Automated Onboarding and Device Provisioning Best Practices

An Industry IoT Consortium
Best Practices Paper

2022-10-04

Authors

John Fornehed (Ericsson), Keao Caindec (Farallon), David Meier (Fraunhofer), Xinxin Fan (IoTeX), Mitch Tseng (Tseng InfoServ), Ben Smeets (Ericsson), Ilhan Gurel (Ericsson), Paul Bradley (Arm), Frederick Hirsch (Upham Security), Jeff Shiner (Micron), Hanu Kommalapti (Microsoft), Alex B. Ferraro (PwC).

Contents

FIGURES

Endpoint security protection is a challenging area for the industrial internet of things (IIoT). Devices deployed in the field may not have the necessary computing power or hardware, compared to their counterparts sitting in managed central offices. Any piece of equipment added to the network should always be vetted when onboarded. This is generally done through device or equipment provisioning when new devices are installed.

IoT services require end-to-end security protection to mitigate risks and defend against malicious attacks. This includes the IP-security protections on the servers and networks and protection for IoT devices. This has to be done as early as possible, preferably at the device-provisioning stage while bringing devices onboard. Contrary to their physically protected counterparts at the server end, IoT devices are deployed on the remote edge of the service area and are more vulnerable, due to their visibility, limited hardware and software capabilities. Nevertheless, with advances in chip manufacturing, security measures can be built-in at the chip level, as well as through other software and identity-management processes that enable the device protection as early as the device-onboarding stage.

Thanks to cheaper computing and memory, adding security protection for edge devices at the onboarding stage is possible before designing and producing the devices. There are multiple means for device onboarding security and the end users can choose the most cost-effective way according to their budget. Hardware-based root-of-trust, software-based root-of-trust, flash-based (firmware) identity management approaches, security certificates and key encryption techniques can all be used for enhancing device security but must be considered holistically. Security-by-design is the best and most straightforward approach to secure a system because security is embedded from the beginning of the concept, architecture and design process.

This whitepaper provides guidance on the security of device deployment and installation, a necessary part of security-by-design. It also provides device-oriented considerations for secure and cost-effective end-to-end service solutions and is intended for system integrators, service providers and those who implement IIoT solutions. It provides an overview of the process, practices and standards related to the security of device onboarding, allowing all stakeholders to understand the concepts and concerns. It begins with the role device provisioning and onboarding should play in the overall device lifecycle. The various aspects of onboarding, such as the pain points or challenges, opportunities and situation analysis, are addressed to help understand best practices in the industry and related standards. Then we discuss key technologies related to provisioning and onboarding, followed by the importance of end-to-end chaining of trust by ID federation.

The Industry IoT Consortium (IIC) has published guidelines for IIoT security, moving from basic security protection to broader trustworthiness for IIoT services, such as protecting supply-chain integrity. Other IIC security publications help end users review their overall security protection objectives and determine the appropriate investment for their security needs. In addition to the

IIC Security Framework (IISF[1]), the IIC Trustworthiness Foundation[2] and many security-related whitepapers, IIC has developed an IoT Security Maturity Model[3] that assists stakeholders in determining the appropriate level of security protection needed for their system and provides for assessments against the goal, enabling continuous improvement.

There are other security guidelines for security that can be relevant in an IIoT setting, such as zero-trust security (NIST SP 800-207)[4] or guidelines and requirements for banking and credit card apps on smartphones. This document focuses on endpoint onboarding security.

1 DEVICE LIFECYCLE AND ONBOARDING

A key challenge in industrial IoT is the secure and automated onboarding and provisioning of devices *en masse*.

Device onboarding is the process by which a device is recognized as a member of a set of devices (e.g. one of "my devices") in a target system. Such a system consists of a platform and a backend running in a device management system (a management system that could be implemented in a private, public or hybrid data center, or even on an industrial gateway). When a device is connected to its target device-management system, when a chain of trust is built through steps from power-on to when the rightful owner is established and it is determined who can control and use the device.

