Workgroup: IOTOPS
Internet-Draft:
draft-vattaparambil-iotops-poa-based-
onboarding-01
Published: 28 March 2023
Intended Status: Informational
Expires: 29 September 2023
Authors: Sreelakshmi
Lulea University of Technology
Olov
Lulea University of Technology
Ulf
Lulea University of Technology

draft-vattaparambil-iotops-poa-based-onboarding-01

Abstract

Industrial network layer onboarding demands a technique that is efficient, scalable, and secure. In this document, we propose Power of Attorney based authorization technique as a decentralized solution for onboarding devices. This enables users such as integrators and subcontractors to onboard devices permanently or temporarily according to terms and requirements set in the PoAs.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 29 September 2023.

Copyright Notice

Table of Contents

1. Introduction

Onboarding devices in industrial setting must be efficient, scalable, and secure. NIST guidelines on network layer onboarding [NIST] explain essential features required by an ideal onboarding model. Many zero touch onboarding models require the manufacturer to build and configure devices with specific onboarding features based on the destination network. It is complex to gather the onboarding requirements from multiple parties involved based on a centralized infrastructure, which makes it expensive and inefficient.

The Power of Attorney (PoA) based onboarding can secure the device with unique onboarding credentials during deployment rather than at the time of manufacture. This approach is based on subgranting or delegation based authorization, in which power or delegation can be granted to another entity for a limited time. This can be used between different parties in the supply chain and with integrators for ultimate onboarding in at the customer site. It can also be used in typical industrial subcontractor usecases where devices owned by subcontractors must/should temporarily (ie., for limited time) be onboarded to an industrial site while the formal ownership is retained by the subcontractor. The PoA based onboarding primarily

addresses autonomous or semi-autonomous devices that are not resource constrained. The PoA ensures authorization between the device and the industrial site onboarding controller, which ultimately approves the onboarding based on certificates. In the proposed model, we establish a trust chain between the subcontractor, device, and the onboarding component for automatic onboarding of devices using power of attorney based authorization technique.

Note that in this document we focus on the onboarding case using PoA while indeed PoA is completely generic and can be used in various other subgranting, ownership transfer, and data sharing usecases, not covered in this document.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Onboarding basics

2.1. State of the art

Device onboarding can be defined as an automated process of securely provisioning the device at the destination network from the manufacturer’s site via the supplychain. One aspect of onboarding is providing the device with network access [nordmark-iotops]. There are different definitions for onboarding; Intel zero touch onboarding [Intel] refers it as an ”Automated service that enables a device to be drop-shipped and powered on to dynamically provision to a customer’s IoT platform of choice in seconds”. According to Amazon Web Services (AWS), ”IoT device onboarding or provisioning refers to the process of configuring devices with unique identities, registering these identities with their IoT endpoint, and associating required permissions”. NIST guidelines are also referred by IETF [t2trg], ”Onboarding is sometimes used as a synonym for bootstrapping and at other times is defined as a subprocess of bootstrapping”. According to the guidelines provided by NIST, onboarding can be performed in two different layers:

*Network layer onboarding

*Application layer onboarding.

The network layer onboarding may ensure device integrity and authorized ownership throughout the initial phases of onboarding. The information gathered during network layer onboarding is passed

to application layer onboarding to make the device operational in the application layer.

2.2. Problem description

The main issues in a device lifecycle are device ownership transfer, management of the device after bootstrapping such as installing required software, its maintenance, and disposition of the device when transitioning to a new owner. Because of the large number of external devices and the security issues caused by their communication, device onboarding is considered as an important process. Multiple entities, transportation methods, sensitive data sharing, and other factors make the onboarding process difficult, necessitating automation and security. Hence, there is a need for an efficient onboarding procedure that secures devices with unique onboarding credentials during deployment rather than at the time of manufacture.

3. Power of Attorney based authorization

PoA-based authorization is a generic authorization technique used to authorize devices to access protected resources on behalf of the user, who owns the device (principal), even if the user is not online. The PoA model in its base form is completely decentralized (like for example Pretty Good Privacy (PGP)), where the user subgrants their power in the form of a self- contained PoA that contains public information such as public keys and a specific set of permissions for a predefined time. It is a decentralized authorization technique, where the different entities involved can access and verify the PoA using a downloadable image or library similar to PGP. Some centralization can be added by optional signatory registers and/or traditional Certificate Authorities (CA). The entities involved in PoA based authorization system are:

*Principal: The entity that generates and sends the PoA to the agent.

*Agent: The device which receives the PoA to sign on behalf of the principal with limited features for a pre-defined time.

*Resource server: The third party with a server that stores the information and credentials entitled to the principal. It serves agents according to subgrants defined in PoAs.

