NIST SPECIAL PUBLICATION 1800-36C

Trusted Internet of Things (IoT) Device Network-Layer Onboarding and Lifecycle Management:

Enhancing Internet Protocol-Based IoT Device and Network Security

Volume C:
How-To Guides

Murugiah Souppaya
Paul Watrobski
National Institute of Standards and Technology
Gaithersburg, Maryland

Chelsea Deane
Joshua Klosterman
Blaine Mulugeta
Charlie Rearick
Susan Symington
The MITRE Corporation
McLean, Virginia

Dan Harkins
Danny Jump
Aruba, a Hewlett Packard
Enterprise company
San Jose, California

Andy Dolan
Kyle Haefner
Craig Pratt
Darshak Thakore
CableLabs
Louisville, Colorado

May 2023

PRELIMINARY DRAFT

This publication is available free of charge from
https://www.nccoe.nist.gov/projects/trusted-iot-device-network-layer-onboarding-and-lifecycle-management

NIST NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

DISCLAIMER

Certain commercial entities, equipment, products, or materials may be identified by name or company logo or other insignia in order to acknowledge their participation in this collaboration or to describe an experimental procedure or concept adequately. Such identification is not intended to imply special status or relationship with NIST or recommendation or endorsement by NIST or NCCoE; neither is it intended to imply that the entities, equipment, products, or materials are necessarily the best available for the purpose.

While NIST and the NCCoE address goals of improving management of cybersecurity and privacy risk through outreach and application of standards and best practices, it is the stakeholder's responsibility to fully perform a risk assessment to include the current threat, vulnerabilities, likelihood of a compromise, and the impact should the threat be realized before adopting cybersecurity measures such as this recommendation.

National Institute of Standards and Technology Special Publication 1800-36C, Natl. Inst. Stand. Technol. Spec. Publ. 1800-36C, 32 pages, May 2023, CODEN: NSPUE2

FEEDBACK

You can improve this guide by contributing feedback. As you review and adopt this solution for your own organization, we ask you and your colleagues to share your experience and advice with us.

Comments on this publication may be submitted to: iot-onboarding@nist.gov.

Public comment period: May 3, 2023 through June 20, 2023

National Cybersecurity Center of Excellence
National Institute of Standards and Technology
100 Bureau Drive
Mailstop 2002
Gaithersburg, MD 20899
Email: nccoe@nist.gov

NATIONAL CYBERSECURITY CENTER OF EXCELLENCE

The National Cybersecurity Center of Excellence (NCCoE), a part of the National Institute of Standards and Technology (NIST), is a collaborative hub where industry organizations, government agencies, and academic institutions work together to address businesses' most pressing cybersecurity issues. This public-private partnership enables the creation of practical cybersecurity solutions for specific industries, as well as for broad, cross-sector technology challenges. Through consortia under Cooperative Research and Development Agreements (CRADAs), including technology partners—from Fortune 50 market leaders to smaller companies specializing in information technology security—the NCCoE applies standards and best practices to develop modular, adaptable example cybersecurity solutions using commercially available technology. The NCCoE documents these example solutions in the NIST Special Publication 1800 series, which maps capabilities to the NIST Cybersecurity Framework and details the steps needed for another entity to re-create the example solution. The NCCoE was established in 2012 by NIST in partnership with the State of Maryland and Montgomery County, Maryland.

To learn more about the NCCoE, visit https://www.nccoe.nist.gov/. To learn more about NIST, visit https://www.nist.gov.

NIST CYBERSECURITY PRACTICE GUIDES

NIST Cybersecurity Practice Guides (Special Publication 1800 series) target specific cybersecurity challenges in the public and private sectors. They are practical, user-friendly guides that facilitate the adoption of standards-based approaches to cybersecurity. They show members of the information security community how to implement example solutions that help them align with relevant standards and best practices, and provide users with the materials lists, configuration files, and other information they need to implement a similar approach.

The documents in this series describe example implementations of cybersecurity practices that businesses and other organizations may voluntarily adopt. These documents do not describe regulations or mandatory practices, nor do they carry statutory authority.

KEYWORDS

application-layer onboarding; bootstrapping; Internet of Things (IoT); Manufacturer Usage Description (MUD); network-layer onboarding; onboarding; Wi-Fi Easy Connect.

ACKNOWLEDGMENTS

We are grateful to the following individuals for their generous contributions of expertise and time.

Name	Organization
Amogh Guruprasad Deshmukh	Aruba, a Hewlett Packard Enterprise company
Bart Brinkman	Cisco
Eliot Lear	Cisco
Peter Romness	Cisco
Tyler Baker	Foundries.io
George Grey	Foundries.io
David Griego	Foundries.io
Fabien Gremaud	Kudelski IoT
Brecht Wyseur	Kudelski IoT
Faith Ryan	The MITRE Corporation
Nicholas Allot	NquiringMinds
Toby Ealden	NquiringMinds
Alois Klink	NquiringMinds
John Manslow	NquiringMinds
Antony McCaigue	NquiringMinds
Alexandru Mereacre	NquiringMinds

Name	Organization
Craig Rafter	NquiringMinds
Loic Cavaille	NXP Semiconductors
Mihai Chelalau	NXP Semiconductors
Julien Delplancke	NXP Semiconductors
Anda-Alexandra Dorneanu	NXP Semiconductors
Todd Nuzum	NXP Semiconductors
Nicusor Penisoara	NXP Semiconductors
Laurentiu Tudor	NXP Semiconductors
Michael Richardson	Sandelman Software Works
Karen Scarfone	Scarfone Cybersecurity
Mike Dow	Silicon Labs
Steve Egerter	Silicon Labs
Steve Clark	WISeKey
Pedro Fuentes	WISeKey
Gweltas Radenac	WISeKey
Kalvin Yang	WISeKey

The Technology Partners/Collaborators who participated in this build submitted their capabilities in response to a notice in the Federal Register. Respondents with relevant capabilities or product components were invited to sign a Cooperative Research and Development Agreement (CRADA) with NIST, allowing them to participate in a consortium to build this example solution. We worked with:

Technology Collaborators		
Aruba, a Hewlett Packard Enterprise company	Kudelski IoT	Sandelman Software Works
CableLabs	NquiringMinds	Silicon Labs
Cisco	NXP Semiconductors	WISeKey
Foundries.io	Open Connectivity Foundation (OCF)	

DOCUMENT CONVENTIONS

The terms “shall” and “shall not” indicate requirements to be followed strictly to conform to the publication and from which no deviation is permitted. The terms “should” and “should not” indicate that among several possibilities, one is recommended as particularly suitable without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. The terms “may” and “need not” indicate a course of action permissible within the limits of the publication. The terms “can” and “cannot” indicate a possibility and capability, whether material, physical, or causal.

CALL FOR PATENT CLAIMS

This public review includes a call for information on essential patent claims (claims whose use would be required for compliance with the guidance or requirements in this Information Technology Laboratory (ITL) draft publication). Such guidance and/or requirements may be directly stated in this ITL Publication or by reference to another publication. This call also includes disclosure, where known, of the existence of pending U.S. or foreign patent applications relating to this ITL draft publication and of any relevant unexpired U.S. or foreign patents.

ITL may require from the patent holder, or a party authorized to make assurances on its behalf, in written or electronic form, either:

a) assurance in the form of a general disclaimer to the effect that such party does not hold and does not currently intend holding any essential patent claim(s); or

b) assurance that a license to such essential patent claim(s) will be made available to applicants desiring to utilize the license for the purpose of complying with the guidance or requirements in this ITL draft publication either:

1. under reasonable terms and conditions that are demonstrably free of any unfair discrimination; or
2. without compensation and under reasonable terms and conditions that are demonstrably free of any unfair discrimination.

Such assurance shall indicate that the patent holder (or third party authorized to make assurances on its behalf) will include in any documents transferring ownership of patents subject to the assurance, provisions sufficient to ensure that the commitments in the assurance are binding on the transferee, and that the transferee will similarly include appropriate provisions in the event of future transfers with the goal of binding each successor-in-interest.

The assurance shall also indicate that it is intended to be binding on successors-in-interest regardless of whether such provisions are included in the relevant transfer documents.