The existing approaches for on boarding are often manual, slow and insecure. The automated solutions that do exist in the industry are often not interoperable. The IoT industry would benefit from secure onboarding and provisioning solutions based on open standards, and the ability to offer secure onboarding across different hardware and software platforms. This is especially important when moving from a proof-of-concept towards a scalable and widely adopted solution.

Automation of onboarding plays an important role, and a federation of bootstrapping (see Chapter 7, Page 16) also needs automation as much as possible because of security, scalability, regulatory compliance checking (e.g. legislation in California,[5] EU[6] and Japan[7]), low operational costs and human mistakes.

Issues around the lack of automation include:

- human error during device configuration,
- trust—you may not trust the installer with credentials,

[1] *https://www.iiconsortium.org/IISF/*
[2] *https://www.iiconsortium.org/pdf/Trustworthiness_Framework_Foundations.pdf*
[3] *https://www.iiconsortium.org/stay-informed/SMM/*
[4] *https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-207.pdf*
[5] The California IoT cybersecurity law SB-327
[6] Cyber Security for Consumer Internet of Things ETSI TS 103 645
[7] METI, IoT security and safety framework

- speed—even semi-automatic onboarding can require 20 minutes per device (Gartner),
- cost—reduces return-on-investment and
- skills—automation requires less skill from the installer.

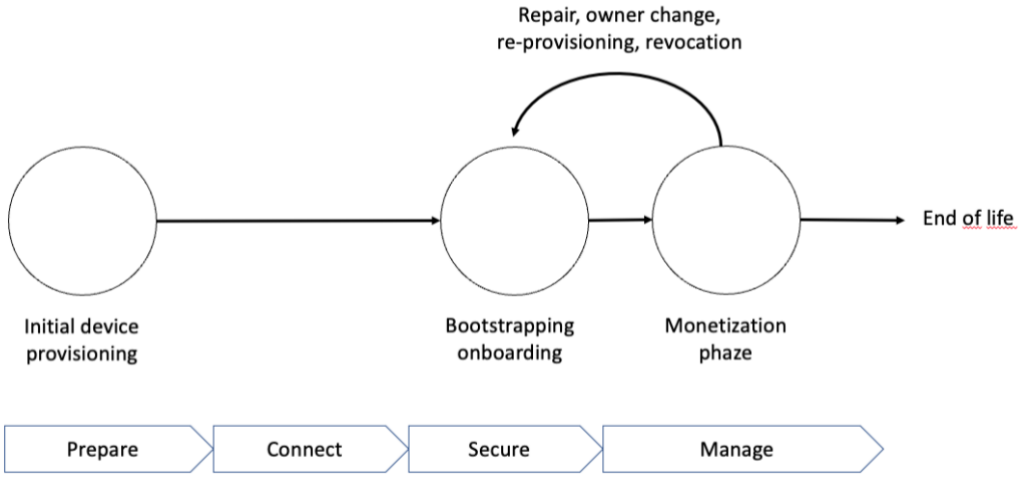

Figure 1-1: Simplified device lifecycle.

Automated solutions are largely non-interoperable, driving costs up on integration.

Open standards drive interoperability by lowering the cost, enabling scale, creating an open market without lock-in, enabling selection of cloud provider and device management partner.

A provisioning approach should also be agnostic to the communications technology, again to allow interoperability.

2 Pain Points (Challenges) & Opportunities, Situation Analysis

2.1 Challenges of Device Trustworthiness and Onboarding

There are many challenges to establishing a trustworthy identity when manufacturing a device.

Endpoint identity is based on the inherent properties of an endpoint that distinguishes it from other endpoints. Endpoint identity needs to be supported with credentials. The endpoint identity is based on the endpoint root-of-trust.

Endpoint root-of-trust is the foundation to secure other functions at the endpoint, from the hardware to applications including firmware, virtualization layer, operating system, execution environment and application. It also provides confidence on the endpoint identity.