*Signatory registry: A database system where PoAs and system-related metadata are stored. It can serve as a trusted third-party in certifying and verifying PoA. This component is optional.

The principal generates the PoA in advance to entitle an agent to autonomously execute tasks in the absence of the principal. The PoA is digitally signed by the principal and the agent uses the limited features of the principal’s account to execute tasks allowed by the PoA.

4. Power of Attorney based Onboarding

This document consider the network layer onboarding and subgranting the power to onboard from one entity to another in the bootstrapping stage. The different roles are:

*Subcontractor (Principal): The subcontractor is the device owner, who obtains the device from the supplychain.

*Device (Agent): The device to be onboarded.

*Gateway: We assume that all the communication between the IoT device, subcontractor, and the onboarding controller is through a secure gateway for better security.

*Onboarding component: Onboards the device to the destination network.

*Certificate Authority (CA): It provides the local cloud compliant certificate to the device for onboarding.

Figure 1 shows the Protocol flow diagram of the proposed model.

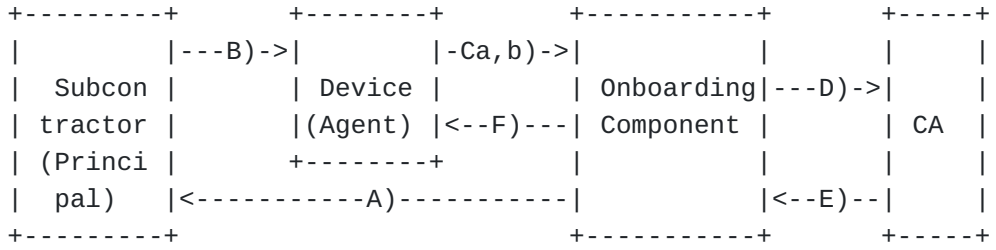

Figure 1: Protocol flow of PoA based onboarding

*A) Onboarding component sends the PoA1 (PoA generated by the onboarding component) to the subcontractor through the gateway. By this, the onboarding component grants authorization to a specific subcontractor to bootstrap any of its trusted devices.

Before this step, both entities should be mutually authenticated using public key certificates.

*B) Subcontractor generates PoA2 and sends it to his/her specific trusted device. This enables the device to work on behalf of the subcontractor. This means, the onboarding component that trusts the subcontractor (through PoA1) implicitly trusts the device. In this step, the subcontractor may add the complete ownership of the device's proof-of-chain information to PoA2, if so required (e.g., as specified in PoA1).

*Ca) The device sends the PoA2 including metadata such as device hash and device bootstrapping credentials to the onboarding component through the gateway. The device bootstrapping credentials can includes device identifier (e.g., X.509 certificate-DevID, Device Identifier Composition Engine [DICE] Compound Device Identifier [CDI], public key), device private key or csr, Wi-Fi channel that the device will use (optional), communications protocols (optional) etc.

*Cb) Secure channel establishment using Mutual TLS (MTLS).

*D) Onboarding component authorizes the device by verifying the PoA2 and sends a certificate request using device private key or csr to the local cloud CA.

*E) The local cloud CA verifies the submitted documents and generates the a local cloud compliant device certificate and sends it to the onboarding component.

*F) The network bootstrapping credentials are sent to the device by the onboarding component via the gateway. This can include network identifier (e.g., X.509 certificate, Service Set Identifier [SSID]). The device validates the network by comparing the network details in the network bootstrapping credentials to the network details in the digitally signed PoA2. This helps the device to determine if the target network is authorized to onboard the device.

The revocation of PoA can be accomplished by setting a low expiration time depending on the use case. In that case the PoA must be reissued periodically.

Once the device obtains the network bootstrapping credentials, it can start communicating with the local cloud. This model for onboarding enables the subcontractor to onboard devices by subgranting his/her power to the device to act on behalf of the subcontractor. A proof of concept of the proposed model can be found at "https://github.com/sreelakshmivs/PoAimplementationinJava" under the MIT license.

5. PoA Structure

The PoAs are self-contained tokens that are structured in JWT format. The entire PoA in the JWT form is digitally signed by the principal using his/her private key. It is compressed into binary format (e.g., CBOR). The various parameters included in a PoA are the following:

Principal Public Key
: REQUIRED. The public key, which uniquely identifies the principal who generates the PoA. We assume that the public key is generated using a secure public-key algorithm by the principal. With this parameter, the authorization server can identify the person who generated the PoA.

Principal Name
: OPTIONAL. The human-readable name of the principal, which is additional information about the principal.

