

IOT Operations
Internet-Draft
Intended status: Experimental
Expires: 11 August 2025

M. Sweet, Ed.
Lakeside Robotics Corporation
7 February 2025

ACME-Based Provisioning of IoT Devices
draft-sweet-iot-acme-07

Abstract

This document extends the Automatic Certificate Management Environment (ACME) to provision X.509 certificates for local Internet of Things (IoT) devices that are accepted by existing web browsers and other software running on End User client devices.

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 11 August 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	2
1.1. Connecting to the Network	3
1.2. Trusting IoT Devices on the Network	3
2. Terminology	4
3. Specification	5
3.1. Default Host Name Convention	5
3.2. Secure Network Connection	5
3.3. ACME Server Discovery	5
3.4. ACME Server Extensions	6
3.4.1. Root (CA) Certificate	6
3.4.2. Accounts	7
3.4.3. IoT Device Certificate Signing Requests	7
3.4.4. Manual Approval of Signing Requests	7
3.4.5. IoT Device Certificates	7
3.5. Client Device Configuration	8
3.6. IoT Device Configuration	8
4. Security Considerations	8
4.1. Trusting Local DHCP and DNS Infrastructure	9
4.2. Certificate Signing Request Validation	9
4.3. Man-in-the-Middle Attacks	9
4.4. Storage of Key Material	9
4.5. Revocation and Reissuance/Regeneration	9
4.6. Use of mDNS	10
4.7. mDNS Domain Name Collisions	10
4.8. Network Identification and Validation	10
4.9. Multiple Network Support	10
4.10. Protection of Certificates and Key Material	11
4.11. Reuse of Key Material	11
5. IANA Considerations	11
5.1. DHCP Option	11
5.2. Service Name	11
6. Normative References	12
7. Informative References	14
Appendix A. Change History	14
Author's Address	15

1. Introduction

IoT Devices are common on local networks and often utilize TLS [RFC8446] with self-signed X.509 certificates [RFC5280] to provide HTTPS [RFC9110] based web pages and services. Unfortunately, web browsers typically do not trust such certificates and show error messages intended to deter usage. Some IoT Devices also have manufacturer-supplied X.509 certificates, however due to the service life of such devices, the need for crypto-agility, and well-known challenges of secure key management, those certificates are better

suited to attestation and secure network connection than direct use with TLS.

The goal of this document is to provide a method for providing trusted X.509 certificates for use with TLS that does not depend on an Internet connection or a service provided by the original device manufacturer, does not depend on credentials or key material provided with the device from the factory, allows a device to be moved to different networks or domains without interaction with the device manufacturer, and supports the needs of both simple home networks and complex enterprise networks.

Because of obvious security concerns on public or untrusted networks, this method is only used on private (home, business, etc.) networks that have at least basic access restrictions - a password or pre-shared key, physical access (Ethernet), and so forth.

1.1. Connecting to the Network

This document is not concerned with the method of connecting Client or IoT Devices to the network. That said, the level of security and/or trust of the network and the proposed solution necessarily depends on the access controls, confidentiality, and validation provided by the network, as well as the quality of any identification and/or key material supplied by the manufacturer.

1.2. Trusting IoT Devices on the Network

This document uses existing infrastructure, namely the network's Extensible Authentication Protocol (EAP) [RFC3748] server to establish IoT device trust and the network's DHCP [RFC2131] and DNS [RFC1034] services to discover the local Automatic Certificate Management Environment (ACME) [RFC8555] Server to use for that network. Local ACME Servers are discovered using either a DHCP option or a DNS-SD [RFC6763] service record from the network's DNS service, and the local ACME Server's X.509 certificate provides a usable and verifiable network identifier as well as the trust anchor for issued IoT Device X.509 certificates.

ACME defines a protocol for network services to obtain trusted X.509 credentials for use with TLS [RFC8446]. However, since existing ACME Servers depend on public Internet connectivity to the ACME Client for validation, and since those same servers cannot issue X.509 certificates for the ".local" domain, some changes are needed to support a local ACME Server. X.509 certificates issued by the local ACME server are only valid when accessing the IoT Device for the local DNS domain, the mDNS (".local") domain, or any link-local or private IP addresses. Local ACME Servers can be standalone servers (common in enterprise networks) or software that runs on a consumer Internet router/modem.

Because devices often connect to multiple, unconnected networks, trust and usage of X.509 certificates provided by a local ACME server is limited to that network, essentially creating an intermediate trust level below global Certificate Authorities (CAs).

