



DRAFT

**BSR E1.88**

Entertainment Technology - Framework for Entertainment Network Cybersecurity  
& Efficiency (FENCE)

Approved by ANSI's Board of Standards Review on \_\_\_\_\_

CP/2026-1010r0

© 202x Entertainment Services and Technology Association  
All rights reserved.

---

**Notice and Disclaimer**

ESTA does not approve, inspect, or certify any installations, procedures, equipment or materials for compliance with codes, recommended practices or standards. Compliance with an ESTA standard or recommended practice is the sole and exclusive responsibility of the manufacturer or provider and is entirely within their control and discretion. Any markings, identification or other claims of compliance do not constitute certification or approval of any type or nature whatsoever by ESTA.

ESTA neither guaranties nor warrants the accuracy or completeness of any information published herein and disclaim liability for any personal injury, property or other damage or injury of any nature whatsoever, whether special, indirect, consequential or compensatory, directly or indirectly resulting from the publication, use of, or reliance on this document.

In issuing and distributing this document, ESTA does not either (a) undertake to render professional or other services for or on behalf of any person or entity, or (b) undertake any duty to any person or entity with respect to this document or its contents. Anyone using this document should rely on their own independent judgment or, as appropriate, seek the advice of a competent professional in determining the exercise of reasonable care in any given circumstance.

**Published By:**

Entertainment Services and Technology Association (ESTA)  
271 Cadman Plaza PO Box 23200  
Brooklyn, NY 11202-3200  
USA  
Phone: +1-212-244-1505  
Email: [standards@esta.org](mailto:standards@esta.org)

## **ESTA's Technical Standards Program**

**ESTA's Technical Standards Program** was created to serve the ESTA membership and the entertainment industry in technical standards-related matters. The goal of the program is to take a leading role regarding technology and safety within the entertainment industry by creating recommended practices and standards, monitoring standards issues around the world on behalf of our members, and improving communications and safety within the industry. In its technical standards development activities, ESTA works closely with other industry organizations, including ESA, CITT, USITT and VPLT, in addition to representing the interests of ESTA members to ANSI, UL, ASCE, ICC, and NFPA. ESTA is an ANSI Accredited Standards Developer.

**The Technical Standards Council (TSC)** established by ESTA's Board of Directors to oversee and coordinate the Technical Standards Program. Made up of individuals experienced in standards development work from throughout our industry, the Committee approves all projects undertaken and assigns them to the appropriate working group. The Technical Standards Council employs a Technical Standards Manager to coordinate the work of the Committee and its working groups as well as maintaining a "Standards Watch" on behalf of members. Working groups include: Control Protocols, Electrical Power, Event Safety, Floors, Fog and Smoke, Followspot Positions, Mental Health & Well-being Management, Photometrics, Rigging, Stage Machinery, and Prop Weapons Safety.

**ESTA encourages active participation in the Technical Standards Program.** There are several ways to become involved. The easiest way to actively participate is to respond to any of the public reviews advertised on ESTA's [public review web page](#). The next level of participation requires completion of an application to become a working group member; applications are available from the TSP's [procedural documents web page](#). Application as an Observer with non-voting status affords access to updates on standards development documents. Application as a voting participant affords full access to a consensus voice that helps shape the industry. Voting status carries responsibilities of responding to letter ballots and attending meetings, but membership in ESTA or any other organization is not a requirement for participation in the TSP. One can also become involved by requesting that the TSC develop a standard or a recommended practice in an area of concern to them.

**The Control Protocols Working Group**, which authored this standard, consists of a cross section of entertainment industry professionals representing a diversity of interests. ESTA is committed to developing consensus-based standards and recommended practices in an open setting.

**Investors in Innovation**

The Technical Standard Program (TSP) is financially supported by ESTA and by companies and individuals who make donations to the TSP. Contributing companies and individuals who have helped fund the TSP are recognized as "Investors in Innovation". The Investors in Innovation when this standard was approved by ANSI's Board of Standards Review are gratefully acknowledged as follows:

*[Insert the current Investors table here]*

**Memorial donor:** The Estate of Ken Vannice

All donations to the Technical Standards Program benefit the entire program, and are not directed to any specific use or project within the program. Please help support the Technical Standards Program by becoming an Investor in Innovation. Visit our website at <http://tsp.esta.org/invest>, or contact the ESTA office at 1-212-244-1505 and select "TSP" from the menu.

**Contact Information****Technical Standards Manager**

Richard J. Nix  
ESTA  
271 Cadman Plaza PO Box 23200  
New York, NY 11202-3200  
USA  
+1-212-244-1505  
[richard.nix@esta.org](mailto:richard.nix@esta.org)

**Technical Standards Council Co-chairpersons**

Alan Rowe  
I.A.T.S.E Local 728  
+1-310-702-2909  
[amrowe@iatse728.org](mailto:amrowe@iatse728.org)

Dan Culhane  
Wenger Corp  
+1-612-868-4769  
[culhane.dan@gmail.com](mailto:culhane.dan@gmail.com)

**Control Protocols Working Group Co-chairpersons**

Javid Butler  
Goddard Design LLC  
+1-702-759-2427  
[javid@goddard.design](mailto:javid@goddard.design)

Maya Nigrosh  
[mnigrosh@alumni.cmu.edu](mailto:mnigrosh@alumni.cmu.edu)

**Acknowledgments**

The Control Protocols Working Group members, when this document was approved by the working group on DD MMMM YYYY, are gratefully acknowledged below.

**Voting members:**

**Observer (non-voting) members:**

**Interest category codes:**

CP = custom-market producer DE = designer  
DR = dealer rental company G = general interest  
MP = mass-market producer U = user

