 06: Proving self-asserted information

This security use case serves to motivate and explain bindings between peer entity authentication and self-asserted information e.g., topology discovery data or identification and maintenance data.

TODO: further elaboration

7. Use case 07: Resource access authorization

This security use case serves to motivate/explain access control and its prerequisites, in particular peer entity authentication (including identification).

TODO: further elaboration

8. Use case 08: Credential/key update during operation

This security use case serves to motivate/explain the handling of key aging in the case of planned/scheduled expiry, considers/establishes bumpless-ness.

TODO: further elaboration

9. Use case 09: Credential/key revocation/invalidation during operation

This security use case serves to motivate/explain the handling of premature termination of key/credential lifetime.

TODO: further elaboration

10. Use case 10: Crypto algorithm-expiry/agility

10.1 General

This security use case serves to motivate/explain the handling of the dawn of crypto algorithms (symmetric or asymmetric).

10.2 Useful resources/mechanisms

- See [7] for IETF guidelines for cryptographic algorithm agility

TODO: further elaboration

11. Use case 11: Robust supply of security core function

11.1 General

This security use case serves to identify/explain foundational principles of security – when using cryptography. Following aspects are considered:

- **Authenticated encryption** (AEAD) vs. classical schemes (first-sign-then-encrypt or first-encrypt-then-sign; sidenote: encrypt-only is no safe harbor)
- **Key protection**
- **Randomness** for symmetric and asymmetric keys, nonces
- **Dedicated HW** for accelerating cryptographic operations and protecting keys/credentials especially long-lived ones

11.2 Constraints

- IA components without periphery/interfaces for human user interaction (display, keyboard…)
- IA components with computational limitations (memory, processor etc.)
- IA components with/without (good) sources of entropy

11.3 Actor concerns

- *Manufacturers*: design of IA components, selection of sub-modules as well as underlying standards
- *Distributors*: n.a.
- *Machine builder*: selection of machine modules

- *System integrator*: selection of IA components/machines
- *Owner/operator*: selection/operation of IA components/machines

11.4 Lifecycle concerns – IA component

- *Manufacturing*: exercise good security practices during product definition and development phases, optionally equip IA component with IDevID(s)
- *Bootstrapping*: supply LDevIDs (when using X.509 public key certificate credentials) or other production site/cell-specific keys/credentials, use protection for key and credential management
- *Operating*: establish/update long/short-lived security association keys, use protection for actual exchanges (network configuration, stream establishment)
- *Maintaining*: update LDevIDs (when using X.509 public key certificate credentials) or other production site/cell-specific keys/credentials, use protection for key and credential management
- *Terminating*: erase LDevIDs (when using X.509 public key certificate credentials) or other production site/cell-specific keys/credentials (keep IDevID), use protection for key and credential management

11.5 Useful resources/mechanisms

- See [3] for an interface and algorithms for authenticated encryption
- See [8], [9], and [18] for NIST recommendations on random number generation

11.6 Functional aspects

11.6.1 Authenticated encryption

TODO: further elaboration

11.6.2 Key protection

Cryptographic keys and their management must be protected – no matter which cryptographic scheme (symmetric or asymmetric) is used. This is rooted in the fundamental pattern behind cryptographic security:

- The to-be-protected input is transformed into a protected output (symmetric and asymmetric schemes)
- In contrast to data, the key remains unaltered by this transformation and must be protected (symmetric and asymmetric keys)

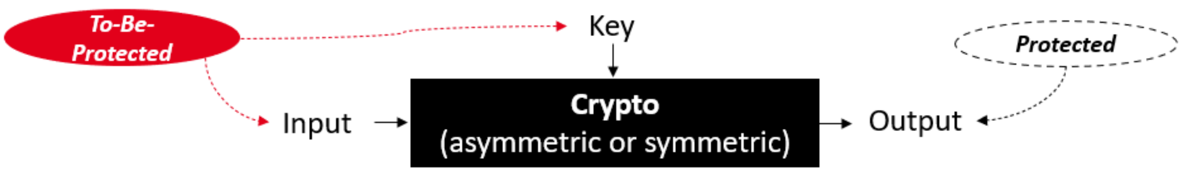

Figure 1: Pattern behind cryptographic security

Critical items:

- Symmetric and private keys: authenticity and confidentiality of the key
- Public keys and trust anchors (CA certificates): authenticity of the key resp. CA certificate

Pitfalls (*deciphering the basics behind public key infrastructure*):

- Signatures in self-signed (CA) certificates only provide the integrity of the certificate object, but no authenticity
- Signatures in EE certificates provide the authenticity of the certificate object, but no authentication for a claimant who/which is presenting this object
  - Note: this document assumes EE certificates to be never self-signed

11.6.3 Randomness

TODO: further elaboration

11.6.4 Dedicated HW

TODO: further elaboration

Annex AKey and credential imprinting models and procedures

A.1 General

This annex digests important models and procedures for imprinting keys and credentials that are applicable to industrial components.