Resource Owner ID
: REQUIRED. The unique identifier or the public key of the resource owner from where the protected resources are granted.

Agent Public Key
: REQUIRED. The public key, which uniquely identifies the agent who receives the PoA from the principal. We assume that the agent public key is generated using a secure public-key algorithm by the owner. This parameter helps the trusted server to identify the agent and check whether it is genuine or not.

Agent Name
: OPTIONAL. The human-readable name of the agent, which is additional information about the agent.

Signing Algorithm
: OPTIONAL. The name of the signature algorithm used by the principal to digitally sign the PoA.

Transferable
: REQUIRED. It is a positive integer defining how many steps the PoA can be transferred. Default is 0, which means that it is not transferable. A PoA can be transferred by including it in another

PoA, i.e., it is signed in several delegation steps (where the number is decreased by one in each step).

iat (Issued at)
REQUIRED. The time at which the PoA is issued by the principal to the agent.

eat (Expires at)
REQUIRED. The time at which the PoA expires. This parameter is predefined by the principal in the PoA and the PoA will be invalid after eat.

Metadata
OPTIONAL. The metadata is associated with the specific application use-case. This parameter includes different sub-parameters that add application-specific information to the PoA.

6. Related Works

[nordmark-iotops] recognize the need for an effective onboarding system in both network and application layers. This approach doesn't require much dependency on the manufacturer and the manufacturer certificates. They define the flexibility of devices that are not resource constrained such as Raspberry Pi and larger. The use of large smart devices enables executing functions that are not envisioned during their manufacturing.

Fast IDentity Online Alliance (FIDO) [fidospec] defines an automatic onboarding protocol for IoT devices. With the late binding feature of this protocol, the IoT platform for the IoT device doesn't need to be selected in the early stage of its life cycle, and reduces the cost and complexity in the supplychain. FIDO uses a rendezvous server for device registration and to find the device owner location, by assuming that the device has an IP connectivity to the rendezvous server. An important feature of FIDO is the tracking of transfer of ownership and the device's late-bound owner throughout the supplychain using the ownership voucher. FIDO Device Onboard enabled Device is configured with required software and hardware along with a Restricted Operating Environment (ROE) and a Management Agent, that manages the device ownership voucher using the onboarding protocols. Another important parameter is the device credentials, it does not permanently identify the user and is only used for the purpose of the ownership transfer. FIDO expects that both the manufacturer and the owner will change their keys frequently. Main protocols in FIDO onboarding are Device initialization protocol (DI), Transfer Ownership Protocol (TO0), TO1, and TO2. The function of DI is to insert FIDO Device Onboard credentials into the device during the manufacturing process. TO0 is used by the owner to identify itself to the rendezvous server, and

similarly TO1 is used by the device to identify itself and to interact with the rendezvous server using the device ROE. TO2 is used by the device ROE to contact and interact with the owner or device onboarding service. After TO2 successfully completes, the device onboarding credentials except the attestation key is replaced by the owner onboarding service.

[eap-onboarding] defines an onboarding method where an unconfigured device can be added to the network using EAP, which later can be onboarded. Here, the onboarding process is divided into different stages such as discovery, authentication, authorization, onboarding, and full network access. The devices that obtained network access using unauthenticated EAP undergoes onboarding process once they enter the captive portal.

[t2trg] provides a survey on different standards and protocols for onboarding. Onboarding is referred using different names as part of the initial security setup of devices. This list of names include bootstrapping, provisioning, enrollment, commissioning, initialization, and configuration. Most approaches rely on an external anchor such as rendezvous server, bootstrap server, chip or QR code.

The communication protocol [mobileIP] uses a home agent and a foreign agent to facilitate mobility. The home agent provides an anchor point for connectivity, while a mobile node can register with a foreign agent to get seamless connectivity at the visited network. This allows the user to move between different networks while having both the home and visitor IP addresses. However, this is primarily to obtain internet access, not to onboard a local realm.

PoA based authorization can be added as a new grant type for OAuth protocol, that introduces a new role "principal" who controls the client, and enables the client to access resources through the OAuth authorization server on behalf of the principal, even if the principal is not available online [poa-oauth-grant-type].

PoA-based authorization is an industrial authorization technique for CPS devices that is designed with different cryptographic algorithms, is a similar work as the proxy signature with warrant [proxy-signature]. The proxy signature is a significant security cryptographic algorithm that strengthens its security by patching newer security loopholes. The main differences are seen in the applicability of the technique and the design methodology. In proxy signature, the agent or proxy signer is required to perform several cryptographic calculations to sign a message, as described in the warrant on behalf of the principal. PoA can be seen as a more industry oriented technique, where the device acts/works on behalf of the principal as described in the PoA. Here, the agent is only

required to verify and forward the PoA (received from the principal) to the resource owner and provide its strong identity, to obtain the resources on behalf of the principal.