Such statements should be addressed to: iot-onboarding@nist.gov.

Contents

List of Figures

1 Introduction

The following volumes of this guide show information technology (IT) professionals and security engineers how we implemented these example solutions. We cover all of the products employed in this reference design. We do not re-create the product manufacturers' documentation, which is presumed to be widely available. Rather, these volumes show how we incorporated the products together in our environment.

Note: These are not comprehensive tutorials. There are many possible service and security configurations for these products that are out of scope for this reference design.

1.1 How to Use This Guide

This NIST Cybersecurity Practice Guide demonstrates a standards-based reference design for implementing trusted IoT device network-layer onboarding and lifecycle management and describes various example implementations of this reference design. Each of these implementations, which are known as *builds,* is standards-based and is designed to help provide assurance that networks are not put at risk as new IoT devices are added to them and to help safeguard IoT devices from connecting to unauthorized networks. The reference design described in this practice guide is modular and can be deployed in whole or in part, enabling organizations to incorporate trusted IoT device network-layer onboarding and lifecycle management into their legacy environments according to goals that they have prioritized based on risk, cost, and resources.

NIST is adopting an agile process to publish this content. Each volume is being made available as soon as possible rather than delaying release until all volumes are completed. Work implementing the example solutions and developing other parts of the content continues. As a preliminary draft, we will publish at least one additional draft for public comment before it is finalized.

When complete, this guide will contain five volumes:

- NIST Special Publication (SP) 1800-36A: *Executive Summary* – why we wrote this guide, the challenge we address, why it could be important to your organization, and our approach to solving this challenge
- NIST SP 1800-36B*: Approach, Architecture, and Security Characteristics* – what we built and why
- NIST SP 1800-36C: *How-To Guides* – instructions for building the example implementations, including all the security-relevant details that would allow you to replicate all or parts of this project **(you are here)**
- NIST SP 1800-36D: *Functional Demonstrations* – use cases that have been defined to showcase trusted IoT device network-layer onboarding and lifecycle management security capabilities and the results of demonstrating these use cases with each of the example implementations

- NIST SP 1800-36E*: Risk and Compliance Management* – risk analysis and mapping of trusted IoT device network-layer onboarding and lifecycle management security characteristics to cybersecurity standards and recommended practices

Depending on your role in your organization, you might use this guide in different ways:

Business decision makers, including chief security and technology officers, will be interested in the *Executive Summary, NIST SP 1800-36A*, which describes the following topics:

- challenges that enterprises face in migrating to the use of trusted IoT device network-layer onboarding
- example solutions built at the NCCoE
- benefits of adopting the example solution

Technology or security program managers who are concerned with how to identify, understand, assess, and mitigate risk will be interested in *NIST SP 1800-36B*, which describes what we did and why.

Also, Section 3 of *NIST SP 1800-36E* will be of particular interest. Section 3, Trusted IoT Device Network-Layer Onboarding and Lifecycle Management Reference Architecture Security Mappings, maps logical components of the general trusted IoT device network-layer onboarding and lifecycle management reference design to security characteristics listed in various cybersecurity standards and recommended practices documents, including *Framework for Improving Critical Infrastructure Cybersecurity* (NIST Cybersecurity Framework) and *Security and Privacy Controls for Information Systems and Organizations* (NIST SP 800-53).

You might share the *Executive Summary, NIST SP 1800-36A*, with your leadership team members to help them understand the importance of using standards-based trusted IoT device network-layer onboarding and lifecycle management implementations.

IT professionals who want to implement similar solutions will find the whole practice guide useful. You can use the how-to portion of the guide, *NIST SP 1800-36C*, to replicate all or parts of the builds created in our lab. The how-to portion of the guide provides specific product installation, configuration, and integration instructions for implementing the example solution. We do not re-create the product manufacturers' documentation, which is generally widely available. Rather, we show how we incorporated the products together in our environment to create an example solution. Also, you can use *Functional Demonstrations, NIST SP 1800-36D*, which provides the use cases that have been defined to showcase trusted IoT device network-layer onboarding and lifecycle management security capabilities and the results of demonstrating these use cases with each of the example implementations.

This guide assumes that IT professionals have experience implementing security products within the enterprise. While we have used a suite of commercial products to address this challenge, this guide does not endorse these particular products. Your organization can adopt this solution or one that adheres to these guidelines in whole, or you can use this guide as a starting point for tailoring and implementing parts of a trusted IoT device network-layer onboarding and lifecycle management solution. Your organization's security experts should identify the products that will best integrate with your existing tools and IT system infrastructure. We hope that you will seek products that are congruent with applicable standards and recommended practices.

A NIST Cybersecurity Practice Guide does not describe "the" solution, but example solutions. As the project progresses, this preliminary draft will be updated, and additional volumes will also be released for comment. We seek feedback on the publication's contents and welcome your input. Comments, suggestions, and success stories will improve subsequent versions of this guide. Please contribute your thoughts to iot-onboarding@nist.gov.

1.2 Build Overview

This NIST Cybersecurity Practice Guide addresses the challenge of network-layer onboarding using standards-based protocols to perform trusted network-layer onboarding of an IoT device. Each build demonstrates one or more of these capabilities:

- Trusted Network-Layer Onboarding: providing the device with its unique network credentials over an encrypted channel
- Network Re-Onboarding: performing trusted network-layer onboarding of the device again, after device reset
- Network Segmentation: assigning a device to a segment of the network
- Trusted Application-Layer Onboarding: providing the device with application-layer credentials over an encrypted channel after completing network-layer onboarding
- Ongoing Device Authorization: continuously monitoring the device on an ongoing basis, providing policy-based assurance and authorization checks on the device throughout its lifecycle

Currently, five builds that will serve as examples of how to onboard IoT devices using the protocols described in NIST SP 1800-36B are being implemented and will be demonstrated as part of this project. The remainder of this practice guide provides step-by-step instructions on how to reproduce the two builds that have been completed so far: Builds 1 and 2. Step-by-step instructions for Builds 3, 4, and 5, as well as a factory use case build, will be included in future updates to this document.

1.2.1 Reference Architecture Summary

The builds described in this document are instantiations of the trusted network-layer onboarding and lifecycle management logical reference architecture that is described in NIST SP 1800-36B. This architecture is organized according to five high-level processes: Device Manufacture and Factory Provisioning, Device Ownership and Bootstrapping Information Transfer, Trusted Network-Layer

Onboarding, Trusted Application-Layer Onboarding, and Continuous Assurance. For a full explanation of the architecture, please see NIST SP 1800-36B: *Approach, Architecture, and Security Characteristics*.