Table of Contents

**Notice and Disclaimer.....i**

**ESTA’s Technical Standards Program.....ii**

**Investors in Innovation.....iii**

**Contact Information.....iv**

**Acknowledgments.....v**

**List of Tables.....x**

**List of Figures.....x**

**1 Introduction.....1**

    1.1 Scope.....1

    1.2 Regulatory Context.....1

    1.3 Compliance.....1

**2 Applicability of other Standards and References.....1**

    2.1 Normative References.....1

    2.2 Informative References.....3

**3 Definitions.....3**

**4 System Architecture & Interfaces.....3**

    4.1 Interfaces.....3

    4.2 Introduction to Security Concepts.....3

        4.2.1 Authentication.....3

        4.2.2 Authorization.....3

        4.2.3 Confidentiality.....4

        4.2.4 Integrity.....4

        4.2.5 Freshness.....4

        4.2.6 Provisioning.....4

        4.2.7 Device Attestation.....4

        4.2.8 Time Synchronization.....4

        4.2.9 Protected Resources.....4

        4.2.10 Operational Modes.....4

    4.3 Summary of Protocols (Informative).....5

        4.3.1 CBOR and COSE.....5

        4.3.2 EDHOC.....5

        4.3.3 OSCORE.....5

4.3.4 ACE-OAuth.....	5
4.3.5 Group OSCORE.....	5
4.3.6 CoAP.....	5
<b>4.4 Architecture.....</b>	<b>6</b>
4.4.1 Planes.....	6
4.4.2 Transport Classes.....	6
4.4.3 Control Plane.....	6
4.4.4 Data Plane.....	7
4.4.5 OSCORE and Group OSCORE (Informative).....	7
<b>4.5 Trust Models.....</b>	<b>7</b>
4.5.1 Manufacturing Attestation Trust.....	7
4.5.2 Operational Trust.....	8
<b>5 Network-Layer Onboarding (informative).....</b>	<b>8</b>
5.1 Air-Gaps and Network Isolation.....	8
5.2 Network Access Control.....	8
<b>6 Identity Management and Device Attestation.....</b>	<b>8</b>
6.1 Trust Hierarchy Overview.....	9
<b>6.2 Attestation.....</b>	<b>9</b>
6.2.1 Factory Provisioning Requirements.....	9
6.2.2 Provisioning Flowchart.....	11
6.2.3 Factory Certificate Usage Constraints.....	11
6.2.4 Factory Certificate Revocation.....	12
<b>6.3 Out-of-Band Credential Format.....</b>	<b>12</b>
<b>7 Application Layer Onboarding.....</b>	<b>12</b>
7.1 Discovery via mDNS.....	15
7.2 Device Announce and Guardian Matching.....	15
7.3 Secure Channel Establishment and Verification.....	15
7.4 Identity Claim.....	16
7.5 Ownership Proof.....	16
7.6 Entity ID.....	16
7.7 Onboarding API Resources.....	16
7.8 Device Reset and Recovery.....	16
7.9 EDHOC Denial of Service Protection.....	17
7.10 Onboarding Race Conditions.....	17

<b>8 Device Authorization.....</b>	<b>17</b>
<b>8.1 Authorization Model.....</b>	<b>17</b>
8.1.1 Permission Concepts.....	17
8.1.2 Credential architecture: Access Token and Authorization Assertion.....	17
<b>8.2 CoAP URI-Path.....</b>	<b>19</b>
8.2.1 URI Structure and Namespace.....	19
8.2.2 Resource Collections and Scoping.....	20
8.2.3 Informative Example.....	20
8.2.4 Security classification.....	20
<b>8.3 Token request.....</b>	<b>21</b>
<b>8.4 Session binding.....</b>	<b>21</b>
<b>8.5 Scope validation (requested vs assigned).....</b>	<b>21</b>
<b>8.6 Credential issuance.....</b>	<b>21</b>
8.6.1 Access Token (CWT).....	21
8.6.2 Access Scope encoding (CBOR).....	23
8.6.3 Authorization Assertion (AA).....	24
<b>8.7 Runtime enforcement on the data plane.....</b>	<b>24</b>
8.7.1 Controller Egress Enforcement.....	24
8.7.2 Responder Ingress Enforcement.....	24
8.7.3 Error Handling and Diagnostics.....	25
<b>8.8 AA Distribution, Retrieval and Caching.....</b>	<b>25</b>
8.8.1 Caching requirements.....	25
8.8.2 Peer-to-peer AA retrieval resource.....	25
8.8.3 Behavior on missing AA.....	26
8.8.4 Pre-emptive AA & Pairwise Setup.....	26
<b>9 Transport Layer.....</b>	<b>27</b>
<b>9.1 General.....</b>	<b>27</b>
<b>9.2 Key Derivation.....</b>	<b>27</b>
<b>9.3 Security Processing.....</b>	<b>27</b>
<b>9.4 Packet Format.....</b>	<b>28</b>
9.4.1 CoAP and Group OSCORE Relationship (Informative).....	29
9.4.2 Data Plane Packet Structure.....	29
9.4.3 Outer CoAP Header.....	29
9.4.4 OSCORE Option and Encrypted Payload.....	30
<b>10 Key Management, Revocation and Online Enhancements.....</b>	<b>30</b>
<b>10.1 Key Classes.....</b>	<b>30</b>
<b>10.2 Guardian Trust Model.....</b>	<b>31</b>

<b>10.3 Protocol Key Rotation</b> .....	<b>31</b>
<b>10.4 Device Revocation</b> .....	<b>32</b>
10.4.1 Revocation and Certificate Lifetimes.....	32
10.4.2 Revocation Procedure.....	32
10.4.3 Responder Behavior During Security Rekey.....	33
10.4.4 Online Enhanced Mode.....	33
<b>10.5 Connectivity and Trust Freshness</b> .....	<b>33</b>
<b>11 Guardian Implementation Requirements</b> .....	<b>34</b>
<b>11.1 Attestation Trust Store Management</b> .....	<b>34</b>
<b>11.2 Registry Storage and Security</b> .....	<b>34</b>
<b>11.3 Denial of Service Protection</b> .....	<b>34</b>
<b>11.4 High Availability and Recovery</b> .....	<b>34</b>
11.4.1 High Availability.....	35
11.4.2 Disaster Recovery.....	35
<b>11.5 Guardian Discovery</b> .....	<b>35</b>
<b>12 User Authentication and Authorization</b> .....	<b>35</b>
<b>12.1 Administrator</b> .....	<b>35</b>
<b>12.2 Operator</b> .....	<b>35</b>
<b>13 Security Considerations</b> .....	<b>35</b>
<b>13.1 Onboarding Attack Vectors</b> .....	<b>35</b>
<b>13.2 Unattested Identity</b> .....	<b>36</b>
<b>13.3 On-Path Attacker (Passive and Active)</b> .....	<b>37</b>
<b>13.4 Guardian Compromise</b> .....	<b>37</b>
<b>13.5 Stale Authorization Information</b> .....	<b>37</b>
<b>13.6 Time Manipulation Attacks</b> .....	<b>37</b>
<b>13.7 Compromised Device / Insider Threat</b> .....	<b>38</b>
<b>13.8 Firmware Integrity</b> .....	<b>38</b>
<b>14 Implementation Guidelines (non-normative)</b> .....	<b>38</b>
<b>14.1 Design Principles</b> .....	<b>38</b>
<b>14.2 Key Generation</b> .....	<b>39</b>
<b>15 Glossary (informative / non-normative)</b> .....	<b>39</b>
<b>Appendix A: Numeric Recommendations (Informative)</b> .....	<b>43</b>

**Appendix B: Compliance Profiles (Normative).....44**

**B.1 Profile: FENCE-2026-BASE.....44**

**B.2 Ownership Proof Structure.....44**

### List of Tables

No table of figures entries found.