Endpoint protection relies on establishing an endpoint root-of-trust (RoT), a unique identity that can solely attest to its own authenticity. The RoT determines the confidence in the system and its identity and ensures integrity and access control to its resources. An endpoint RoT must be established to secure other functions at the endpoint, from the hardware to applications

including firmware, virtualization layer, operating system, execution environment and application. It also provides confidence on the endpoint identity.

Establishing a trustworthy identity during the manufacturing process has key generation and storage challenges:

- generating a unique identity with suitable entropy that meets industry standards such as NIAP and NIST 800-63,
- generating an identity securely, for example within a tamper-resistant environment such as a Trusted Platform Module (TPM), Hardware Security Module (HSM) or other secure elements,
- storing the keys securely within a hardware-protected environment and
- establishing a trusted relationship between the manufacturer or vendor and the operator.

There are also provisioning and automation challenges:

- provisioning the identity management infrastructure,
- inserting the unique device identity into the device,
- supporting different system architectures and IoT compute, memory and power constraints,
- provisioning a software client on the device to automate identity management,
- provisioning Public Key Infrastructure (PKI) management systems (public/private key generation, PKI management, key storage, certificate authority, etc.) and
- automating certificate signing.

In identity lifecycle management we have managing the lifecycle of IoT machine identities, keys and certificates for:

- update,
- revocation,
- rotation,
- transfer of ownership and
- end-of-life.

In integration with cryptographic operations, we must integrate the endpoint identity and RoT into other cryptographic operations for:

- authentication,
- encryption and
- firmware updates.

And finally, meeting compliance standards for:

- NIAP—entropy and key generation standards and
- NIST 800-63 Authentication Assurance Level.

3 BEST PRACTICES USED IN THE INDUSTRY (STANDARDS)

There exist several secure provisioning concepts and standards for IoT devices, including standards built on top of other standards. Key concepts related to the standards are:

- The late-binding of identity for the device upon deployment. This is reflected in RFC 8366, RFC 8572, BRSKI, FIDO Device Onboard (FDO) and OPC UA, using vouchers. LWM2M provides late binding with its bootstrapping process. (Late binding is the property that the device-management system does not have to be specified during manufacturing.)
- The use of wireless communication standards for provisioning.
- This includes Device Provision Protocol (DPP).
- Device management and service-enablement frameworks.

These include LwM2M and FIDO Device Onboard (FDO), OPC UA and OMA-DM.

3.1 LATE BINDING OF DEVICES USING VOUCHERS

RFC 8366,[8] entitled *A Voucher Artifact for Bootstrapping Protocols* lays the basis for several provisioning mechanisms. It mainly defines a voucher artifact that can be used to support the secure provisioning of IoT devices. New devices, called *pledges*, need to be identified by an initial device ID certificate.

The voucher artifact needs to be signed by the device manufacturer or, to be precise, a Manufacturer Authorized Signing Authority (MASA). The RFC defines a basic set of information that needs to be encoded in the voucher, but this can be extended by subsequent standards.

RFC 8572: Secure Zero Touch Provisioning (SZTP), standardized as RFC 8572,[9] is one of these voucher-based provisioning solutions. It aims to simplify the provisioning process for an operator, including typical functions like boot-image updates, initial configuration settings and the possibility of arbitrary extensions. SZTP defines multiple possible sources for the secure provisioning, like DNS, DHCP or removable media.

BRSKI: Bootstrapping Remote Secure Key Infrastructures (BRSKI) is another RFC draft[10] that used the RFC 8366 bootstrapping voucher and uses another established standard, Enrollment over Secure Transport (EST), to implement its provisioning.

[8] [RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <*https://www.rfc-editor.org/info/rfc8366*>.

[9] [RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <*https://www.rfc-editor.org/info/rfc8572*>.

[10] [RFC8995] Watsen, K. et al., "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC - Proposed Standard, October 2021, <*https://www.rfc-editor.org/rfc/rfc8995.txt*>.