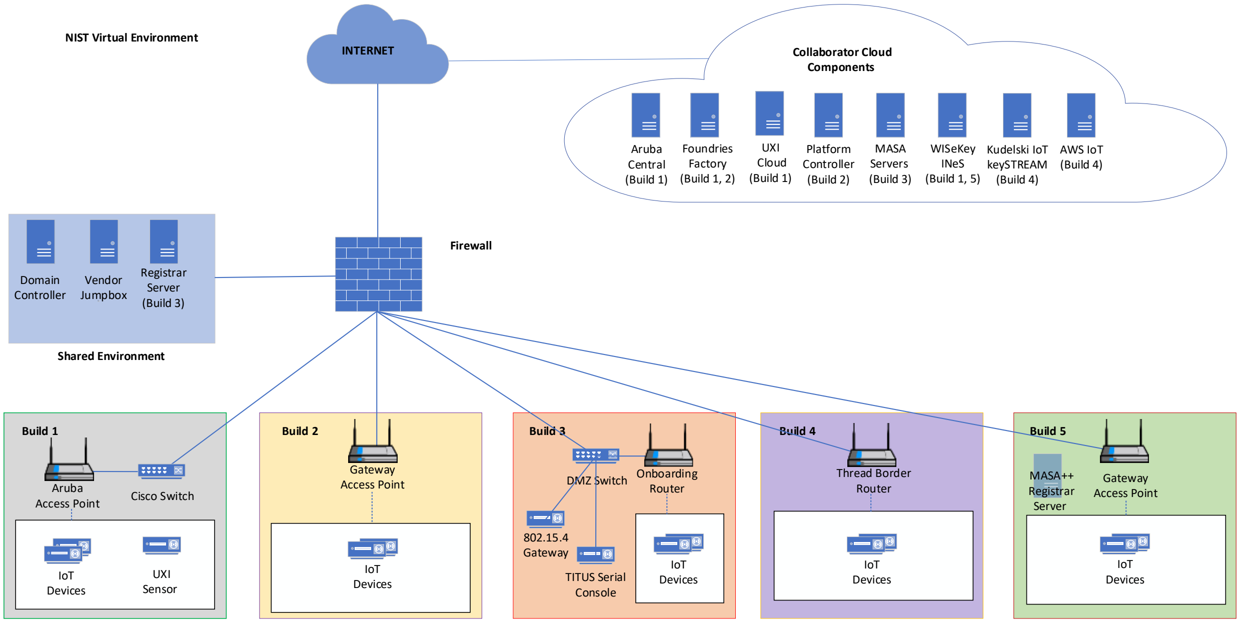
1.2.2 Physical Architecture Summary

Figure 1-1 depicts the high-level physical architecture of the NCCoE IoT Onboarding laboratory environment in which the five trusted IoT device network-layer onboarding project builds and the factory use case build are being implemented. The NCCoE provides virtual machine (VM) resources and physical infrastructure for the IoT Onboarding lab. As depicted, the NCCoE IoT Onboarding laboratory hosts collaborator hardware and software for the builds. The NCCoE also provides connectivity from the IoT Onboarding lab to the NIST Data Center, which provides connectivity to the internet and public IP spaces (both IPv4 and IPv6). Access to and from the NCCoE network is protected by a firewall.

Access to and from the IoT Onboarding lab is protected by a pfSense firewall, represented by the brick box icon in Figure 1-1. This firewall has both IPv4 and IPv6 (dual stack) configured. The IoT Onboarding lab network infrastructure includes a shared virtual environment that houses a domain controller and a vendor jumpbox. These components are used across builds where applicable. It also contains five independent virtual LANs, each of which houses a different trusted network-layer onboarding build.

The IoT Onboarding laboratory network has access to cloud components and services provided by the collaborators, all of which are available via the internet. These components and services include Aruba Central and the User Experience Insight (UXI) Cloud (Build 1), Platform Controller (Build 2), a Manufacturer Authorized Signing Authority (MASA) server (Build 3), Kudelski IoT keySTREAM application-layer onboarding service and Amazon Web Services (AWS) IoT (Build 4), and FoundriesFactory and WISeKey INeS, which we anticipate will be used across numerous builds.

Figure 1-1 NCCoE IoT Onboarding Laboratory Physical Architecture



All five network-layer onboarding laboratory environments, as depicted in the diagram, have been installed:

- The Build 1 network infrastructure within the NCCoE lab consists of two components: the Aruba Access Point and the Cisco Switch. Build 1 also requires support from Aruba Central for network-layer onboarding and the UXI Cloud for application-layer onboarding. These components are in the cloud and accessed via the internet. The IoT devices that are onboarded using Build 1 include the UXI Sensor and the Raspberry Pi.
- The Build 2 network infrastructure within the NCCoE lab consists of a single component: the Gateway Access Point. Build 2 also requires support from the Platform Controller, which also hosts the IoTivity Cloud Service. The IoT devices that are onboarded using Build 2 include three Raspberry Pis.
- The Build 3 network infrastructure components within the NCCoE lab include a Wi-Fi capable home router (including Join Proxy), a DMZ switch (for management), and an ESP32A Xtensa board acting as a Wi-Fi IoT device, as well as an nRF52840 board acting as an IEEE 802.15.4 device. A management system acts as a serial console (the “titus” machine). A registrar server (“minerva-fountain”) has been deployed as a virtual appliance on the NCCoE private cloud system. Build 3 also requires support from a MASA server which is accessed via the internet. In addition, an RPI3 (“satine”) provides an ethernet/802.15.4 gateway, as well as a test platform.

- The Build 4 network infrastructure components within the NCCoE lab include an Open Thread Border Router, which is implemented using a Raspberry Pi, and a Silicon Labs Gecko Wireless Starter Kit, which acts as an 802.15.4 antenna. Build 4 also requires support from the Kudelski IoT keySTREAM service, which is in the cloud and accessed via the internet. The IoT device that is onboarded in Build 4 is the Silicon Labs Thunderboard (BRD2601A) with an EFR32MG24 System-on-Chip. The application service to which it onboards is AWS IoT.
- The Build 5 network infrastructure components within the NCCoE lab include an OpenWRT router, a Turis Omnia Wi-Fi access point, the MASA++ Registration Server, and a USB hub. This build leverages the NquiringMinds' cloud service called tdx Volt in conjunction with the RADIUS service that resides on the router to provide authentication capabilities for network-layer onboarding to take place. The IoT device that is onboarded using Build 5 is a Feather HUZAH ESP8266.

The remainder of this guide will focus on the setup and configuration of Builds 1 and 2. Information for Builds 3, 4, and 5, as well as the factory use case build, are planned for future updates to this document.

1.3 Typographic Conventions

The following table presents typographic conventions used in this volume.

Typeface/Symbol	Meaning	Example
Italics	file names and path names; references to documents that are not hyperlinks; new terms; and placeholders	For language use and style guidance, see the *NCCoE Style Guide*.
Bold	names of menus, options, command buttons, and fields	Choose **File** > **Edit.**
`Monospace`	command-line input, onscreen computer output, sample code examples, and status codes	`mkdir`
`Monospace Bold`	command-line user input contrasted with computer output	**`service sshd start`**
blue text	link to other parts of the document, a web URL, or an email address	All publications from NIST's NCCoE are available at https://www.nccoe.nist.gov.

2 Build 1

This section of the practice guide contains detailed instructions for installing and configuring all the products used to build an instance of the example solution. For additional details on Build 1's logical and physical architectures, see NIST SP 1800-36B: *Approach, Architecture, and Security Characteristics*.

The network-layer onboarding component of Build 1 utilizes Wi-Fi Easy Connect, also known as the Device Provisioning Protocol (DPP). The Wi-Fi Easy Connect standard is maintained by the Wi-Fi Alliance [1]. The term "DPP" is used when referring to the network-layer onboarding protocol, and "Wi-Fi Easy Connect" is used when referring to the overall implementation of the network onboarding process.

2.1 Aruba Central/Hewlett Packard Enterprise (HPE) Cloud

This build utilized Aruba Central as a cloud management service that provided management and support for the Aruba Wireless Access Point (AP) and provided authorization and DPP onboarding capabilities for the wireless network. A cloud-based application programming interface (API) endpoint provided the ability to import the DPP Uniform Resource Identifiers (URIs) in the manner of a Supply Chain Integration Service. Due to this capability and Build 1's support for Wi-Fi Easy Connect, Build 1's infrastructure fully supported interoperable network-layer onboarding with Build 2's Reference Clients ("IoT devices") provided by CableLabs.

2.2 Aruba Wireless Access Point

Use of DPP is implicitly dependent on the Aruba Central cloud service. Aruba Central provides a cloud Infrastructure as a Service (IaaS) enabled architecture that includes initial support for DPP in Central 2.5.6 / ArubaOS (AOS) 10.4.0. Central and AOS support multiple deployment formats:

1. As AP only referred to as an underlay deployment, where traffic is bridged locally from the APs
2. An overlay deployment where all data is securely tunneled to an on-prem gateway where advanced services can route, inspect, and analyze the data before it's either bridged locally or routed to its next hop
3. A mixed-mode deployment, which is a combination of the two where a returned 'role/label' is used to determine how the data is processed and forwarded

With the initial release of DPP in 2022, Aruba supports underlay-mode. Support for overlay and mixed-mode is expected mid-2023.

At the time of this publication, a user can leverage any 3xx, 5xx, or 6xx APs to support a DPP deployment, with a view that all future series APs will implicitly include support. For an existing or new user there is a prerequisite of the creation of a Service Set Identifier (SSID). Note that DPP today is not supported under Wi-Fi Protected Access 3 (WPA3); this is a roadmap item with no current published timeline.