### List of Figures

Figure 1 FENCE Architecture.....8

Figure 2 Factory Provisioning Workflow..... 13

Figure 3 Onboarding Sequence..... 16

Figure 4 Authorization Credential Architecture.....23

Figure 5 Data Plane Communication Flow.....34

Figure 6 Data Plane Group OSCORE Packet Format.....35

## 1 Introduction

### 1.1 Scope

This standard describes the Framework for Entertainment Network Cybersecurity Enhancements (FENCE), an application-layer security protocol designed for entertainment control networks. It specifies a secure communication system for the transmission of real-time control data, providing cryptographic protection as an alternative to legacy, unencrypted transport methods.

The primary objective of this standard is to transition entertainment control systems from "Open by Default" to "Secure by Design" without compromising the low-latency requirements of live performance.

The protocol adopts a Zero Trust architecture, ensuring that security is bound to the data rather than relying on network boundaries. It is designed to be fully functional in ad-hoc, offline environments and does not require Internet connectivity or an external network time server (NTP).

The protocol enforces an internal time synchronization where a centralized network entity known as a Guardian acts as the authoritative time source.

FENCE is designed to operate fully in air-gapped environments. When external network connectivity is available, additional automated security capabilities may be enabled by a Guardian, including trust store updates and revocation processing. Such connectivity shall not be required for normal protocol operation.

### 1.2 Regulatory Context

This standard is designed to assist manufacturers and implementers in adhering to emerging cybersecurity regulations, such as the EU Cyber Resilience Act (CRA), California SB-327, and Oregon HB-2395.

### 1.3 Compliance

Compliance with this standard is strictly voluntary and the responsibility of the implementor. Markings and identification or other claims of compliance do not constitute certification or approval by the E1 accredited standards committee.

## 2 Applicability of other Standards and References

### 2.1 Normative References

[sACN] ANSI E1.31-2016 Entertainment Technology - Lightweight streaming protocol for transport of DMX512 using ACN

This standard is maintained by ESTA.

ESTA  
P.O. Box 23200  
Brooklyn, NY 11202-3200  
+1-212-244-1505  
<http://tsp.esta.org>

ESTA is a standardization body accredited by ANSI to develop, maintain and withdraw American National Standards.

ANSI  
25 West 43rd Street  
4th floor  
New York, NY 10036  
+1-212-642-4900

<http://www.ansi.org>

[E1.88-1] ANSI E1.88-20xx Entertainment Technology – In development

This standard is maintained by ESTA.

[ASCII] ISO/IEC 646 Information Technology – ISO 7-bit Coded Character Set for information interchange. 1991

This standard is maintained by ISO.

ISO  
International Organization for Standardization  
1, Rue de Varembe  
Case Postale 56  
CH-1211 Geneva 20  
Switzerland  
+41 22 74 901 11  
[www.iso.ch](http://www.iso.ch)

[UDP] RFC 0768 UDP User Datagram Protocol

This standard is maintained by the IETF.

IETF Administration LLC  
1000 N West Street, Suite 1200  
Wilmington, DE 19801  
USA  
+1 (302) 319-3421+1 (302) 319-3421+1 (302) 319-3421+1 (302) 319-3421  
[www.ietf.org/](http://www.ietf.org/)

[CoAP] RFC 7252 The Constrained Application Protocol (CoAP)

This standard is maintained by the IETF.

[OSCORE] RFC 8613 Object Security for Constrained RESTful Environments (OSCORE)

This standard is maintained by the IETF.

[CBOR] RFC 8949 Concise Binary Object Representation (CBOR)

This standard is maintained by the IETF.

[COSE] RFC 9052 CBOR Object Signing and Encryption (COSE): Structures and Process

This standard is maintained by the IETF.

[ACE-OAuth] RFC 9200 Authentication and Authorization for Constrained Environments Using the OAuth 2.0 Framework (ACE-OAuth)

This standard is maintained by the IETF.

[ACE-OSCORE] RFC 9203 The ACE-OAuth Framework for CoAP and OSCORE

This standard is maintained by the IETF.

[EDHOC] RFC 9528 Ephemeral Diffie-Hellman Over COSE (EDHOC)

This standard is maintained by the IETF.

[GROUP OSCORE] IETF draft-ietf-core-oscore-groupcomm. Group Object Security for Constrained RESTful Environments (Group OSCORE).

This standard is maintained by the IETF.

[ACE-GROUP-OSCORE] RFC 9594 Key Provisioning for Group Communication using Authentication and Authorization for Constrained Environments (ACE).

This standard is maintained by the IETF.

## 2.2 Informative References

[DMX] ANSI E1.11-2008 Entertainment Technology – USITT DMX512-A Asynchronous Serial Digital Data Transmission Standard for controlling lighting equipment and accessories.

This standard is maintained by ESTA.

[RDM] ANSI E1.20-2010 Entertainment Technology – Remote Device Management over DMX512 networks.

This standard is maintained by ESTA.

## 3 Definitions

**3.1 Byte & Octet:** Octet is an eight-bit byte. Octet and byte are used interchangeably in this document.

**3.2 Byte Ordering:** All multi-byte data shall be transmitted in network byte order (Big-Endian).

## 4 System Architecture & Interfaces

### 4.1 Interfaces

FENCE defines three device types.

A Guardian is the entity that controls authentication and authorization.

A Controller is a device that once authenticated and authorized, generates data.

A Responder is a device that once authenticated and authorized, consumes data.

Controllers and Responders are both referred to as devices.

### 4.2 Introduction to Security Concepts

#### 4.2.1 Authentication

Authentication is the process of verifying the identity of a device. This is used to ensure that the device connecting to the network is legitimate and not an imposter. FENCE achieves this using [EDHOC].

#### 4.2.2 Authorization

Authorization is the process of determining what an authenticated device is allowed to do. This includes permissions such as whether a device can transmit or receive data. A device may be authenticated (allowed to communicate with other FENCE devices) but not authorized to control specific attributes. FENCE achieves this using [ACE-OAuth]

### 4.2.3 Confidentiality

Confidentiality is the protection of data to prevent unauthorized disclosure. It is achieved by encrypting the data payload. This ensures that a bad actor who captures network traffic cannot easily read or interpret it. FENCE achieves this using [COSE] and [OSCORE].

All OSCORE and Group OSCORE operations shall use the Profile AEAD Algorithm specified in Appendix B.

### 4.2.4 Integrity

Integrity is the guarantee that data has not been altered in transit. This ensures, for example, that a lighting level sent as "50%" has not been changed (maliciously or otherwise) to "100%" by the time it reaches the Responder. FENCE achieves this using the OSCORE protocol.

### 4.2.5 Freshness

Freshness is the guarantee that a message is current and not a recording of a previous valid command (known as a replay attack). This, for example, prevents a bad actor from recording a Blackout command and replaying it during a subsequent performance. FENCE achieves this using the OSCORE protocol (via Sequence Numbers and partial IV).

### 4.2.6 Provisioning

Factory Provisioning is the mandatory initial setup phase occurring during the manufacturing of a product. During this phase, the device is permanently programmed with its unique cryptographic identity (Identity Key Pair and X.509 Certificate) and its onboarding credentials (Onboarding Key Pair).