3.2 DEVICE PROVISIONING PROTOCOLS

The *Device Provisioning Protocol* (DPP), also known as WiFi Easy Connect,[11] is a protocol developed by the WiFi Alliance as a successor of WIFI Protected Access and is intended for wireless IoT devices and part of the WPA3 security mechanisms. Main point of DPP is the usage of another out-of-band channel for the secure provisioning. This could be a QR code or NFC tag. After an initial secure channel is established between new device and access point, the access point is responsible for the further configuration of the device.

3.3 DEVICE MANAGEMENT AND SERVICE ENABLEMENT FRAMEWORKS

LwM2M[12] is a device management and service-enablement framework that enables efficient lifecycle management of IoT devices and building applications using standardized protocols, data models, and semantics. LwM2M enables bootstrapping new devices to the management system with initial credentials, adding new and changing existing management servers for operational phase (owner change including credentials for "late binding"), performing firmware and software updates and configuration changes, to de- and re-commissioning the devices. LwM2M has extensive support for device connectivity management, wireless network configuration, and key and certificate lifecycle management. Future support for remote SIM provisioning is planned in GSMA standardization.

Based on the Intel Secure Device Onboarding (SDO), the *FIDO Device Onboard*[13] (FDO) allows for the secure provisioning of IOT devices and provides "late binding". This realized using an *ownership voucher* that can travel parallel to the device through the supply chain. This voucher is sent by the final owner to the device management system of its choosing, to prepare for the actual provisioning by sending the voucher to a rendezvous (RV) server.

After powering-up the device, it establishes a connection to this rendezvous server, that can be located on-premises or online. A list of such servers is stored on the device during manufacturing, alongside a root-of-trust key. Using this device key and the voucher from the RV, both are able to perform mutual authentication. From the RV, the device receives information about the target device management system, from which it can receiver arbitrary credentials for the final integration into the owner's environment, similar to LWM2M bootstrap. Authentication between device and the device management system is performed analog to the connection between device and RV.

The FDO model is based on a client that needs to run on the device and that will be dormant after provisioning is completed. It can be reactivated to perform re-provisioning at later stages.

---

[11] *https://www.wi-fi.org/discover-wi-fi/wi-fi-easy-connect*

[12] OMA LightweightM2M: Open Mobile Alliance. November 2020, <*https://technical.openmobilealliance.org/Overviews/lightweightm2m_overview.html*>

[13] FIDO Device Onboard Specification, Review Draft, FIDO Alliance, December 2020, *https://fidoalliance.org/specs/FDO/FIDO-Device-Onboard-RD-v1.0-20201202.html*

OPC UA[14] is a machine-to-machine communication standard that supports securing transmitted data using a secure channel. Its specification comprises several parts. One of these parts, 21 (currently in draft), focuses on the secure device provisioning. In this specification, the provisioning process is incorporated in the device lifecycle, considering all the steps and involved parties between manufacturer and operator.

As other standards, in OPC UA Part 21 devices are equipped with an initial device identity (IDevID). This IDevID is used to create a DeviceIdentityTicket that is signed by the manufacturer. During manufacturing, multiple DeviceIdentityTickets can be included in a MachineIdentityTicket, which enables support of complex machinery comprising individual devices, and the involvement of multiple manufacturers ('composite builders') in the supply chain.

TR-069[15] is another industry-specific device management framework specifically created for SIP boxes and OMA-DM[16]for mobile phones. It enables remote device provisioning, security management and extended functions like performing firmware upgrades. The specification describes the communication protocol between customer premises equipment (CPE) and an auto configuration server (ACS). The CPE needs to seek out the connection to the ACS by various means, for example by using a pre-configured URL, a DHCP-provided address or a default address.

For *mobile network devices*, the device management can be coupled to the network access credential in the device SIM. This is particularly interesting when the SIM is an eSIM, an integrated SIM in the device that can be remotely managed through GSMA RSP protocols.