Assuming there is an existing SSID or a new one is created based upon the above security restrictions, the next step is to enable DPP (as detailed below in Section 2.2.1) such that the SSID can support multiple authentication and key managements (AKMs) on a Basic Service Set (BSS). If the chosen security type is DPP, only a single AKM will exist for that BSS.

Powering the AP from a standards-compliant 802.3at port is the easiest method. An external power supply can also be used.

Within this document, we do not cover the specifics of radio frequency (RF) design and placement of APs. Guidance and assistance is available within the Aruba community site, https://community.arubanetworks.com or the Aruba Support Portal, https://asp.arubanetworks.com.

Additionally, we do not cover onboarding and licensing of Aruba Central hardware. Documentation can be found here: https://www.arubanetworks.com/techdocs/ArubaDocPortal/content/docportal.htm.

2.2.1 Wi-Fi Network Setup/Configuration

The following instructions detail the initial setup and configuration of the Wi-Fi network upon powering on and connecting the AP to the network.

1. Navigate to the Aruba Central cloud management interface.
2. On the sidebar, navigate under **Global** and choose the AP-Group you want to configure/modify. (This assumes you have already grouped your APs by location/functions.)
3. Under **Devices**, click on **Config** in the top right side.
4. You will now be in the Access Points tab and WLANs tab. Click on **+ Add SSID** or float your cursor over the previously pre-created SSID name you wish to configure and click on the edit icon.
5. If creating a new SSID, after entering the Name (SSID) in Step 1 and configuring options as necessary in Step 2, when you get to Step 3 (Security), it will default on the slide-bar to the Personal Security Level. The alternative would be to choose the Enterprise Security Level. Both options are detailed below.
    a. If you choose the **Personal Security Level**, under **Key-Management** ensure you select either **DPP** or **WPA2-Personal**.
        i. If you choose **WPA2-Personal**, expand the **Advanced Settings** section and enable the toggle button for **DPP** so that the SSID can broadcast the AKM. Note that this option is not available if choosing DPP for Key-Management.
    b. If you choose the **Enterprise Security Level**:
        i. In this mode, only WPA2-Enterprise Key-Management currently supports DPP. Expand the **Advanced Settings** section and enable the toggle button for **DPP** so that the SSID can broadcast the AKM.
6. If you plan to enable DPP on a previously created SSID, ensure you are running version 10.4+ on your devices. You also need an SSID that is configured for WPA2-Personal or WPA2-Enterprise.

   a. Edit the SSID, click on **Security,** and expand the **Advanced Settings section** and enable the toggle button for **DPP**.

   b. Click **Save Settings**.

For SSIDs that have been modified to add DPP AKM, it's also necessary to enable DPP within the radio profile.

1. Under the **Access Point** Tab, click on **Radios.**
2. It's expected you'll see a **default** radio-profile. If a custom one has been created, you'll need to review your configuration before proceeding.
3. Assuming a **default** radio-profile, click on the **Edit** icon, expand **Show advanced settings,** and scroll down to **DPP Provisioning.** You can selectively enable this for 2.4 GHz or 5.0 GHz. Support for DPP on 6.0 GHz is a roadmap item at this time and is not yet available.

2.2.2 Wi-Fi Easy Connect Configuration

Configuration of the Access Point occurred through the Aruba Central cloud management interface. Standard configurations were used to stand up the Build 1 wireless network. The instructions for enabling DPP capabilities for the overall wireless network are listed below:

1. Navigate to the Aruba Central cloud management interface.
2. On the sidebar, navigate to **Security > Authentication and Policy > Config**.
3. In the **Client Access Policy** section, click **Edit**.
4. Under the **Wi-Fi Easy Connect™ Service** heading, ensure that the name of your wireless network is selected.
5. Click **Save**.

2.3 Cisco Catalyst 3850-S Switch

This build utilized a Cisco Catalyst 3850-S switch. This switch utilized a minimal configuration with two separate virtual local area networks (VLANs) to allow for IoT device network segmentation and access control. The switch also provided Power-over-Ethernet support for the Aruba Wireless AP.

2.3.1 Configuration

The switch was configured with two VLANs, and a trunk port dedicated to the Aruba Wireless AP. You can find the relevant portions of the Cisco iOS configuration below:

```
interface Vlan1
```

```
 no ip address
interface Vlan2
 no ip address
interface GigabitEthernet1/0/1
 switchport mode trunk
interface GigabitEthernet1/0/2
 switchport mode access
 switchport access vlan 1
interface GigabitEthernet1/0/3
 switchport mode access
 switchport access vlan 2
```

2.4 Aruba User Experience Insight (UXI) Sensor

This build utilized an Aruba UXI Sensor as a Wi-Fi Easy Connect-capable IoT device. Model G6 and G6C support Wi-Fi Easy Connect, and all available G6 and G6C models support Wi-Fi Easy Connect within their software image. This sensor successfully utilized the network-layer onboarding mechanism provided by the wireless network and completed onboarding to the application-layer UXI cloud service. The network-layer onboarding process is automatically initiated by the device on boot.

2.4.1 Configuration

All Aruba's available G6 and G6C UXI sensors support the ability to complete network-layer and application-layer onboarding. No specific configuration of the physical sensor is required. As part of the supply-chain process, the cryptographic public-key for your sensor(s) will be available within the cloud tenant. This public/private-key-pair for each device is created as part of the manufacturing process. The public-key effectively identifiers the sensor to the network and as part of the Wi-Fi Easy Connect/DPP onboarding process. This allows unprovisioned devices straight from the factory to be onboarded and subsequently connect to the UXI sensor cloud to obtain their network-layer configuration. An administrator will have to define the 'tasks' the UXI sensor is going to perform such as monitoring SSIDs, performing reachability tests to on-prem or cloud services, and making the results of these tests available within the UXI user/administrator portal.

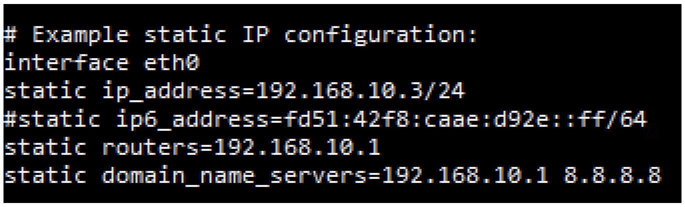
2.5 Raspberry Pi

In this build, the Raspberry Pi 3B+ acts as a DPP enrollee. In setting up the device for this build, a DPP-capable wireless adapter, the Alfa AWUS036NHA network dongle, was connected to enable the Pi to send and receive DPP frames. Once fully configured, the Pi can onboard with the Aruba AP.

2.5.1 Configuration

The following steps were completed for the Raspberry Pi to complete DPP onboarding:

1. Set the management IP for the Raspberry Pi to an IP address in the Build 1 network. To do this, add the following lines to the file *dhcpcd.conf* located at `/etc/dhcpcd.conf`. For this build, the IP address was set to 192.168.10.3.



```
# Example static IP configuration:
interface eth0
static ip_address=192.168.10.3/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
static routers=192.168.10.1
static domain_name_servers=192.168.10.1 8.8.8.8
```

2. Install Linux Libraries using the apt package manager. The following packages were installed:
   a. autotools-dev
   b. automake
   c. libcurl4-openssl-dev
   d. libnl-genl-3-dev
   e. libavahi-client-dev
   f. libavahi-core-dev
   g. aircrack-ng
   h. openssl-1.1.1q
3. Install the DPP utilities. These utilities were installed from the GitHub repository https://github.com/HewlettPackard/dpp using the following command:

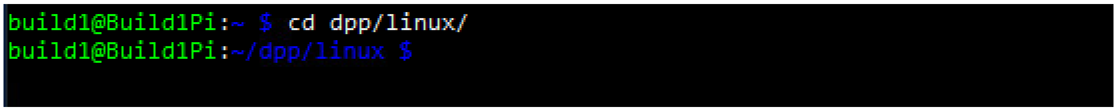
```
git clone https://github.com/HewlettPackard/dpp
```

2.5.2 DPP Onboarding

This section describes the steps for using the Raspberry Pi as a DPP enrollee. The Pi uses a DPP utility to send out chirps to make its presence known to available DPP configurators. Once the Pi is discovered, the DPP configurator (Aruba Wireless AP) initiates the DPP authentication protocol. During this phase, DPP *connectors* are created to onboard the device to the network. As soon as the Pi is fully authenticated, the Pi is fully enrolled and can begin normal network communication.