The different techniques mentioned above use a delegation-based authorization model for security, which relies on centralized servers or complex cryptographic algorithms, limiting their flexibility in the onboarding process. The PoA-based authorization technique, that does not rely on a centralized server and employs an industry-friendly PoA structure, enables for a reliable and flexible onboarding process.

7. Security Considerations

The security of the entire onboarding process relies on issues with security in different phases such as manufacturing, supply chain, bootstrapping, and application. The characteristics of these phases differ depending on the onboarding approach. The following are the different approaches:

* Use hardware manufacturer certificates. Using the manufacturing certificate, this method authenticates the device. However, there is no option to authorize the target network, which prevents the device from being onboarded to fraudulent networks.

* Tracking ownership transfers throughout the supply chain. This secure late binding to the management system/controller allows the controller to trust the device and ensure that it is not compromised during the supply chain transmission.

* Imprinting/configuring for/by the owner of the device. This approach configures the device for its future owner/controller by imprinting the future owner's identity. This methods enables the device to only onboard to the trusted owner/controller. However, it requires the manufacture to build devices with customized features based on their future owner/controller.

* PoA based onboarding. This decentralized approach employs the subgranting based authorization technique, that enables the controller to grant authorization to the subcontractor (principal) and the device to obtain authorization from the subcontractor. PoA approach compliments the above three approaches with the use of digitally signed PoAs that enables mutual authorization between the device and the controller, and the use of PoA to keep track of the ownership transfer, which is submitted to the controller on demand.

7.1. Attacks out of scope

The payload data in the form of PoAs is immutable and protected by cryptographic signatures. Therefore, integrity threats like replay, message insertion, modification and man in the middle are out of scope.

7.2. Attacks in scope

Confidentiality threats like eavesdropping exist when PoAs are sent as clear data. However, this can be resolved by e2e encryption. For authentication, the PoAs rely on strong unique identities, e.g., the identity of an must be verified when it turns up with a PoA where it obtains some authorized credentials based on its public key. In some cases, a private key can serve for proving identity, but it should be noted that a private key can be stolen (Identity theft). This can be resolved by coupling the identity uniquely to the device, e.g., a device hash, X.509 certificate–DevID, Device Identifier Composition Engine [DICE], Compound Device Identifier [CDI], public key. The protocol interface for receiving and processing PoAs is susceptible to denial-of-service attacks, where potential overload attacks using meaningless or unacceptable PoAs could be issued. Possible resolutions to this threat will be addressed in future versions of this draft.

We will conform to prefer industry standards e.g., as described in [draft-moran-iot-nets-01]

8. References

8.1. Normative References

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

8.2. Informative References

[NIST] National Institute of Standards and Technology, "Trusted Internet of Things (IoT) device network-layer onboarding

and lifecycle management (draft) No. NIST CSWP 16 ipd", 2020.

[Intel] INTEL, "Intel® secure device onboard," More secure, automated IoT device onboarding in seconds, pp. 1–4", 2017.

[t2trg] Internet Engineering Task Force, "draft-irtf-t2trg-secure-bootstrapping-02", 2022.

[nordmark-iotops] Internet Engineering Task Force, "draft-nordmark-iotops-onboarding-00", 2021.

[fidospec] Fido Alliance, "Fast Identity Online Alliance, "FIDO Device Onboard Specification"", 2021, <https://fidoalliance.org/specifications/download-iot-specifications/>.

[mobileIP] "IP mobility support. No. rfc2002", 1996.

[proxy-signature] "Proxy signatures: Delegation of the power to sign messages," IEICE transactions on fundamentals of electronics, communications and computer sciences, vol. 79, no. 9, pp. 1338–1354", 1996.

[draft-moran-iot-nets-01] Internet Engineering Task Force, "A summary of security-enabling technologies for IoT devices", 12062022.

[poa-oauth-grant-type] Internet Engineering Task Force, "draft-vattaparambil-oauth-poa-grant-type-00", 11032023.

[eap-onboarding] Internet Engineering Task Force, "draft-richardson-emu-eap-onboarding-02", 4022023.

Contributors

Thanks to all of the contributors.

Authors' Addresses

Sreelakshmi
Lulea University of Technology
SE-97187 Lulea
Sweden

Email: srevat@ltu.se

Olov
Lulea University of Technology

SE-97187 Lulea
Sweden

Email: olov.schelen@ltu.se

Ulf
Lulea University of Technology
SE-97187 Lulea
Sweden

Email: ulf.bodin@ltu.se