### 4.2.7 Device Attestation

Device Attestation is the process by which a Guardian verifies the provenance of a device's hardware identity. While authentication verifies a device's active network presence, attestation ensures that the device is a genuine product from a trusted manufacturer and not a counterfeit or unauthorized clone.

FENCE achieves this during the onboarding process by validating the device's factory-installed X.509 Identity Certificate against the Guardian's supply chain root of trust.

### 4.2.8 Time Synchronization

To enforce credential expiry in an offline network, FENCE relies on "Guardian Time." A Guardian is the authoritative time source for the FENCE network. All devices shall synchronize their internal clocks to a Guardian's time upon successful authentication. Devices lacking a battery-backed Real Time Clock (RTC) shall treat this synchronized time as a monotonic baseline.

A Guardian may or may not be synchronized with a time server over the Internet.

### 4.2.9 Protected Resources

The term Protected is used throughout this document to describe any data, message, or resource that is secured using FENCE protocols. A protected resource is one that is encapsulated within an OSCORE or Group OSCORE message.

### 4.2.10 Operational Modes

FENCE supports two operational modes:

- Offline Mode, in which a Guardian operates without external connectivity and relies solely on locally configured and cached trust information.

- Online Mode, in which a Guardian has external connectivity and may automatically retrieve updated trust and revocation information.

Operational mode selection shall not affect protocol message formats or device behavior on the data plane.

Devices shall process protected control data only when valid authentication, authorization, and time synchronization are established.

### 4.3 Summary of Protocols (Informative)

#### 4.3.1 CBOR and COSE

CBOR (Concise Binary Object Representation) provides the binary encoding for all data. COSE (CBOR-Object Signing and Encryption) uses CBOR to structure the cryptographic elements, such as keys and signatures.

#### 4.3.2 EDHOC

EDHOC (Ephemeral Diffie-Hellman Over COSE) is the authenticated key exchange protocol. When a device connects to a Guardian, it uses EDHOC to verify identities and establish shared secrets.

It supports mutual authentication, verifying the Guardian's identity against the FENCE Trust Root and the Device's identity against its Manufacturer Certificate.

#### 4.3.3 OSCORE

OSCORE (Object Security for Constrained RESTful Environments) is used to protect the transport of data for all unicast traffic between the device and Guardian.

#### 4.3.4 ACE-OAuth

ACE (Authentication and Authorization for Constrained Environments) is the framework used to transfer Authorization information. It operates over the secure OSCORE channel to issue Access Tokens that define the device's Scope and permissions.

#### 4.3.5 Group OSCORE

Group OSCORE (Secure Group Communication for CoAP) is a mechanism used for runtime operations.

FENCE uses both of the Group OSCORE transports:

- Pairwise Mode allows devices to securely derive keys for efficient unicast communication based on their assigned Scope.
- Group Mode uses multicast and is used for one-to-many applications, for example sync packets and timecode.

#### 4.3.6 CoAP

CoAP (Constrained Application Protocol) is the data transfer protocol used by FENCE. It operates over UDP and provides two distinct delivery modes:

- Confirmable (CON) messages, which guarantee delivery via retries (used for the Control Plane).
- Non-Confirmable (NON) messages, which prioritize low latency (used for the Data Plane).

When used with OSCORE, CoAP serves a dual role:

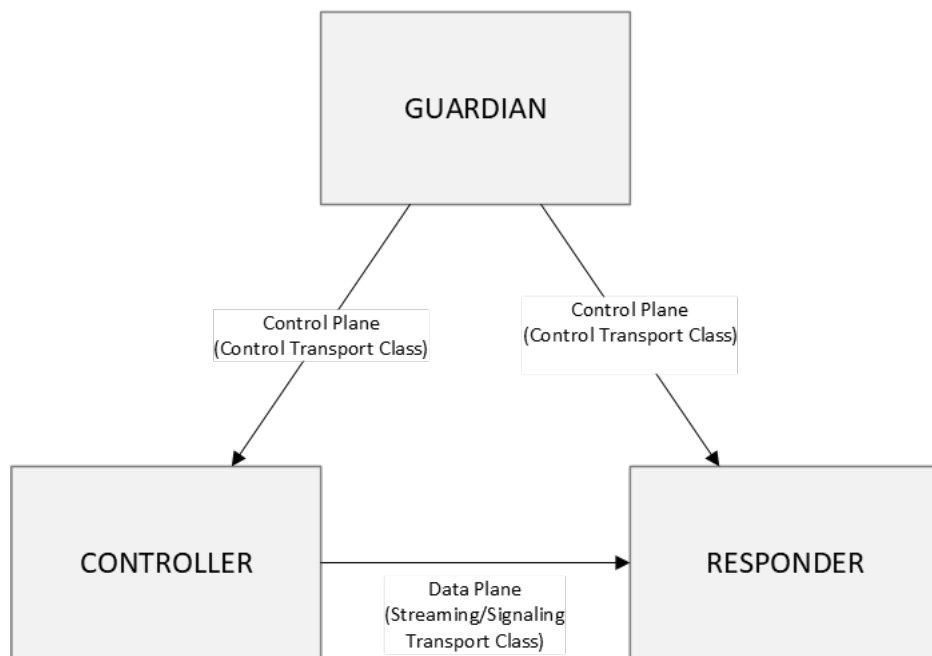
- Outer CoAP: The visible envelope that encapsulates the encrypted data and handles routing across the network.
- Inner CoAP: The encrypted payload of the OSCORE message which contains the application command or data.

## 4.4 Architecture

### 4.4.1 Planes

The FENCE architecture is divided into two logical planes to separate administrative operations from real-time performance: the Control Plane and the Data Plane.

Figure 1 FENCE Architecture



### 4.4.2 Transport Classes

In order to ensure extensibility and consistent security application, FENCE abstracts its network communications into the following formally defined transport classes:

#### Control Transport Class

- Definition: Authenticated [OSCORE] over [CoAP]/[UDP] using Confirmable (CON) messages.
- Mandate: All FENCE Control Plane communications shall utilize the Control Transport Class.

#### Signaling Transport Class

- Definition: [Group OSCORE] over [CoAP]/[UDP] using Confirmable (CON) messages.
- Mandate: All FENCE Data Plane communications requiring application-layer acknowledgement (e.g., Peer-to-Peer Authorization, AA Retrieval) shall utilize the Signaling Transport Class.

#### Streaming Transport Class

- Definition: [Group OSCORE] over [CoAP]/[UDP] using Non-Confirmable (NON) messages.
- Mandate: All FENCE Data Plane communications requiring low latency (e.g., Real-time Lighting Data) shall utilize the Streaming Transport Class.

### 4.4.3 Control Plane

The control plane is the administrative layer used for communication between a Guardian and devices, where operational trust is established and public identities are exchanged. On this plane, the Guardian issues both a private Access Token (containing the device's assigned OSCORE input material) and a shareable Authorization Assertion (containing the device's authorization scope and Identity Public Key). Together, these credentials

provide the cryptographic and authorization foundations required for data plane operations. All communication on the control plane shall use the Control Transport Class.