1. Navigate to the DPP utilities directory which was installed during setup:

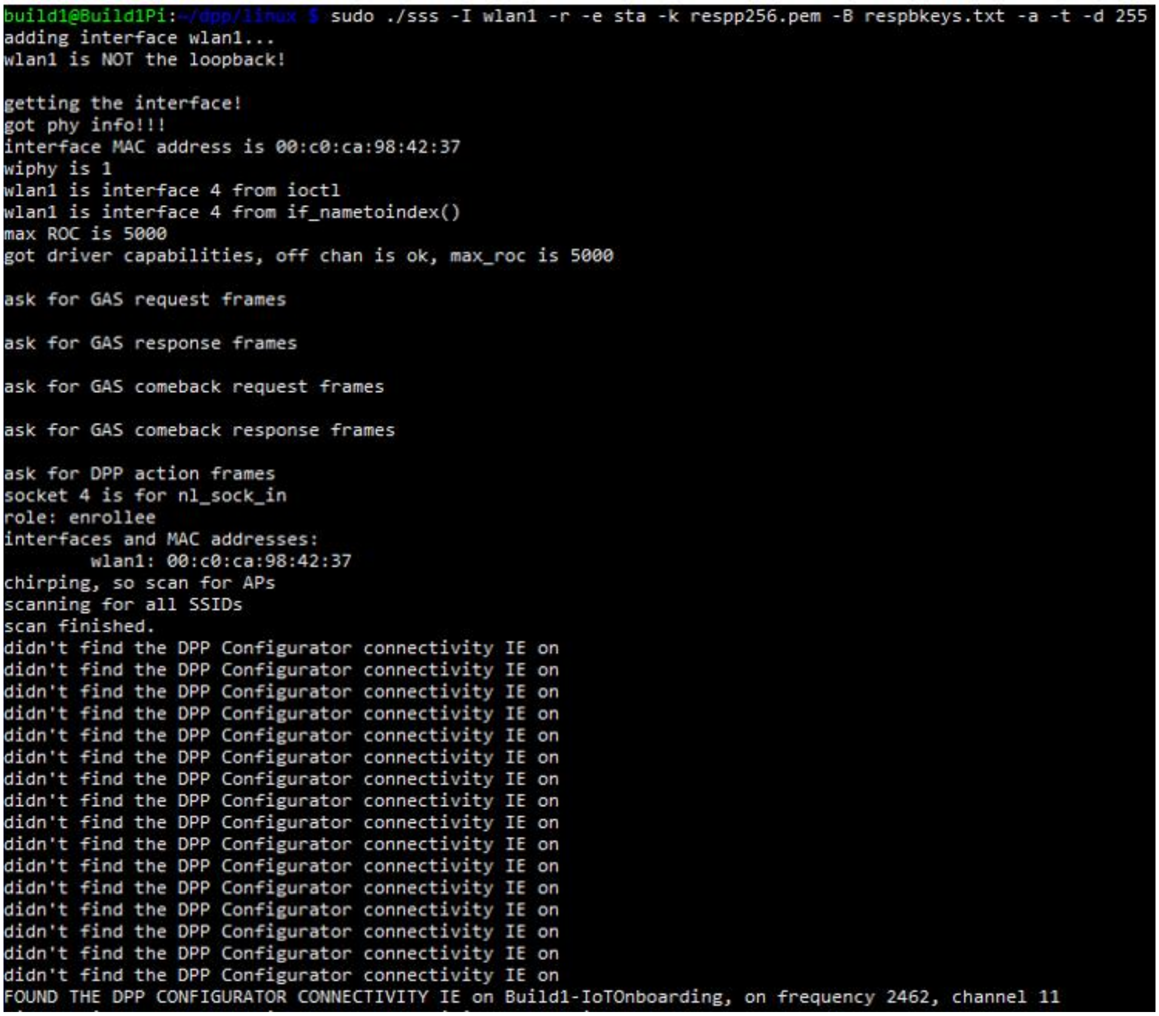
```
cd dpp/linux
```



```
build1@Build1Pi:~ $ cd dpp/linux/
build1@Build1Pi:~/dpp/linux $
```

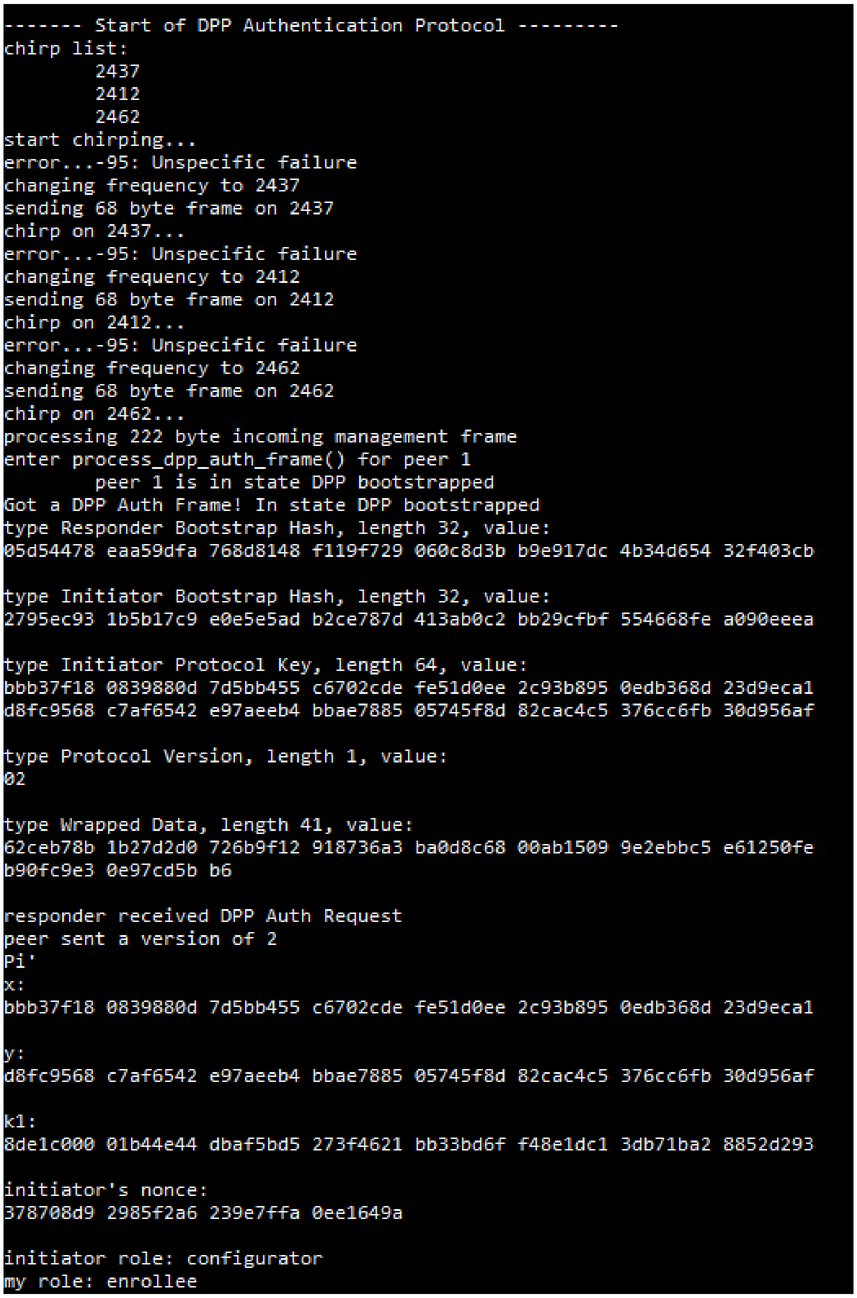
2. From the DPP utilities directory, run the following command to initiate a DPP connection:

```
sudo ./sss -I wlan1 -r -e sta -k respp256.pem -B respbkeys.txt -a -t -d 255
```



```
build1@Build1Pi:~/dpp/linux $ sudo ./sss -I wlan1 -r -e sta -k respp256.pem -B respbkeys.txt -a -t -d 255
adding interface wlan1...
wlan1 is NOT the loopback!

getting the interface!
got phy info!!!
interface MAC address is 00:c0:ca:98:42:37
wiphy is 1
wlan1 is interface 4 from ioctl
wlan1 is interface 4 from if_nametoindex()
max ROC is 5000
got driver capabilities, off chan is ok, max_roc is 5000

ask for GAS request frames

ask for GAS response frames

ask for GAS comeback request frames

ask for GAS comeback response frames

ask for DPP action frames
socket 4 is for nl_sock_in
role: enrollee
interfaces and MAC addresses:
        wlan1: 00:c0:ca:98:42:37
chirping, so scan for APs
scanning for all SSIDs
scan finished.
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
didn't find the DPP Configurator connectivity IE on
FOUND THE DPP CONFIGURATOR CONNECTIVITY IE on Build1-IoTOnboarding, on frequency 2462, channel 11
```