3.4 Cloud-Services Standards

Well-known device management-service providers also define mechanisms to integrate new devices securely. The device needs to be equipped with specific credentials prior to shipment and aim to integrate the new device securely in their respective device management-environments. These device management providers include:

- Microsoft Azure IoT Hub,
- Amazon AWS and
- Google Cloud IoT Core.

3.5 Key Findings in Current Provisioning Best Practices

There are more solutions in this area. Established standards point out properties that can be used to differentiate between approaches and help select the appropriate solution for the desired use

[14] OPC Unified Architecture, Part 21: Device Provisioning (Draft), *https://reference.opcfoundation.org/*

[15] TR-069 Amendment 6, CPE WAN Management Protocol, Broadband Forum, April 2018, *https://www.broadband-forum.org/technical/download/TR-069.pdf*.

[16] OMA-DM, *https://technical.openmobilealliance.org/Overviews/dm_overview.html*

case. For example, security properties can differ and affect which side gets authenticated during provisioning (new device or operator, or both using mutual authentication).

Another common feature found in modern provisioning methods is the high degree of automation, requiring only minimal manual actions. On the other hand, initial trust often needs to be established beforehand, for example by preparing the device for its intended operator already at the manufacturing stage. Choosing a standard that includes owner change by late binding will increase the flexibility.

Most solutions are also depending on the usage of identities and trust anchors on devices. Relevant technologies for this are presented in the following section.

4 KEY TECHNOLOGIES

4.1 IDENTITIES

Identities are used to identify entities securely and are defined by an identifier and a secret, for IIoT platforms named as UUID and in chain of trust anchored in RoT.

There are two extremes when it comes how device identities are established. Either the device manufacturer performs this task independent of the device user or owner or the user contacts the manufacturer to have identities programmed to the device. Even if the technology for the identities, the protocols and the hardware were the same, these two approaches to programming identities require different logistics in respect to how information for the devices securely reaches the device management system. The latter is needed so a device management system can recognize its devices when they communicate.

Regardless of how device identities are handled, there are also differences between identity technologies themselves. Different manufacturers and device vendors may have different technical solutions. These differences may be due to ordinary competition but can also be due to regulatory requirements on devices as well as due to national interests and standards. Hence, solutions for onboarding should be capable of dealing with different procedures and technologies for device identities. For special use-cases, including those involving safety-critical systems such as those in the aircraft industry or medical industry, the cost of tailored onboarding solutions can be acceptable, but for the majority of use-cases, low-cost alternatives are needed.

4.2 TRUST STORAGE/ROOT OF TRUST

A root-of-trust in the context of identities, or root-identity technology, is the concept of storing cryptographic keys rooted in the hardware of a device and handling cryptographic operations within this isolated trust domain. If the keys are generated and stored within the hardware, it provides increased trust throughout the lifecycle of the device but may come with decreased flexibility.

NIST SP 800-172[17] (*NIST Roots of Trust Project*) defines root-of-trust as: "*Highly reliable hardware, firmware, and software components that perform specific, critical security functions. Because roots of trust are inherently trusted, they must be secure by design. Roots of trust provide a firm foundation from which to build security and trust*.".

Similar definitions are also provided by *the Global Platform,*[18] *the TCG*[19] and Arm PSA.[20]

This definition is insufficient in an IIoT setting where we want secure onboarding and remote assessment of whether a device has a trusted RoT. For that we need an identity that can be used to have an RoT trust of reporting (also called RTR capability).

We also require that an RoT can perform secure update of firmware and configurations.

The critical security functions and secure processing capabilities that a generic secure IoT device include in an RoT are:

- Hardware features to put a device into a secure state so the device can perform functions that are secured against manipulation. For example, secure boot combined with fused keys that can be used for verification of digital signatures during the boot sequence. This implies that there is a protected environment where processing takes place.
- Protected storage of data-in-use, such as state variables and measurements.
- Secure storage for data-at-rest to hold integrity-protected keys or hash values for code integrity verification.
- Secure storage for data-at-rest hold key material for device identity authentication and remote attestation.