#### 4.4.4 Data Plane

The data plane is the runtime layer where control data is streamed directly between peer devices (Controllers and Responders). All communication on the data plane utilizing UDP shall use the Streaming Transport Class or the Signaling Transport Class.

#### 4.4.5 OSCORE and Group OSCORE (Informative)

FENCE implements both OSCORE and Group OSCORE. While they share common cryptographic primitives (based on [COSE]), they are utilized on separate planes to distinguish administrative operations from runtime performance:

- [OSCORE] secures the Control Plane data using ephemeral keys negotiated directly via [EDHOC], in accordance with the Control Transport Class. Its primary purpose is to safely bootstrap operational trust and request credentials from a Guardian.
- [Group OSCORE] secures the Data Plane data using the Master Secret distributed inside the Access Token, in accordance with the Streaming or Signaling Transport Class.

FENCE uses two distinct modes of [Group OSCORE] to support its diverse traffic requirements:

- Pairwise Mode allows any two authorized devices within a Security Group to derive a unique, shared key for direct communication. This derivation is performed autonomously at runtime using the Master Secret previously distributed by the Guardian.
- Group Mode uses multicast. It allows a device to encrypt a message once using the group key such that all members of the group can decrypt it. This is essential for low-latency one-to-many applications, such as timecode and synchronization packets.

By decoupling these functions, FENCE ensures that the high-latency negotiation required for the control plane never impacts the low-latency requirements of the data plane.

### 4.5 Trust Models

FENCE employs two distinct and complementary trust hierarchies to manage security across a device's lifecycle: Manufacturing Attestation and Operational Trust. Both hierarchies are rooted in Certificate Authorities (CAs), and the trust anchors for these CAs shall be represented as standard X.509 certificates. These hierarchies are intentionally decoupled to support different security goals.

#### 4.5.1 Manufacturing Attestation Trust

Purpose: To verify a device's authenticity and provenance during initial onboarding.

Mechanism: This hierarchy is used by a Guardian. It is rooted in one or more Manufacturer CAs. A Guardian's trust store is configured by the system operator with the X.509 certificates of these trusted Manufacturer CAs. This collection of trusted root and intermediate certificates is referred to as the Attestation Trust Store. The Guardian uses these certificates to validate the signature on the device's X.509 Identity Certificate presented during the Identity Claim phase (Section 7.4). This validation assesses whether the device is a genuine product from a trusted source according to the Guardian's configured policy.

Guardian Policy for Attestation Outcome: If the Guardian's validation of a device's X.509 Identity Certificate against its Attestation Trust Store succeeds, the device's provenance is cryptographically verified, and onboarding proceeds without further administrative intervention regarding provenance.

If validation fails (e.g., the certificate does not exist, is self-signed, expired, revoked, or signed by an untrusted CA), the Guardian shall flag the device as having an un-attested identity. The Guardian's administrative interface shall then require explicit administrator approval to proceed with onboarding the device. This provides the flexibility for a system administrator to accept devices that lack cryptographic supply-chain attestation.

Scope: The trust established by this hierarchy is used only for the one-time act of device onboarding and is not used for runtime operational security.

#### **4.5.2 Operational Trust**

Purpose: To establish authority and secure communication on a live network.

Mechanism: This hierarchy is used by a Device. It is rooted in a local CA known as the "FENCE Trust Root," which defines a unique security domain. A device acquires and pins the FENCE Trust Root's X.509 certificate during the initial onboarding process (Section 10.2).

Scope: The trust established by this hierarchy governs all control plane communications, including pairing, authorization, and key management.

### **5 Network-Layer Onboarding (informative)**

The FENCE protocol operates at the application layer. It assumes that the device has already successfully connected to the physical network, obtained an IP address (via DHCP or Static assignment), and can route UDP packets to a Guardian.

While the specific methods used to secure the physical link are outside the scope of the FENCE normative requirements, the security of the overall system can be significantly enhanced by following standard network best practices.

#### **5.1 Air-Gaps and Network Isolation**

Historically, the industry has relied on physical network isolation ("air-gapping") as the primary mechanism for securing control systems. Whilst physical isolation remains a valid strategy to reduce the external attack surface, it is increasingly difficult to maintain in modern environments that demand remote management, cloud connectivity, and widespread integration.

Furthermore, reliance solely on perimeter defense leaves a network vulnerable to internal threats or accidental breaches once the physical boundary is crossed.

By shifting trust from the physical network perimeter to cryptographic authentication and authorization at the application layer, FENCE ensures robust security whether the system is physically isolated or connected to the Internet.

#### **5.2 Network Access Control**

To prevent unauthorized hardware from being admitted onto the network, the use of port-based authentication is a best practice.

The preferred mechanism is IEEE 802.1x. A properly configured 802.1x system places any newly connected device into a restricted network segment. Only after the device's identity has been successfully authenticated is the switch port moved to the trusted network where FENCE devices operate. This approach mitigates threats from both casually-used empty ports and malicious connection attempts.

Because 802.1x is a logical control, it should be complemented by physical security measures. Steps should be taken to prevent unauthorized access to network infrastructure, which protects against attacks such as unplugging a trusted device to connect a malicious one in its place.

### **6 Identity Management and Device Attestation**

## 6.1 Trust Hierarchy Overview

FENCE defines layered trust hierarchies separating industry provenance, device identity, and operational authorization:

- A set of trusted Manufacturer Certificate Authorities used for the Attestation Trust Store.
- A FENCE Trust Root used to establish operational trust within a specific security domain.
- Guardian issued operational credentials used for runtime security.

These layers are evaluated at different phases of the device lifecycle and are intentionally decoupled to support both offline and connected operation.

## 6.2 Attestation

A device shall possess a unique cryptographic identity before participating in a FENCE network. This standard defines a single, mandatory factory provisioning model for all devices. This identity is established at the point of manufacture and is rooted in a supply chain trust model.

To support a secure, operator-initiated onboarding workflow, this identity is composed of two distinct asymmetric key pairs (P-256): an Onboarding Key Pair used to prove physical proximity, and a permanent Identity Key Pair used to prove cryptographic ownership. This section defines the normative requirements for these elements and how they are attested to a Guardian. w

### 6.2.1 Factory Provisioning Requirements

1. All devices shall be provisioned at the point of manufacture with a unique cryptographic identity conforming to the FENCE-2026-BASE compliance profile defined in Appendix B. This provisioning shall include the following four elements:
2. A unique Asset ID. This shall be a 128-bit random integer generated using a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). This identifier is static and shall not change for the life of the device.
3. An Onboarding Key Pair. This is an asymmetric key pair (<onboarding\_keypair>) used exclusively for bootstrapping the initial secure connection. The private key shall be stored in secure hardware and must not be exportable.
4. An Identity Key Pair. This is a second, distinct asymmetric key pair (<identity\_keypair>) that represents the device's permanent, long-term identity. The private key shall be stored in secure hardware and must not be exportable.
5. An X.509 Device Identity Certificate. This certificate shall contain the public key from the Identity Key Pair and shall be signed by a Manufacturer or Batch key. This certificate serves as the device's permanent, verifiable identity credential.