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

ACME Client: A device that uses the ACME protocol to request certificate management actions, such as issuance or revocation.

ACME Server: A device that implements the ACME protocol to respond to ACME Client requests, performing the requested actions if the client is authorized.

Certificate Authority (CA): A trusted source for X.509 certificates used during negotiation of a TLS session. (TODO: Update from current TLS/X.509 specifications)

Client Device: A computer, tablet, phone, or other End User device that accesses an IoT Device.

End User: A person or software process that is authorized to use Client Devices and, through the Client Device, access and use IoT Devices.

IoT Device: A camera, printer, switch, or other local device that provides services or functions to a Client Device.

Media Access Control (MAC) Address: A unique identifier assigned to a network interface controller for use as a network address in communications within a network segment.

Private Network: A network that requires some level of authorization and/or physical access to join such as an Ethernet network or a Wi-Fi network with a password or pre-shared key.

Public Network: A network that does not require any level of authorization and/or physical access to join such as a cellular network or a public or guest Wi-Fi network with no password or pre-shared key at a cafe.

Service Set Identifier (SSID): The name associated with a wireless network.

Trust On First Use (TOFU): An unauthenticated public key obtained on first contact (and retained for future use) will be good enough to secure future communication [RFC7435].

Uniform Resource Identifier (URI): A compact sequence of characters that identifies an abstract or physical resource [RFC3986].

3. Specification

3.1. Default Host Name Convention

IoT devices SHOULD use a default host name composed of a vendor prefix followed by a unique identifier such as a serial number or MAC address, e.g., "ex123456abcdef" for an "Example" IoT device with a MAC address of 12:34:56:ab:cd:ef. This name can be used as a peer identifier for secure network connection and as a label in an mDNS and/or DNS name, e.g., "ex123456abcdef.local" and "ex123456abcdef.example.com". The use of unique identifiers minimizes the risk of accidental name collisions.

3.2. Secure Network Connection

Client and IoT devices SHOULD securely connect to networks using IEEE 802.1X Extensible Authentication Protocol (EAP) [RFC3748], EAP-TLS Authentication Protocol [RFC5216], and the EAP Key Management Framework [RFC5247]. IoT devices that do not support these protocols MUST support stable MAC addresses to enable manual authorization of the device on the network.

Note: IoT devices with randomized MAC addresses would require manual authorization every time the device powers up.

3.3. ACME Server Discovery

Client and IoT devices discover the local ACME Server using one of two methods (in order of precedence):

1. Via DHCP Option NNN (ACME Server) when obtaining IPv4/IPv6 addresses. Note: DHCP Option 60 (Vendor Class Identifier [RFC3925]) with enterprise number 55357 (Lakeside Robotics Corporation) shall be used for purposes of prototyping this document.
2. Via a subsequent DNS-SD query sent to the configured DNS server and domain for the "_acme-server._tcp.domain" SRV record.

Home networks will typically use the DHCP Option while larger (enterprise) networks providing a dedicated DNS domain will use the DNS-SD method.

Note: DNS-SD queries MUST NOT be performed using Multicast DNS (mDNS) [RFC6762] for security reasons.

3.4. ACME Server Extensions

ACME [RFC8555] defines a protocol for managing trusted X.509 certificates. Organizations such as "Let's Encrypt" provide publicly available ACME servers, and such servers have led to the ubiquitous usage of TLS for internet web and email servers. However, public ACME servers typically cannot access local (private) devices and will not issue certificates for the mDNS ".local" domain. A local ACME server can both access local devices and issue certificates for local domains.

3.4.1. Root (CA) Certificate

A local ACME server will typically generate a self-signed X.509 certificate as its root (CA) certificate and the local network's trust anchor. The certificate MUST use a SHA2 hash of at least 256 bits and MUST use either RSA encryption with a key length of at least 3072 bits or ECDSA encryption with the secp384r1 (P-384) or secp521r1 (P-521) curves. The expiration of the self-signed certificate MUST be between 1 and 10 years, inclusive. The certificate MUST contain subjectAltName extensions for the mDNS (.local) and local domain name(s), and MAY contain subjectAltName extensions for the current IP address(es) of the server. For example, if the local ACME server name is "router-fdb531" and the local domain is "example.com", the certificate will at least contain two subjectAltName extensions with values "DNS:router-fdb531.example.com" and "DNS:router-fdb531.local".