4.2.1 HARDWARE-BASED ROOTS OF TRUST TECHNOLOGIES

Different ROT realizations are in use today. For hardware, the following technologies appear:

- TPM, secure elements (e.g. NXP A71CH, NXP SE050/051, STSAFE A110, GSMA IoT SAFE), secure hardware components that combined with a device processing system implement an RoT and application security functions. TPM is designed to handle source-of-randomness, crypto and key management operations, entropy source and measured boot of a device.

[17] *https://doi.org/10.6028/NIST.SP.800-171r2*

[18] Global Platform Technology Root of Trust Definitions and Requirements Version 1.1: *https://globalplatform.wpengine.com/wp-content/uploads/2018/07/GP_RoT_Definitions_and_Requirements_v1.1_PublicRelease-2018-06-28.pdf*

[19] The TCG glossary: *http://www.trustedcomputinggroup.org/developers/glossary*

[20] *https://www.arm.com/architecture/security-features*

- Intel SGX,[21] AMD SEV,[22] Intel TDX,[23] Arm CCA.[24] Secure hardware integrated with a processor that have RoT functions that provides a hardware protected separation between two execution environments for code and data offering data and code encryption and possibly integrity protection.
- PUF:[25] A Physical Unclonable Function is on its own not an RoT, but can be used in an RoT to act as an identity. As such, PUFs are combined with other (hardware-based) security functions to have secure firmware and hardware that implements to secure processing capability of an RoT.
- Arm TrustZone:[26] ARM technology for a hardware-protected separation between two execution environments for code and data. Combined with secure-boot firmware ARM (e.g. *Trusted Firmware*), TrustZone can realize RoT in a device.
- Secure Element, SIM, eSIM: SIM, the remote profile manageable eSIM and its evolutions can serve as strong identities. Implementations such as GSMA's IoT SAFE,[27] GBA (3G&4G) and AKMA (5G). There exists no RoT solution in the market as well defined in the complete lifecycle, from provisioning to authentication, encryption, protocols, certification, tamper resistance, globally interoperable on all devices as 3GPP/GSMA solutions.
- HSM: A hardware security module can be used in an RoT system of device where it will be a separate component. The challenge is to secure the storage of the credentials to access the HSM.

4.2.2 Software-Based Roots of Trust Technologies

Instead of using hardware features in an RoT, it is possible to have software-based RoTs either by assuming certain operational conditions (e.g. the device is physically in a benign environment, or the software RoT resides in an environment that is sufficiently isolated, using HW features like ARM TrustZone or standard MMU/MPU isolation under control of virtual machine managers or security kernels).

Two examples of software-based RoTs are the virtual TPM (vTPM) and the firmware TPM implemented in commercial CPU systems.

---

[21] Intel SGX: Intel Software Guard eXtensions: *https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html*

[22] AMD SEV: *https://developer.amd.com/sev/*

[23] Intel TDX: *https://software.intel.com/content/www/us/en/develop/articles/intel-trust-domain-extensions.html*

[24] Arm CCA: *https://www.arm.com/why-arm/architecture/security-features/arm-confidential-compute-architecture*

[25] *https://puffin.eu.org/*

[26] Arm TrustZone for Cortex A: *https://developer.arm.com/ip-products/security-ip/trustzone/trustzone-for-cortex-a*

[27] *https://www.gsma.com/iot/iot-safe/*

A software-based RoT must be based on, and chained to, a hardware-based RoT, for example by establishing chaining mechanisms such as secure boot or hardware-RoT-based attestation. The chaining mechanisms create trust to software running different privilege levels that serve as a software-based RoT for other parts in the device.