Informative Note on Dual Key Pair Architecture:

The use of two distinct key pairs (Onboarding and Identity) is a deliberate security design choice.

- The Onboarding Key serves as a temporary "proof of proximity." Its public half is discoverable during the initial handshake, proving that the Guardian is talking to the physical device that was scanned.
- The Identity Key represents the device's permanent, trusted identity. Its public half is only revealed after the secure channel is established, and is used to prove cryptographic ownership of that identity.

This separation ensures that the device's long-term identity is never exposed during the initial, unauthenticated discovery phase.

Informative Note on IEEE 802.1AR Alignment:

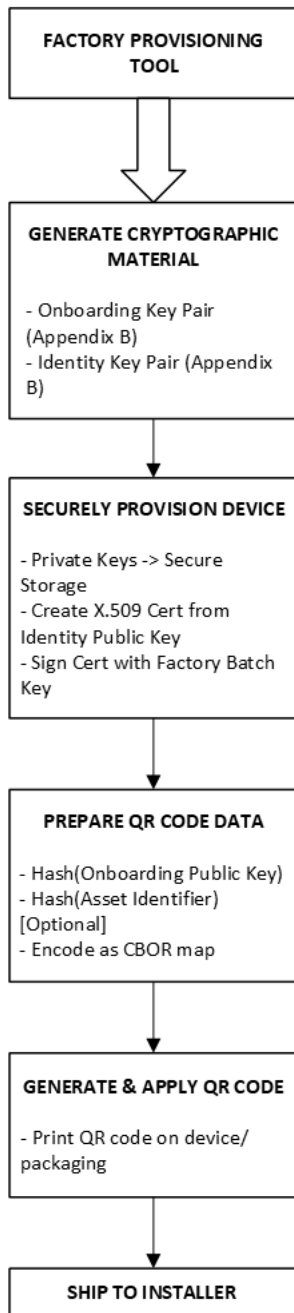
The Identity Key Pair and its associated X.509 Device Identity Certificate function effectively as a Secure Device Identifier (DevID) as defined in IEEE 802.1AR. Specifically, the permanent, factory-provisioned Identity Certificate serves as the Initial Device Identifier (IDevID). It provides a long-lived, cryptographically verifiable proof of the device's hardware identity and manufacturing provenance.

In contrast, the Access Token issued by the Guardian (Section 8.6.1) serves a function analogous to a Locally Significant Device Identifier (LDevID). It represents the device's operational identity within the specific local trust domain of the venue.

FENCE adopts this layered identity model to ensure that a device's permanent manufacturer identity (IDevID) remains distinct from its local network privileges (LDevID), facilitating secure lifecycle management and transfer of ownership.

### 6.2.2 Provisioning Flowchart

Figure 2 Factory Provisioning Workflow



### 6.2.3 Factory Certificate Usage Constraints

Factory attestation certificates shall be used solely for device provenance verification during the initial FENCE onboarding. To maintain a clear separation between long-term identity and short-term operational credentials, the following constraints shall apply within the FENCE protocol:

Factory certificates shall not be used for:

- EDHOC authentication after registration;
- OSCORE or Group OSCORE key derivation;

- Authorization decisions; or
- Runtime data protection.

Upon successful registration, a Guardian registry becomes the authoritative source of trust for the device.

### 6.2.4 Factory Certificate Revocation

A Guardian shall manage a trust store of Manufacturer CAs and shall support revocation information for each. Revocation shall prevent future registration of devices whose attestation chains to a revoked certificate.

### 6.3 Out-of-Band Credential Format

To support operator-initiated onboarding and mitigate privacy risks associated with tracking, devices shall utilize an Out-of-Band (OOB) Credential. This credential is physically associated with the device (e.g., printed on a label, embedded as an RFID tag) and contains specific, privacy-preserving cryptographic information required for device discovery and matching during onboarding.

The OOB credential shall contain a CBOR map with the following keys:

1. Onboarding Key Hash: A cryptographic hash of the device's public Onboarding Key, calculated using the Profile Hash Algorithm specified in Appendix B. This is used by a Guardian to uniquely identify the device during discovery.
2. Asset ID Hash (Optional): A cryptographic hash of the device's Asset Identifier (Section 6.2.1). This allows for inventory and asset tracking without revealing the device manufacturer or network address.

The CBOR encoding keys shall be:

```
oob-credential-map = {
  1 => bstr, ; Onboarding Key Hash (Profile Hash Algorithm)
  ? 2 => bstr ; Asset ID Hash (Profile Hash Algorithm)
}
```

Implementations shall support at least one of the following normative OOB credential types:

- QR Code: The oob-credential-map shall be encoded as a QR code printed on the device or its packaging.
- RFID Tag: The oob-credential-map shall be stored in an RFID tag embedded within or physically attached to the device, readable by compatible Guardian hardware.

## 7 Application Layer Onboarding

Application Layer Onboarding is the unified, operator-initiated process by which a new device becomes a fully operational and trusted member of a FENCE network. The process begins with the device discovering a Guardian, is authorized by an operator who provides an Out-of-Band (OOB) Credential from the device (e.g., by scanning a QR code or an RFID tag), and culminates in the device receiving its crucial Access Token credential.

This Access Token, detailed in Section 8, contains both authorization permissions (from the Guardian's Authorization Server role) and the cryptographic material needed to join the data plane (from its Group Manager role). Therefore, the entire onboarding sequence described in this section is the necessary prerequisite for obtaining this token and participating in runtime operations.