A.2 The Resurrecting Duckling

TODO: digest/elaborate on [1], a foundational model introducing TOFU

A.3 IEEE 802.1AR Device Identity

TODO: digest/elaborate on [13], a foundational model introducing IDevID/LDevID

A.4 IETF BRSKI

A.4.1 General

An IETF procedure (see [23] and [16] for protocol exchanges between system components) that allows to incarnate the imprinting model established in IEEE 802.1AR Device Identity.

A.4.2 Synopsis

IETF BRSKI cares about *LDevID-by-IDevID supply* to IoT/OT components - without touching them. This comprises (see [23]):

- LDevID CA certificate-by-voucher supply via "PROVISIONAL accept of server cert". The voucher object is a manufacturer-signed container comprising the LDevID CA certificate (local trust anchor). The voucher object is specified in [16]. By consuming a voucher (after its successful validation), a new trust anchor (LDevID CA certificate) is imprinted to the IoT/OT component
- LDevID EE certificate-by-IDevID supply via plain-vanilla (D)TLS. BRSKI supports key pair generation internally on the IoT/OT component as well as external key pair generation. The decision which key pair generation mode is used is allocated with the IoT/OT component.

A.4.3 Digest

IETF BRSKI mandates IoT/OT components to be equipped by an IDevID and describes the LDevID-by-IDevID trick on the base of 2 actors, 4 system components and 4 exchanges:

- *Actors*:
    1. Owner/operator (running a production site or cell, called "domain" in IETF BRSKI)
    2. Manufacturer (of the IoT/OT component)
- *System components*:
    1. IoT/OT component (called "pledge" in IETF BRSKI, an IA device/controller or station in TSN-IA terms), aiming to join a production site or cell esp. its applications (1)
    2. Join proxy, a part of the production site or cell (1)
    3. Registrar, a part of the production site or cell (1)
    4. MASA (Manufacturer Authorized Signing Service), a service of the manufacturer of the IoT/OT component (1)
- *Exchanges*:
    1. Voucher exchange: imprinting the LDevID CA certificate to the IoT/OT component (using voucher objects and "PROVISIONAL accept of server cert")
    2. Voucher status exchange: okay/not okay feedback for the voucher exchange by the IoT/OT component
    3. Enrollment exchange: imprinting a LDevID EE certificate to the IoT/OT component
    4. Enrollment status exchange: okay/not okay feedback for the enrollment exchange by the IoT/OT component

Note: the (default/specified) trigger for performing these exchanges is: IoT/OT components starts (power-on) in factory default state

The IETF BRSKI actors, system components and exchanges are digested in following figure:

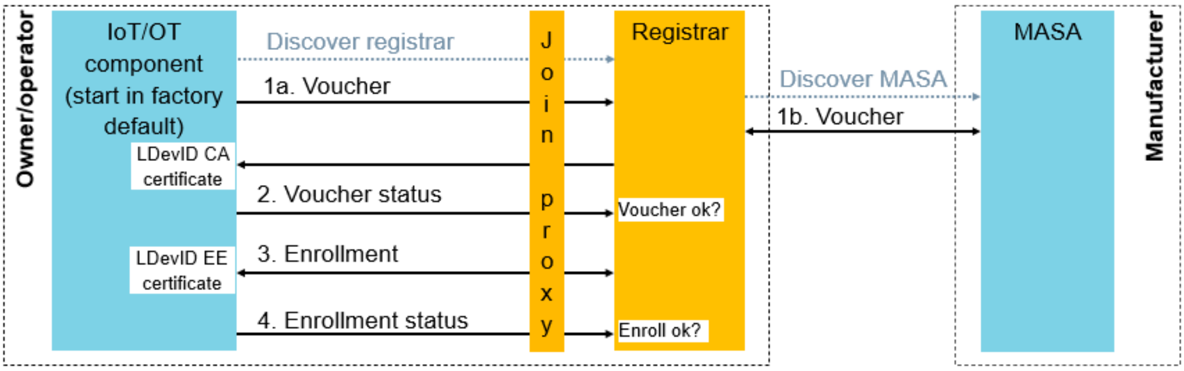

Figure 2: IETF BRSKI actors, components and exchanges

A.4.4 Properties

IETF BRSKI comes with several assumptions and properties. The most relevant ones are:

- Is not specifically geared towards NETCONF
- Requires IoT/OT components to possess IDevIDs. Assumes IDevID EE certificates to contain a MASA URI extension
- Supports arbitrary LDevID EE and CA certificate contents
- Demands production site/cell to run a service (“Registrar”) that conducts the joining process and acquires (from manufacturer through its MASA) and supplies voucher objects to imprint LDevID CA certificates as (new) trust anchors to IoT/OT components
- Covers the pull model (IA components need to proactively apply at the registrar for their joining). Does not cover the push model.
- Uses HTTP(-over-TLS) and/or CoAP(-over-DTLS) for its exchanges:
    - IoT/OT component/registrar exchanges: HTTP-over-TLS or CoAP-over-DTLS
    - Registrar/MASA exchanges: HTTP-over-TLS
- Extends the functional behavior of (D)TLS by “PROVISIONAL accept of server cert”
- Requires manufacturer to run a service (“MASA”) that issue objects (vouchers) that are specific to instances of IA components (this assumption is relaxed by draft-richardson-anima-voucher-delegation-03, work-in-progress)
- Allows arbitrary strategies for deciding about joining requests (at production site/cell; represented by the registrar) in order to match actual vs. expected
- Allows arbitrary strategies for deciding about voucher issuance/assignment (manufacturer; represented by the MASA)
- Can support ownership tracking (manufacturer)

A.5 IETF SZTP

A.5.1 General

Another IETF procedure (see [19] and [16] for protocol exchanges between system components) that allows to incarnate parts of the imprinting model established in IEEE 802.1AR Device Identity. IETF SZTP emerged from the NETCONF WG and is a native building block of NETCONF security.

A.5.2 Synopsis

IETF SZTP cares about *LDevID CA certificate-by-initial key/credential e.g., IDevID supply* to IoT/OT components - without touching it.
IETF SZTP is not limited to an imprinting of LDevID CA certificate and can also supply:

- Boot image
- Configuration information

This supply can happen in protected (or plain) way. The protection of SZTP bootstrap data employs object security, uses basic object security means e.g., CMS. This protection is profiled by SZTP.
IETF SZTP does not cover *LDevID EE certificate-by-initial key/credential e.g., IDevID supply*

A.5.3 Digest

IETF SZTP requires IoT/OT components to possess initial keys/credentials, that were established before SZTP exchanges take place (this may be IDevIDs). SZTP uses 2 actors, 2 system components and 2 exchanges:

- *Actors*:
  1. Owner/operator (running a production site or cell)
  2. Manufacturer (of the IoT/OT component)
- *System components*:
  1. IoT/OT component (called “device” in IETF SZTP, an IA device/controller or station in TSN-IA terms), aiming to join a production site or cell esp. its applications (1)
  2. SZTP Bootstrap server or DHCP/DNS server supplying SZTP artifacts (0..n)[4].
     -
- *Exchanges*:
  1. Redirect exchange: imprinting the LDevID CA certificate to the IoT/OT component (using voucher objects, not employing “PROVISIONAL accept of server cert”)
  2. Bootstrap exchange: supplying boot image, configuration info (protected on object-level based on keys/credentials that can be deprotected based on the trust anchor and key material established in the redirect exchange)

  Note: the (default/specified) trigger for performing these exchanges is: IoT/OT components starts (power-on) in factory default state

The IETF SZTP actors, system components and exchanges are digested in following figure:
TODO

A.5.4 Properties

IETF SZTP comes with several assumptions and properties. The most relevant ones are:

- Emerged from the NETCONF WG and is specifically geared towards NETCONF
- Requires IoT/OT components to possess initial keys/credentials, that were established before SZTP exchanges take place (this may be an IDevID)
- Supports arbitrary LDevID CA certificate contents (does not cover LDevID EE certificates)
- Demands production site/cell to run a SZTP service (“Bootstrap server”). Alternatively, a production site/cell may supply SZTP artifacts through DHCP/DNS servers. Using removable storage is another option for supplying SZTP artifacts.
- Covers the pull model (IA components need to proactively apply at the registrar for their joining). Also covers the push model (removable storage) but demands physical access for push.
- Uses HTTP(-over-TLS) or DHCP/DNS for its exchanges

[4] Removable storage also allows to do SZTP bootstrapping with no SZTP Bootstrap server and no DHCP/DNS server supplying SZTP artifacts

- Allows (not: requires) manufacturer to run a service (“Bootstrap server”) that issues SZTP artifacts. Demands manufacturers to issue voucher objects but does not demand manufacturer services such as MASA.