5 EMERGING DECENTRALIZED TECHNOLOGY FOR DEVICE ONBOARDING

The current device-onboarding techniques mainly rely on centralized entities (e.g. certificate authorities) and public key infrastructures (PKIs). Such centralized approaches typically use data centers to manage IoT devices. Emerging Self-Sovereign Identity (SSI) techniques, such as Decentralized Identifiers (DIDs[28]) and Verifiable Credentials (VCs), provide promising solutions to protect the identities of the devices from being misused or altered as part of architectural planning and the onboarding stage. SSI technology has shed light on developing a decentralized device onboarding approach for IoT, as highlighted by the ongoing standardization effort by IEEE P2958[29] working group.

SSI comprises three technology pillars:

- Decentralized Identifiers (DIDs): A DID is a new type of globally unique identifier, namely a string of characters that uniquely identifies a resource or entity. DIDs are the foundational technologies of SSI. In IoT systems, DIDs can be used to identify all kinds of machines (e.g. IoT devices, edge nodes, servers). Each machine DID is associated with a private/public key pair that might be generated on the machine or by the owner on behalf of the machine. For more details on DIDs, see the "Decentralized Identifiers (DIDs) v1.0"[30] specification by W3C.
- DIDComm Messaging: DIDComm is a messaging protocol for people, organizations and IoT devices interacting with machine-readable messages and creating DID-based relationships. The exchange of DIDs enables two parties to authenticate each other and form a secure messaging channel called DIDComm. DIDComm messages are exchanged between software agents of entities for implementing a variety of security protocols. In IoT systems, DIDComm can be used to establish secure and private communication channels between two entities (e.g. two IoT devices or an IoT device and its owner) without relying on a certificate authority (CA). For more details about DIDComm, see the "DIDComm Messaging"[31] specification by DIF.[32]

[28] *https://www.iiconsortium.org/pdf/IIC-Edge-DID-Tech-Brief.pdf*

[29] IEEE P2958—Standard for a Decentralized Identity and Access Management Framework for Internet of Things (see *https://sagroups.ieee.org/2958/)*

[30] *https://www.iiconsortium.org/pdf/Trustworthiness_Framework_Foundations.pdf*

[31] *https://identity.foundation/didcomm-messaging/spec/*

[32] *https://identity.foundation/working-groups/did-comm.html*

- Verifiable Credentials (VCs): A VC is a digitally signed electronic credential about identity attributes, which allows an entity to form its identity under a certain context based on a set of claims about an identifier. VCs are cryptographically secure, privacy-respecting and automatically verifiable. Each VC has an issuer that stores the credential definition, public DID, credential schema, and revocation registry on a blockchain. The credential protocols defined in DIDComm govern the process of exchanging VCs peer to peer between software agents. In IoT systems, VCs are used to attest the security related attributes of machines, thereby establishing their trustworthiness. For more details about VCs, see the "Verifiable Credentials Data Model v1.1[33] specification by W3C.

A decentralized device-onboarding approach relies on DIDs that are usually generated within IoT devices during the manufacturing process. Once IoT devices and other stakeholders (e.g. device manufacturers, device owners and system integrators) register their DIDs on a verifiable data registry (e.g. a distributed ledger), devices are able to establish DID-based peer relationships via the DIDComm messaging protocols with various stakeholders during their lifecycles. Pairwise secure communication channels can be established between IoT devices and other stakeholders for provisioning network credentials, application firmware images etc. Moreover, VCs could be used to attest a variety of device attributes (i.e. microcontroller, operating system, secure hardware) as well as their ownerships, thereby enabling stakeholders to evaluate device trustworthiness during the device onboarding process.

6 END-TO-END CHAINING OF TRUST BY ID FEDERATION

To avoid concerns of being locked-in to a specific platform or technology and to allow an easy integration with existing device management systems, a standardized federation with bootstrapping enhances secure device management features as a generic part of device lifecycle management.

The onboarding or bootstrapping of devices into a device management system is a generic step in the lifecycle of most if not all IoT devices that are part of some form of service.

The onboarding step is crucial as many operations after onboarding assume that the devices are trustworthy.