The onboarding journey is a single, consistent workflow for all FENCE devices, detailed in the following sections. It consists of a sequence of cryptographic verifications:

- Guardian Discovery (Section 7.1): An un-onboarded device powers on and uses DNS-Service Discovery (DNS-SD) to find the IP address of one or more potential Guardians on the network.
- Device Announce and Operator Action (Section 7.2): The device announces its presence to a discovered Guardian. In parallel, an operator scans the device's QR code, which informs the Guardian to expect that

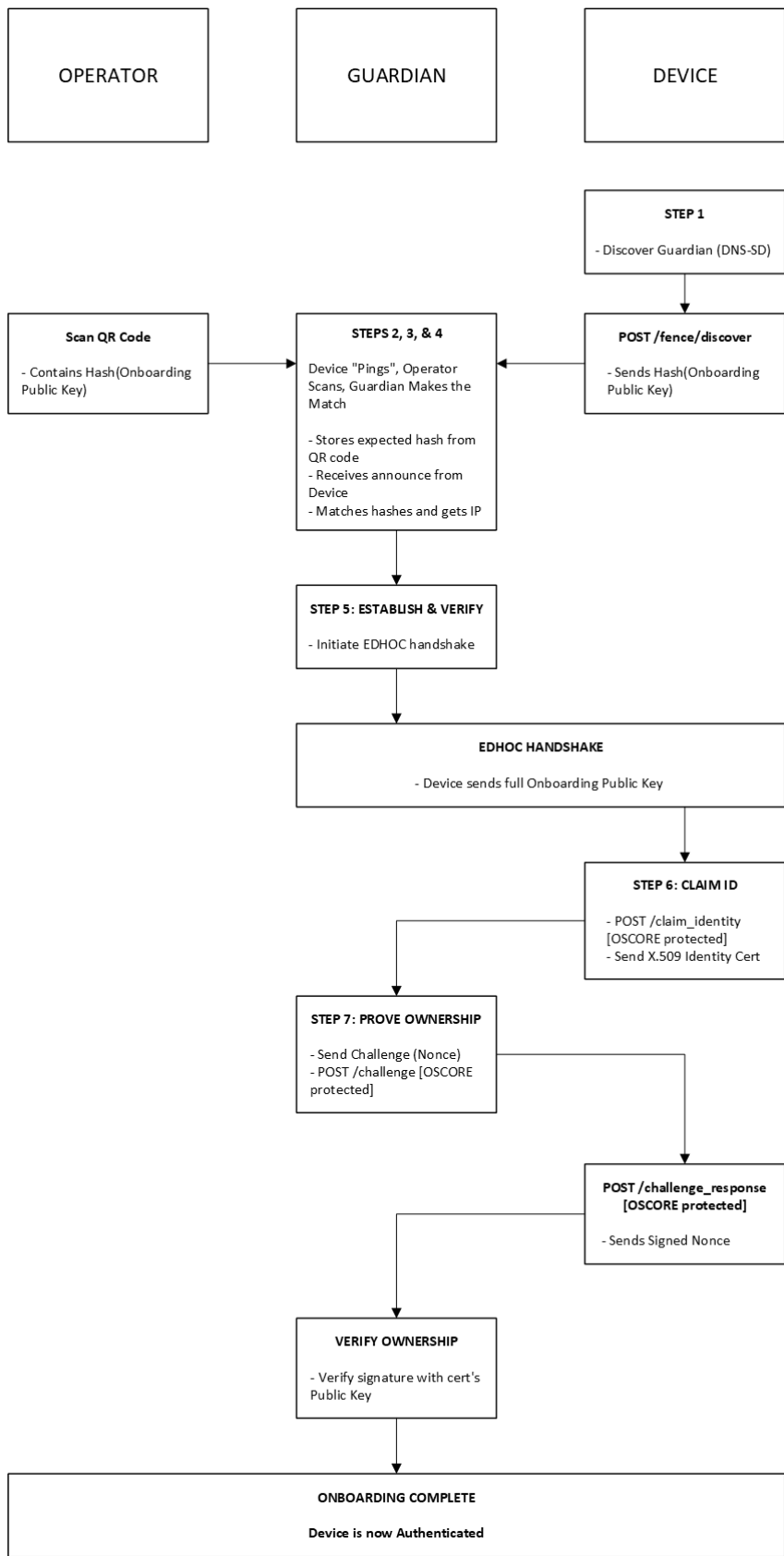
specific device. The Guardian matches the device's announcement with the operator's scan, linking the network session to the physical device.

- **Secure Channel Establishment & Verification (Section 7.3):** The Guardian initiates a secure handshake with the device. A critical step in this phase cryptographically proves the Guardian is communicating with the physical device that was scanned (Proof of Proximity).
- **Identity Claim & Ownership Proof (Sections 7.4 & 7.5):** Over the established secure channel, the device presents its permanent identity and proves it possesses the corresponding private key, confirming its authenticity (Proof of Ownership).

#### Operational Workflow (Informative):

This unified process is designed for efficiency and security. A new device automatically finds the network's authority. An operator with physical access to the device simply scans its QR code to authorize a fully automated cryptographic handshake. This workflow securely validates the device's authenticity and brings it online without requiring manual entry of passwords, tokens, or network addresses, while being robust across both simple and complex network topologies.

Figure 3 Onboarding Sequence



## 7.1 Discovery via mDNS

To support both simple and complex network topologies, discovery is initiated by the device using DNS-Service Discovery [RFC 6763].

An un-onboarded device shall, after successfully obtaining an IP address, perform a DNS-SD query for the FENCE Guardian service. The standard service name for this query shall be `_fence-guardian._udp`.

The device shall perform this query using both:

- Multicast DNS [mDNS] for discovery on the local network segment.
- Conventional Unicast DNS, if a DNS server was provided via DHCP.

A Guardian shall advertise its FENCE onboarding service using the `_fence-guardian._udp` service name. Guardian implementations should support being advertised via both an mDNS responder and a static entry in a conventional DNS server.

Upon receiving one or more IP addresses for a Guardian from the DNS-SD query, the device shall proceed to the Announce step described in Section 7.2.

## 7.2 Device Announce and Guardian Matching

After discovering one or more potential Guardian IP addresses, the device shall announce its presence.

For each Guardian IP address discovered, the device shall send a Confirmable (CON) [CoAP] POST request to the resource URI-Path `/esta/e1.88/<version>/discover`. The payload of this request shall be a [CBOR] map containing a cryptographic hash of its Onboarding Public Key

The [CBOR] encoding key for the hash shall be 1.

`discover-payload = { 1 => bstr } ; Onboarding Key Hash (Profile Hash Algorithm)`

A device shall retry this announcement using randomized exponential backoff, starting with a minimum delay of `<announce_backoff_interval>` and increasing. The device shall continue these periodic announcements, regardless of whether it receives a [CoAP] Acknowledgement (ACK) from the Guardian, until it receives an [EDHOC] handshake initiation from a Guardian.

In parallel, the operator provides the device's Out-of-Band (OOB) Credential (e.g., by scanning a QR code or RFID tag). The Guardian application reads the hash(`OnboardingPublicKey`) from this OOB credential and shall store this hash in a temporary list of expected devices for onboarding.

A Guardian shall listen for incoming requests on the unprotected resource `/esta/e1.88/<version>/discover`. Upon receiving a request, the Guardian shall compare the hash in the payload against its list of expected devices. If a match is found, the Guardian has linked the operator's physical action with the device's network presence and shall proceed to the next step using the source IP address of the request.

The Guardian shall acknowledge Confirmable discovery messages in accordance with [CoAP]. However, to prevent state exhaustion, a Guardian may silently discard the payload of a discovery request if the hash does not match a currently pending Out-of-Band credential.