3. Once the enrollee has found a DPP configurator, the DPP authentication protocol is initiated.



```
------- Start of DPP Authentication Protocol ---------
chirp list:
        2437
        2412
        2462
start chirping...
error...-95: Unspecific failure
changing frequency to 2437
sending 68 byte frame on 2437
chirp on 2437...
error...-95: Unspecific failure
changing frequency to 2412
sending 68 byte frame on 2412
chirp on 2412...
error...-95: Unspecific failure
changing frequency to 2462
sending 68 byte frame on 2462
chirp on 2462...
processing 222 byte incoming management frame
enter process_dpp_auth_frame() for peer 1
        peer 1 is in state DPP bootstrapped
Got a DPP Auth Frame! In state DPP bootstrapped
type Responder Bootstrap Hash, length 32, value:
05d54478 eaa59dfa 768d8148 f119f729 060c8d3b b9e917dc 4b34d654 32f403cb

type Initiator Bootstrap Hash, length 32, value:
2795ec93 1b5b17c9 e0e5e5ad b2ce787d 413ab0c2 bb29cfbf 554668fe a090eeea

type Initiator Protocol Key, length 64, value:
bbb37f18 0839880d 7d5bb455 c6702cde fe51d0ee 2c93b895 0edb368d 23d9eca1
d8fc9568 c7af6542 e97aeeb4 bbae7885 05745f8d 82cac4c5 376cc6fb 30d956af

type Protocol Version, length 1, value:
02

type Wrapped Data, length 41, value:
62ceb78b 1b27d2d0 726b9f12 918736a3 ba0d8c68 00ab1509 9e2ebbc5 e61250fe
b90fc9e3 0e97cd5b b6

responder received DPP Auth Request
peer sent a version of 2
Pi'
x:
bbb37f18 0839880d 7d5bb455 c6702cde fe51d0ee 2c93b895 0edb368d 23d9eca1

y:
d8fc9568 c7af6542 e97aeeb4 bbae7885 05745f8d 82cac4c5 376cc6fb 30d956af

k1:
8de1c000 01b44e44 dbaf5bd5 273f4621 bb33bd6f f48e1dc1 3db71ba2 8852d293

initiator's nonce:
378708d9 2985f2a6 239e7ffa 0ee1649a

initiator role: configurator
my role: enrollee
```

2.6 Certificate Authority

The function of the certificate authority (CA) in this build is to issue network credentials for use in the network-layer onboarding process.

2.6.1 Private Certificate Authority

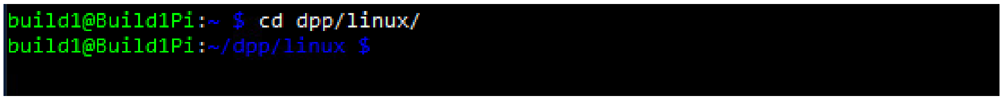
A private CA was provided as a part of the DPP demonstration utilities in the HPE GitHub repository. For demonstration purposes, the Raspberry Pi is used as the configurator and the enrollee.

2.6.1.1 Installation/Configuration

The following instructions detail the initial setup and configuration of the private CA using the DPP demonstration utilities and certificates located at https://github.com/HewlettPackard/dpp.

1. Navigate to the DPP utilities directory on the Raspberry Pi: `~dpp/linux`

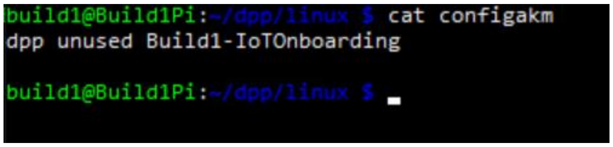
   ```
   cd dpp/linux/
   ```

   ```
   build1@Build1Pi:~ $ cd dpp/linux/
   build1@Build1Pi:~/dpp/linux $
   ```

2. The README in the GitHub repository (https://github.com/HewlettPackard/dpp/blob/master/README) references a text file called *configakm* which contains information about the network policies for a configurator to provision on an enrollee. The format is: `<akm> <EAP server> <ssid>`. Current AKMs that are supported are DPP, dot1x, sae, and psk. For this build, DPP is used. For DPP, an Extensible Authentication Protocol (EAP) server is not used.

3. Configure the file *configakm* located in `~/dpp/linux/`. This file instructs the configurator on how to deploy a DPP connector (network credential) from the configurator to the enrollee. As shown below, the *configakm* file is filled with the following fields: `dpp unused Build1-IoTOnboarding`.

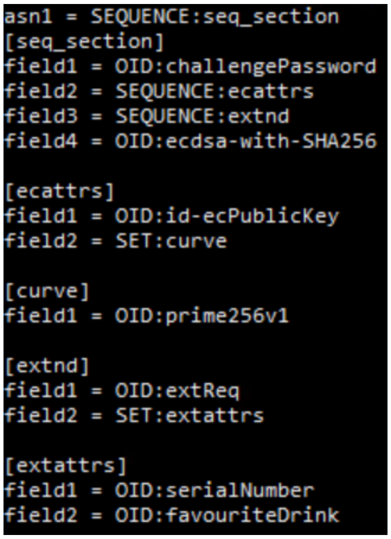
   

   ```
   build1@Build1Pi:~/dpp/linux $ cat configakm
   dpp unused Build1-IoTOnboarding

   build1@Build1Pi:~/dpp/linux $
   ```

4. The file *csrattrs.conf* contains attributes to construct an Abstract Syntax Notation One (ASN.1) string. This string allows the configurator to tell the enrollee how to generate a certificate signing request (CSR). The following fields were used for this demonstration:

   ```
   asn1 = SEQUENCE: seq_section
   [seq_section]
   field1 = OID:challengePassword
   field2 = SEQUENCE:ecattrs
   field3 = SEQUENCE:extnd
   ```

```
field4 = OID:ecdsa-with-SHA256

[ecattrs]
field1 = OID:id-ecPublicKey
field2 = SET:curve

[curve]
field1 = OID:prime256v1

[extnd]
field1 – OID:extReq
field2 = SET:extattrs

[extattrs]
field1 = OID:serialNumber
field2 = OID:favouriteDrink

```



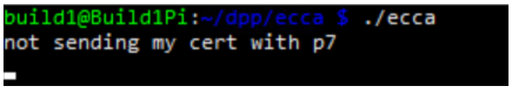
```
asn1 = SEQUENCE:seq_section
[seq_section]
field1 = OID:challengePassword
field2 = SEQUENCE:ecattrs
field3 = SEQUENCE:extnd
field4 = OID:ecdsa-with-SHA256

[ecattrs]
field1 = OID:id-ecPublicKey
field2 = SET:curve

[curve]
field1 = OID:prime256v1

[extnd]
field1 = OID:extReq
field2 = SET:extattrs

[extattrs]
field1 = OID:serialNumber
field2 = OID:favouriteDrink
```

2.6.1.2 Operation/Demonstration

Once setup and configuration have been completed, the following steps can be used to demonstrate utilizing the private CA to issue credentials to a requesting device.

1. Open three terminals on the Raspberry Pi: one to start the certificate program, one to show the configurator's point of view, and one to show the enrollee's point of view.