Procedures for secure onboarding must be efficient and automated when there are many devices. For that reason, devices receive individual or group identities so users can assess that the devices are genuine automatically and consequently have the required security properties. Through certified procedures, these identities are programmed into the devices and with secure cryptographic protocols and device management systems can authenticate the device or receive

[33] *https://www.w3.org/TR/vc-data-model/*

device attestation reports for an even higher degree of assessment of the identity and security properties of the device.

A *unified identity bootstrapping approach* aims to provide the glue layer between identity domains, device-identity solutions in the onboarding processes of device management systems.

Through its capability to automate onboarding across different device identities, unified identity bootstrapping offers scalability and flexibility to users and protects enterprises against technology lock-ins. It also addresses the practical questions related to ensuring the integrity of devices operating in a managed network that are outsourced to managed service providers (MSPs) for remote operation and maintenance.

These aspects becoming increasingly a concern when we scale to a massive use of devices where services interact with large numbers of IoT devices.

These concerns hamper growth in IIoT. They have to be met by solutions that make it easier for users and enterprises to create services in IIoT. Automation of lifecycle handling has to be possible from the beginning without worrying that devices may come with different root identities technologies.

Lifecycle management of identities and credentials is closely coupled to device lifecycle management. Lifecycle management of identities requires:

- generation and provisioning of credentials associated with identities,
- update and renew credentials associated with an identity,
- revocation of identities and
- bootstrap a new identity using an existing identity.

Protocols through which the device can be reached for the bootstrap may differ. For example, the device may use LwM2M, FDO or UPC-UA standards. Unified identity bootstrapping provides interoperability across device technologies and support long device lifetimes.

A scheme for unified identity bootstrapping should be able to achieve cryptographic agility and be capable of handling different key sizes and types. This is needed when devices are in use for a long time during which a given crypto algorithm may lose its protective capability.

The operations of updating and bootstrapping identities play a role lifecycle management in unified identity bootstrapping, and they are part of the identity system of the device. The identity system comprises:

- identity (credentials) technology in the device (could be pre-shared key (PSK), raw public key or X.509 based identities),
- a system (tools, servers etc.) to provision identities into the devices and
- a system to provide users of the identity securely with the needed information for identifying devices (e.g. trusted provisioning of root certificates).

When realizing a secure identity system, the identity technology should follow best security practices in cryptography, using appropriate cryptographic algorithms, protocols and having crypto agility to cope with long life-time expectancy of devices.

These operations must exist in the device and must be provided by a trustworthy engine in the device. The engine also has to provide APIs to use the identities, e.g. in an authentication protocol, TLS/ DTLS or OSCORE, but these are implementation details. The trustworthy engine is created around a device RoT.

With this you could achieve automated onboarding of devices, using a trusted ID technology, provided by a trusted device source, by building a chain of trust from device RoT to your IoT platform.

7 CONCLUSIONS

The process of device onboarding makes a device functionally usable in a given system and a process by which the systems establish trust in the operation of the device and the data the system collects from the device. Devices may use different technologies for the automation of the onboarding which makes it desirable from a system point of view to be able to perform automatic onboarding regardless the technology. A common denominator is the binding of identities to device. Here, late-binding is attractive.

How identities are established and managed and the concept of root-of-trust should be used to achieve trust in the solution. Various frameworks, communication protocols and approaches toward root of trust and identity can be considered and related standards and best practices have been reviewed in this paper.

AUTHORS & LEGAL NOTICE

This document is a work product of the Industry IoT Consortium Security Working Group, chaired by Keao Caindec (Farallon).

Authors: John Fornehed (Ericsson), Keao Caindec (Farallon), David Meier (Fraunhofer), Xinxin Fan (IoTeX), Mitch Tseng (Tseng InfoServ), Ben Smeets (Ericsson), Ilhan Gurel (Ericsson), Paul Bradley (Arm), Frederick Hirsch (Upham Security), Jeff Shiner (Micron), Hanu Kommalapti (Microsoft), Alex B. Ferraro (PwC).

Technical Editor: Stephen Mellor (IIC CTO) oversaw the process of organizing the contributions of the above Authors into an integrated document.