3.4.2. Accounts

ACME account objects contain an array of contact strings. Normally this array consists of "mailto:" URIs, however for local IoT devices an array of "https:" URIs should be used instead, one for each fully-qualified domain name used by the device.

3.4.3. IoT Device Certificate Signing Requests

The certificate signing request supplied by the IoT Device MUST use a SHA2 hash of at least 256 bits and MUST use either RSA encryption with a key length of at least 3072 bits or ECDSA encryption with the secp384r1 (P-384) or secp521r1 (P-521) curves.

The commonName MUST be the device ID.

The request MUST also contain subjectAltName extensions for the mDNS (".local") and any local domain name(s), MAY contain subjectAltName extensions for the current IP address(es) of the device, and MUST NOT contain subjectAltName extensions for "localhost". For example, if the device name is "device-12cd56" and the local domain is "example.com", the signing request will at least contain two subjectAltName extensions with values "DNS:device-12cd56.example.com" and "DNS:device-12cd56.local".

3.4.4. Manual Approval of Signing Requests

The local ACME server MUST provide interfaces to manually approve initial IoT Device certificate signing requests and revoke approvals of IoT Devices in order to protect against rogue devices obtaining trusted certificates. Once approved, the server can automatically approve renewals unless such approval has been revoked.

When the local ACME server resides on a consumer Internet modem/router, such approvals will typically be authorized using the router's administrative account on the router's embedded web interface and/or mobile application.

3.4.5. IoT Device Certificates

Certificates generated by the local ACME server MUST have an expiration of three months or less.

3.5. Client Device Configuration

Client Devices, upon connecting to a network, MUST use ACME Server Discovery to determine whether the local network has an ACME Server. If it does, the Client Device connects to the server using HTTPS and copies the X.509 certificates for use in validating future connections to IoT Devices. The Client Device SHOULD utilize a TOFU validation policy for self-signed X.509 certificates unless otherwise configured, for example in a managed enterprise network environment. Client Devices can present UI informing and/or obtaining consent from the user to use or trust the root certificate, however such UI is beyond the scope of this document.

The Client Device MUST NOT use the supplied X.509 certificate when validating certificates on other networks. The certificate is typically associated with the network interface name, network SSID, and/or MAC address of the default router and MAY be associated with the local domain name. Client Devices MUST validate the host name(s) and/or IP address(es) to validate the CA certificate against the name(s) or IP address(es) supplied by the DHCP or DNS server during discovery. Since a certificate MAY be used for multiple networks, for example with a wireless cable modem that provides both Wi-Fi and Ethernet connectivity, such validation MUST allow for the presence of subjectAltName extensions containing values other than those provided by the DHCP or DNS server the Client uses.

3.6. IoT Device Configuration

IoT Devices, upon connecting to a network, MUST use ACME Server Discovery to determine whether the local network has an ACME Server. If it does, the IoT Device connects to the server using HTTPS and uses the ACME protocol to obtain, renew, or verify an X.509 certificate for each network the device is connected to. The IoT Device SHOULD utilize a TOFU validation policy for self-signed X.509 certificates unless otherwise configured, for example in a managed enterprise network environment.

The IoT Device MAY share/reuse certificates between networks when those networks utilize the same ACME server and X.509 certificate.

4. Security Considerations

The security considerations of IoT provisioning are similar to those described in [RFC1034], [RFC2131], [RFC6763], [RFC8446], and [RFC8555]. The following subsections describe additional security considerations.

4.1. Trusting Local DHCP and DNS Infrastructure

This specification necessarily depends on all devices trusting the underlying network infrastructure, specifically the local DHCP and DNS servers. Public Networks cannot be trusted and MUST NOT use a local ACME server or its root certificate. Private Networks can be trusted more and SHOULD use a local ACME server and its root certificate. Sites can utilize Network Endpoint Assessment [RFC5209] and Trusted Network Connect (TNC) [RFC5792][RFC5793] to provide enhanced security and trust in the local network.

4.2. Certificate Signing Request Validation

The local ACME Server MUST validate the subjectAltName values in certificate signing requests from IoT Devices. DNS name suffixes MUST be restricted to ".local" and the configured local domain name(s), and the leftmost label MUST NOT be the name of the local ACME Server or "localhost". IP addresses MUST be limited to link-local, loopback, and private use addresses.