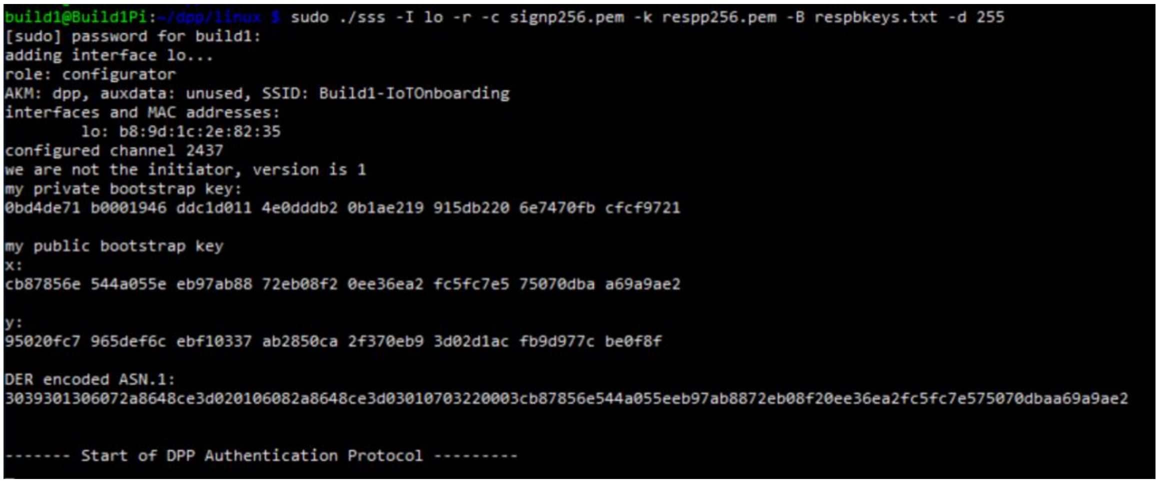
2. The demonstration uses an OpenSSL certificate. To run the program from the first terminal, navigate to the following directory: `~/dpp/ecca/`, and run the command: `./ecca`.



```
build1@Build1Pi:~/dpp/ecca $ ./ecca
not sending my cert with p7
```

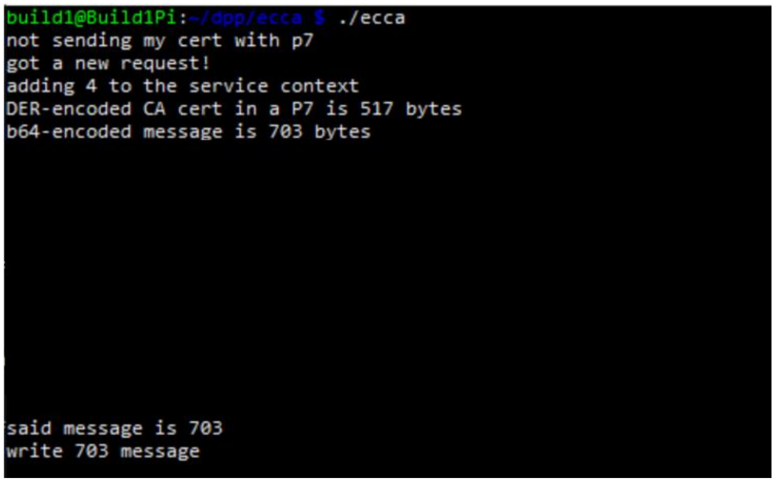
3. On the second terminal, start the configurator using the following command:

```
sudo ./sss -I lo -r -c signp256.pem -k respp256.pem -B resppbkeys.txt -d 255
```



```
build1@Build1Pi:~/dpp/linux $ sudo ./sss -I lo -r -c signp256.pem -k respp256.pem -B respbkeys.txt -d 255
[sudo] password for build1:
adding interface lo...
role: configurator
AKM: dpp, auxdata: unused, SSID: Build1-IoTOnboarding
interfaces and MAC addresses:
        lo: b8:9d:1c:2e:82:35
configured channel 2437
we are not the initiator, version is 1
my private bootstrap key:
0bd4de71 b0001946 ddc1d011 4e0dddb2 0b1ae219 915db220 6e7470fb cfcf9721

my public bootstrap key
x:
cb87856e 544a055e eb97ab88 72eb08f2 0ee36ea2 fc5fc7e5 75070dba a69a9ae2

y:
95020fc7 965def6c ebf10337 ab2850ca 2f370eb9 3d02d1ac fb9d977c be0f8f

DER encoded ASN.1:
3039301306072a8648ce3d020106082a8648ce3d03010703220003cb87856e544a055eeb97ab8872eb08f20ee36ea2fc5fc7e575070dbaa69a9ae2


------- Start of DPP Authentication Protocol ---------
```

As shown in the terminal where the ecca program is running, the configurator contacts the CA and asks for the certificate.

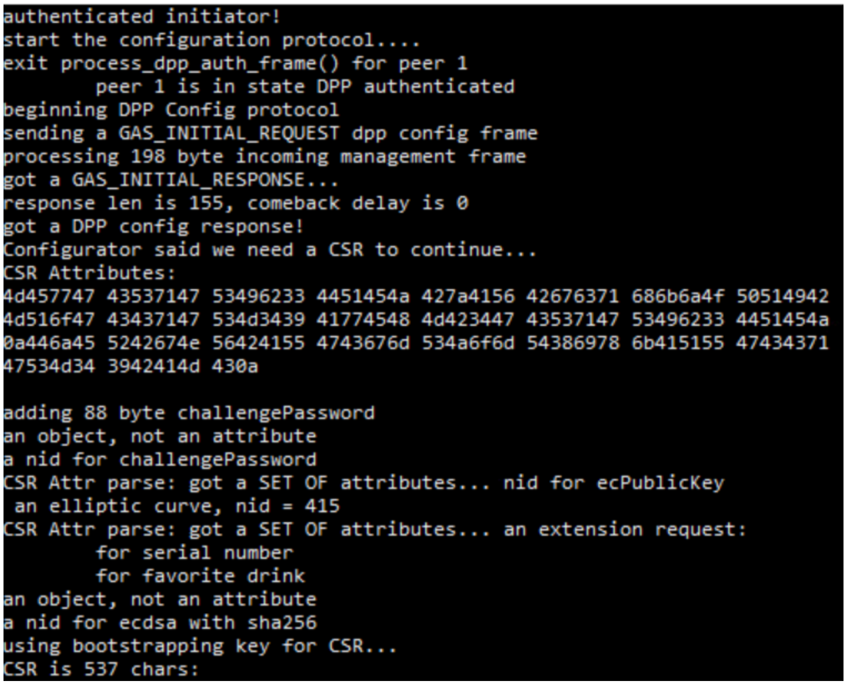


```
build1@Build1Pi:~/dpp/ecca $ ./ecca
not sending my cert with p7
got a new request!
adding 4 to the service context
DER-encoded CA cert in a P7 is 517 bytes
b64-encoded message is 703 bytes









said message is 703
write 703 message
```

4. On the third terminal, start the enrollee using the following command:

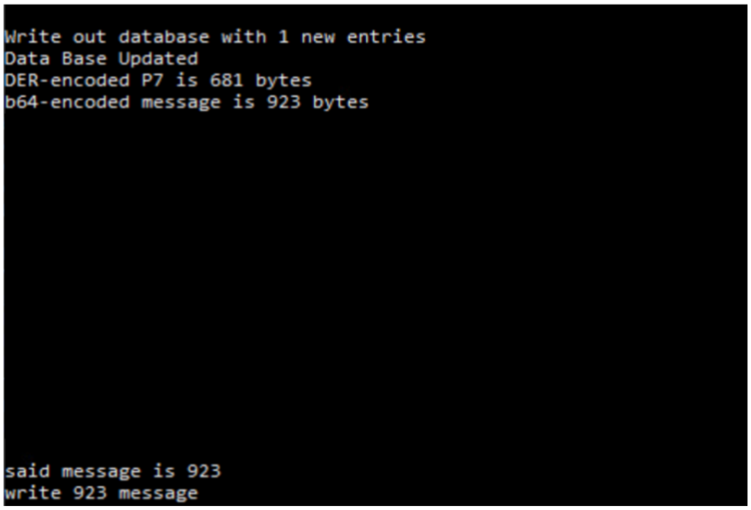
   ```
   sudo ./sss -I lo -r -e sta -k initp256.pem -B initbkeys.txt -t -a -q -d 255
   ```

   From the enrollee's perspective, it will send chirps on different channels until it finds the configurator. Once found, it sends its certificate to the CA for signing. The snippet below is of the enrollee generating the CSR.



```
authenticated initiator!
start the configuration protocol....
exit process_dpp_auth_frame() for peer 1
        peer 1 is in state DPP authenticated
beginning DPP Config protocol
sending a GAS_INITIAL_REQUEST dpp config frame
processing 198 byte incoming management frame
got a GAS_INITIAL_RESPONSE...
response len is 155, comeback delay is 0
got a DPP config response!
Configurator said we need a CSR to continue...
CSR Attributes:
4d457747 43537147 53496233 4451454a 427a4156 42676371 686b6a4f 50514942
4d516f47 43437147 534d3439 41774548 4d423447 43537147 53496233 4451454a
0a446a45 5242674e 56424155 4743676d 534a6f6d 54386978 6b415155 47434371
47534d34 3942414d 430a

adding 88 byte challengePassword
an object, not an attribute
a nid for challengePassword
CSR Attr parse: got a SET OF attributes... nid for ecPublicKey
 an elliptic curve, nid = 415
CSR Attr parse: got a SET OF attributes... an extension request:
        for serial number
        for favorite drink
an object, not an attribute
a nid for ecdsa with sha256
using bootstrapping key for CSR...
CSR is 537 chars:
```

5. In the ECCA terminal, the certificate from the enrollee is shown:



```
Write out database with 1 new entries
Data Base Updated
DER-encoded P7 is 681 bytes
b64-encoded message is 923 bytes













said message is 923
write 923 message
```

2.6.2 WISeKey INeS

The WISeKey INeS Certificate Management System provides CA and certificate management capabilities for Build 1. Implementation of this system will provide Build 1 with a trusted, public CA to support issuing network credentials. This collaboration is in progress and will be described in a future version of this document.

2.7 UXI Cloud

The UXI Cloud is a web-based application that serves as a monitoring hub for the UXI sensor. It provides visibility into the data captured by the performance monitoring that the UXI sensor conducts. For the purposes of this build, the dashboard was used to demonstrate application-layer onboarding, which occurs once the UXI sensor has completed network-layer onboarding. Once application-layer onboarding was completed and the application configuration had been applied to the device, our demonstration concluded.

3 Build 2

This section of the practice guide contains detailed instructions for installing and configuring all of the products used to build an instance of the example solution. For additional details on Build 2's logical and physical architectures, see NIST SP 1800-36B: *Approach, Architecture, and Security Characteristics*.

The network-layer onboarding component of Build 2 utilizes Wi-Fi Easy Connect, also known as the Device Provisioning Protocol (DPP). The Wi-Fi Easy Connect standard is maintained by the Wi-Fi Alliance [1]. The term "DPP" is used when referring to the network-layer onboarding protocol, and "Wi-Fi Easy Connect" is used when referring to the overall implementation of the network onboarding process.

3.1 CableLabs Platform Controller

The CableLabs Platform Controller provides an architecture and reference implementation of a cloud-based service that provides management capability for service deployment groups, access points with the deployment groups, registration and lifecycle of user services, and the secure onboarding and lifecycle management of users' Wi-Fi devices. The controller also exposes APIs for integration with third-party systems for the purpose of integrating various business flows (e.g., integration with manufacturing process for device management).

The Platform Controller would typically be hosted by the network operator or a third-party service provider. It can be accessed via web interface. Additional information for this deployment can be accessed at the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Ref-AP-Setup-for-NCCoE/nccoe-ap-setup.md.

3.1.1 Operation/Demonstration

Once configuration of the Platform Controller, Gateway, and Reference Client have been completed, full operation can commence. Instructions for this are located at the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Raspberry_Pi_Deployment.md.

3.2 CableLabs Custom Connectivity Gateway

In this deployment, the gateway software is running on a Raspberry Pi 3B+, which acts as a router, firewall, wireless access point, Open Connectivity Foundation (OCF) Diplomat, and OCF Onboarding Tool. The gateway is also connected to the CableLabs Platform Controller, which manages much of the configuration and functions of the gateway. Due to Build 2's infrastructure and support of Wi-Fi Easy Connect, Build 2 fully supported interoperable network-layer onboarding with Build 1's IoT devices.

3.2.1 Installation/Configuration

Hardware requirements, pre-installation steps, installation steps, and configuration instructions for the gateway can be found at the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Ref-AP-Setup-for-NCCoE/nccoe-ap-setup.md.

3.2.2 Integration with CableLabs Platform Controller

Once initial configuration has occurred, the gateway can be integrated with the CableLabs Platform Controller. Instructions can be found at the official CableLabs repository:

https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Ref-AP-Setup-for-NCCoE/nccoe-ap-setup.md

3.2.3 Operation/Demonstration

Once configuration of the Platform Controller, Gateway, and Reference Client have been completed, full operation can commence. Instructions for this are located at the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Raspberry_Pi_Deployment.md.

3.3 Reference Clients/IoT Devices

Three reference clients were deployed in this build, each on a Raspberry Pi 3B+. They were each configured to emulate either a smart light switch or a smart lamp. The software deployed also included the capability to perform network-layer onboarding via Wi-Fi Easy Connect/DPP, and application-layer onboarding using the OCF onboarding method. These reference clients were fully interoperable with network-layer onboarding to Build 1.

3.3.1 Installation/Configuration

Hardware requirements, pre-installation, installation, and configuration steps for the reference clients are detailed in the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Raspberry_Pi_Deployment.md.

3.3.2 Operation/Demonstration

Once configuration of the Platform Controller, Gateway, and Reference Client have been completed, full operation can commence. Instructions for this are located at the official CableLabs repository: https://github.com/cablelabs/Streamlined_Onboarding_Demo/blob/nccoe-release/docs/Raspberry_Pi_Deployment.md.

For interoperability with Build 1, the IoT device’s DPP URI was provided to Aruba Central, which allowed Build 1 to successfully complete network-layer onboarding with the Build 2 IoT devices.

4 Build 3

In future releases of this practice guide, this section will contain detailed instructions for installing and configuring all of the products used to build an instance of the example solution.

5 Build 4

In future releases of this practice guide, this section will contain detailed instructions for installing and configuring all of the products used to build an instance of the example solution.

6 Build 5

In future releases of this practice guide, this section will contain detailed instructions for installing and configuring all of the products used to build an instance of the example solution.

7 Factory Use Case Build

In future releases of this practice guide, this section will contain detailed instructions for installing and configuring all of the products used to build an instance of the example solution.

Appendix A List of Acronyms

AKM	Authentication and Key Management
AOS	ArubaOS
AP	Access Point
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
AWS	Amazon Web Services
BSS	Basic Service Set
CA	Certificate Authority
CRADA	Cooperative Research and Development Agreement
CSR	Certificate Signing Request
DMZ	Demilitarized Zone
DPP	Device Provisioning Protocol (Wi-Fi Easy Connect)
EAP	Extensible Authentication Protocol
HPE	Hewlett Packard Enterprise
IaaS	Infrastructure as a Service
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
LAN	Local Area Network
MASA	Manufacturer Authorized Signing Authority
MUD	Manufacturer Usage Description
NCCoE	National Cybersecurity Center of Excellence
NIST	National Institute of Standards and Technology
OCF	Open Connectivity Foundation

RF	Radio Frequency
SP	Special Publication
SSID	Service Set Identifier
URI	Uniform Resource Identifier
USB	Universal Serial Bus
UXI	User Experience Insight
VM	Virtual Machine
VLAN	Virtual Local Area Network
WLAN	Wireless Local Area Network
WPA2	Wi-Fi Protected Access 2
WPA3	Wi-Fi Protected Access 3

Appendix B References

[1] Wi-Fi Alliance. *Wi-Fi Easy Connect*. Available: https://www.wi-fi.org/discover-wi-fi/wi-fi-easy-connect.