4.3. Man-in-the-Middle Attacks

Because the local ACME Server will often rely on a self-signed certificate and TOFU validation policy, a man-in-the-middle attack is possible with successful DHCP, DNS, and/or mDNS request interception and/or redirection. Such attacks can be detected using network monitoring tools, and the use of a long-lived root certificate helps to mitigate the possibility that compromised network connections or infrastructure will go undetected by the Client Device.

4.4. Storage of Key Material

It is important for all devices to protect stored encryption keys from disclosure. Disclosure of the local ACME Server's private key will compromise all encrypted traffic on the local network. Disclosure of an IoT Device's private key will only affect that device's traffic.

4.5. Revocation and Reissuance/Regeneration

All devices MUST provide a way for an End User to re-issue X.509 certificates and regenerate a new private/public key pair for certificates and certificate requests. The most common way is through a so-called "factory reset" process that restores a device to its original, factory configuration/state.

All devices SHOULD provide a way for an End User to revoke X.509 certificates.

4.6. Use of mDNS

Multicast DNS (mDNS) [RFC6762] has a number of known security limitations. DHCP Option NNN provides the local ACME Server's fully-qualified domain name which can be resolved using mDNS, providing a small window for a man-in-the-middle attack during initial device connection. Such attacks can be detected using network monitoring tools and/or through the use of a root X.509 certificate from a trusted, public CA on the local ACME Server.

4.7. mDNS Domain Name Collisions

Multicast DNS (mDNS) domain names ("example.local.") can collide with other network devices. While mDNS does define an algorithm to resolve name collisions, IoT Devices SHOULD use a default name with a unique identifier, e.g., "device-12cd56.local.", so that name changes are less likely. When an IoT Device's mDNS changes, it MUST revoke all certificates for the old name with the (current) local ACME Server and request new certificate(s) for the new name. Portable IoT Devices that connect to many different networks MUST track their mDNS hostname separately for each network and only revoke certificates for the currently connected network(s).

4.8. Network Identification and Validation

Client and IoT Devices SHOULD identify networks using the local network interface name, MAC address of the default router, and/or the Wi-Fi SSID and validate the local ACME Server's root certificate when connecting. Wi-Fi validation is necessarily limited since Wi-Fi SSIDs are not unique. Client Devices MUST and IoT Devices SHOULD notify the End User when the root certificate changes for a network.

4.9. Multiple Network Support

Multiple network configurations pose an interesting implementation challenge. The most typical multiple-network configurations are Wi-Fi + cellular and Wi-Fi + Ethernet. Cellular networks are Public Networks and MUST NOT be used for local ACME servers or root certificates. Ethernet networks are usually Private Networks whose mDNS namespace might or might not be bridged with the Wi-Fi network.

Client Devices MUST separately track and validate the root X.509 certificate for each local ACME Server. Similarly, IoT Devices MUST separately track, store, and use X.509 certificates for each local ACME Server. Client and IoT Devices MAY purge "old" network information if sufficient storage space is not available.

4.10. Protection of Certificates and Key Material

IoT Devices and local ACME Servers MUST protect access to certificate and key material, allowing access only to approved software running on the device or server. The private key for an X.509 certificate MUST NOT be accessible outside of the corresponding device or server.

4.11. Reuse of Key Material

IoT Devices MUST NOT reuse key material when generating an X.509 certificate signing request. Local ACME Servers MUST NOT reuse key material when generating the root X.509 certificate.

5. IANA Considerations

5.1. DHCP Option

In accordance with [RFC2132], IANA has added the following new DHCP option to the BOOTP Vendor Extensions and DHCP Options [DHCP-REGISTRY] registry:

Tag: NNN

Name: ACME Server

Data Length: N (variable length)

Meaning: Fully-qualified domain name of the local ACME server

Reference: This document

5.2. Service Name

In accordance with [RFC6335], IANA has added the following new service name to the Service Name and Transport Protocol Port Number Registry [SERVICE-REGISTRY]:

Service Name: acme-server

Port Number: None

Transport Protocol: tcp

Description: Automatic Certificate Management Environment (ACME) server

Assignee: Michael Sweet

Contact: Michael Sweet

Reference: This document

Assignment Notes: Defined TXT keys: None

6. Normative References

- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, DOI 10.17487/RFC1034, November 1987, <<https://www.rfc-editor.org/info/rfc1034>>.
- [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>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <<https://www.rfc-editor.org/info/rfc2132>>.
- [RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <<https://www.rfc-editor.org/info/rfc3748>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC5209] Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, "Network Endpoint Assessment (NEA): Overview and Requirements", RFC 5209, DOI 10.17487/RFC5209, June 2008, <<https://www.rfc-editor.org/info/rfc5209>>.
- [RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, DOI 10.17487/RFC5216, March 2008, <<https://www.rfc-editor.org/info/rfc5216>>.
- [RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, DOI 10.17487/RFC5247, August 2008, <<https://www.rfc-editor.org/info/rfc5247>>.

- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <<https://www.rfc-editor.org/info/rfc5280>>.
- [RFC5792] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute (PA) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5792, DOI 10.17487/RFC5792, March 2010, <<https://www.rfc-editor.org/info/rfc5792>>.
- [RFC5793] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker (PB) Protocol Compatible with Trusted Network Connect (TNC)", RFC 5793, DOI 10.17487/RFC5793, March 2010, <<https://www.rfc-editor.org/info/rfc5793>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<https://www.rfc-editor.org/info/rfc6763>>.
- [RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <<https://www.rfc-editor.org/info/rfc6335>>.
- [RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <<https://www.rfc-editor.org/info/rfc7435>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8555] Barnes, R., Hoffman-Andrews, J., McCarney, D., and J. Kasten, "Automatic Certificate Management Environment (ACME)", RFC 8555, DOI 10.17487/RFC8555, March 2019, <<https://www.rfc-editor.org/info/rfc8555>>.
- [RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/info/rfc9110>>.

7. Informative References

[DHCP-REGISTRY]

IANA, "BOOTP Vendor Extensions and DHCP Options",
<[https://www.iana.org/assignments/bootp-dhcp-parameters/
bootp-dhcp-parameters.xhtml#options](https://www.iana.org/assignments/bootp-dhcp-parameters/bootp-dhcp-parameters.xhtml#options)>.

[RFC3925] Littlefield, J., "Vendor-Identifying Vendor Options for
Dynamic Host Configuration Protocol version 4 (DHCPv4)",
RFC 3925, DOI 10.17487/RFC3925, October 2004,
<<https://www.rfc-editor.org/info/rfc3925>>.

[SERVICE-REGISTRY]

IANA, "Service Name and Transport Protocol Port Number
Registry", <[https://www.iana.org/assignments/service-
names-port-numbers/service-names-port-numbers.xhtml](https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml)>.

Appendix A. Change History

[RFC Editor: This section to be deleted before RFC publication]

February 7, 2025 - draft-sweet-acme-iot-07

- * Added discussion of a default host name with a unique identifier.
- * Added discussion of network connection security.
- * Added discussion of manual approval of certificate requests.
- * Added 802.1X device ID to CSRs/certificates.
- * Added EAP RFCs

August 9, 2024 - draft-sweet-acme-iot-06

- * Added discussion of network types - public, private, "domain".
- * Add security considerations - do not enable local trust root on public networks.

January 30, 2024 - draft-sweet-acme-iot-05

- * Clarified the contents and validation of X.509 root (CA) certificates.
- * Added note that a Client Device can display UI when connecting to a network with an ACME server.

- * Clarified key material security considerations.
- * Clarified that devices may purge old network information.
- * Updated X.509 certificate revocation requirement to SHOULD.
- * Clarified that X.509 certificates are only revoked on the current network and the mDNS hostname is tracked separately per network.

August 2, 2023 - draft-sweet-acme-iot-04

- * Updated introduction to provide a clearer explanation of the scope and purpose of the document.
- * Added security considerations for protecting the certificate and key materials to address concerns of malicious software running on the device.
- * Added security considerations for reusing key materials to address concerns that manufacturers might hardcode keys.

February 6, 2023 - draft-sweet-acme-iot-03

- * Added security considerations for trusting the local network infrastructure with references to NEA and TNC.

July 14, 2022 - draft-sweet-acme-iot-02

- * Added clarifications and more detail per Printer Working Group review at May 2022 face-to-face meeting, specifically more detail in the introduction and security considerations for mDNS Domain Name Collisions, Network Identification and Validation, and Multiple Network Support.

April 14, 2022 - draft-sweet-acme-iot-01

- * Added temporary use of DHCP vendor class option (60), per guidance from DHCP WG chair

April 6, 2022 - draft-sweet-acme-iot-00

- * Initial revision.

Author's Address

Michael Sweet (editor)
Lakeside Robotics Corporation
1094 Valecrest St
Blezard Valley Ontario P0M 1E0
Canada
Email: msweet@msweet.org