## 7.3 Secure Channel Establishment and Verification

Upon matching the hash of a device's announced public key with an expected hash from a scanned QR code, the Guardian shall initiate an EDHOC handshake with the source IP address of the announcement. The handshake shall negotiate a cipher suite compliant with the requirements of the Profile EDHOC Suite defined in Appendix B.

During the handshake, the device shall transmit its full, unhashed Onboarding Public Key.

The Guardian shall immediately perform a verification step: it shall calculate a hash of the public key it just received and compare it to the hash it obtained from the Out-of-Band (OOB) Credential.

If the hashes match, the Guardian has cryptographically proven it is communicating with the correct physical device. It shall proceed to complete the EDHOC handshake to establish a secure, end-to-end encrypted channel. If the hashes do not match, the Guardian shall abort the connection.

#### 7.4 Identity Claim

Over the newly established secure channel, the device shall send its permanent, factory-installed X.509 Identity Certificate to the Guardian. The Guardian shall then attempt to validate this certificate against its Manufacturing Attestation Trust Store as defined in Section 4.5.1.

#### 7.5 Ownership Proof

To complete the onboarding, the Guardian shall challenge the device to prove it possesses the private key corresponding to the Identity Certificate.

The Guardian shall generate a cryptographically secure random nonce and send it to the device. The device shall construct the OwnershipProofChallenge structure as defined in Appendix B.2, populating its fields with the received nonce and the specified challenge\_type. The device shall then deterministically CBOR encode this structure and sign the resulting byte string using the private key from its Identity Key Pair, returning the signature.

The Guardian shall verify the signature using the public key from the X.509 certificate and by independently constructing and encoding the expected OwnershipProofChallenge structure for verification. If the signature is valid, ownership is proven, and the device is considered successfully onboarded and authenticated.

#### 7.6 Entity ID

The entity\_id is a logical identifier (e.g., a user-friendly name) assigned by the operator via the Guardian's interface. It serves as a handle for the operator to manage the device's permissions and group assignments. It is not required for protocol operations, as the device is identified cryptographically by its Identity Public Key.

#### 7.7 Onboarding API Resources

**This section is under construction. A full API will be defined in next version.**

The Identity Claim and Ownership Proof steps shall be performed using CoAP POST requests to protected resources on the Guardian. The following URI paths are defined for this purpose:

- Identity Claim: POST /esta/e1.88/<version>/claim\_identity
  - The OSCORE-protected payload of this request shall contain the device's X.509 Identity Certificate.
- Ownership Proof Challenge: POST /esta/e1.88/<version>/challenge
  - The OSCORE-protected payload of this request shall contain the Guardian's randomly generated nonce.
- Ownership Proof Response: POST /esta/e1.88/<version>/challenge\_response
  - The OSCORE-protected payload of this request shall contain the device's signature over the received nonce.

#### 7.8 Device Reset and Recovery

To mitigate "First-Use Hostage" attacks where a device associates with a malicious or inaccessible Guardian identity, all devices shall support a local mechanism to clear stored Guardian credentials and return the device to the UNAUTHENTICATED state.

Devices shall implement a secure mechanism to return the device to an UNAUTHENTICATED state.

Upon reset, the device shall delete any stored `guardian_endpoint`, `master_secret`, and `sender_id`, and revert to the UNAUTHENTICATED state (Section 7.1).

## 7.9 EDHOC Denial of Service Protection

A Guardian shall implement EDHOC Cookie mechanisms (RFC 9528, Section 6) to protect against handshake state exhaustion attacks.

A Guardian should implement rate limiting to restrict the number of EDHOC handshake attempts per source IP address. Recommended: no more than 10 handshake initiations per IP address per minute.

Devices should implement exponential backoff when EDHOC handshakes fail, with a minimum delay of `<edhoc_backoff_interval>` between retry attempts.

## 7.10 Onboarding Race Conditions

Two operators could scan the same device and initiate onboarding via two different Guardians simultaneously. To avoid this potential race condition, a device shall process the first EDHOC handshake request it receives and ignore all subsequent mDNS responses or EDHOC handshake initiations until the current attempt is completed or times out.

# 8 Device Authorization

## 8.1 Authorization Model

Once a device is onboarded and has completed the EDHOC pairing handshake (Section 7.2 – 7.5), it possesses a secure communication channel to a Guardian. However, it does not yet have the cryptographic material or permissions to operate on the data plane.

This section defines the ACE-OAuth token issuance process, in which a device requests the credentials required to join a Security Group and participate in runtime data exchange. The Guardian, acting as both a Group Manager (GM) and an Authorization Server (AS), is responsible for issuing these credentials.

### 8.1.1 Permission Concepts

FENCE permissions comprise the following components:

- **Security Group:** A set of devices that share common cryptographic material (a Master Secret) and operate within a single Group OSCORE context. A Security Group typically represents a logical or geographic boundary such as “Main Theatre” or “Foyer”.
- **Scope:** The permissions defining which resources (e.g. `/.../univ/...`) a device may READ (receive/consume) and/or WRITE (transmit/control). In FENCE, scope is represented as CBOR and supports large sets of universes expressed as inclusive ranges.

Informative Note on Terminology:

Readers familiar with ANSI E1.33 (RDMnet) should note a difference in terminology. In FENCE, the concept of a logical network boundary, which E1.33 calls a “Scope,” is functionally implemented by a Security Group.

The term Scope is used in this document consistent with the underlying IETF ACE-OAuth framework, where it strictly refers to a set of granted permissions (e.g., “access rights”).

### 8.1.2 Credential architecture: Access Token and Authorization Assertion

To manage runtime security, a Guardian issues two distinct but complementary cryptographic artifacts to each authenticated device: a private Access Token and a shareable Authorization Assertion.

- The Access Token is a confidential credential, encoded as a CBOR Web Token (CWT), intended exclusively for the device to which it is issued. It contains sensitive cryptographic material (in the cnf claim) necessary for the device to derive its Group OSCORE keys and join its assigned Security Group. It also contains the device's granted permissions (in the scope claim), which the device uses to enforce its own operational boundaries. This token is delivered over the secure OSCORE control plane and shall never be shared with other devices.

Informative Note on the FENCE Access Token:

Readers familiar with web-based OAuth 2.0 protocols should note a critical distinction in how the FENCE Access Token is used. In typical bearer token models, an Access Token is a credential presented to third-party Resource Servers and must not contain secret keying material.

In contrast, the FENCE Access Token is not a bearer token. It is a confidential package containing both authorization scope and Group OSCORE keying material, delivered from the Guardian to a device over a pre-established, end-to-end encrypted OSCORE channel. It is for the device's internal use only, serving to bootstrap its data plane security context. The role of a shareable, verifiable credential is fulfilled by the separate, non-secret Authorization Assertion (AA).

- The Authorization Assertion (AA) is a Guardian-signed, non-secret object that is safe to share with peers. It binds a device's Sender ID within a Security Group to its explicit scope (permissions). The AA's purpose is to allow other devices to verify a sender's permissions without having access to any secret information.

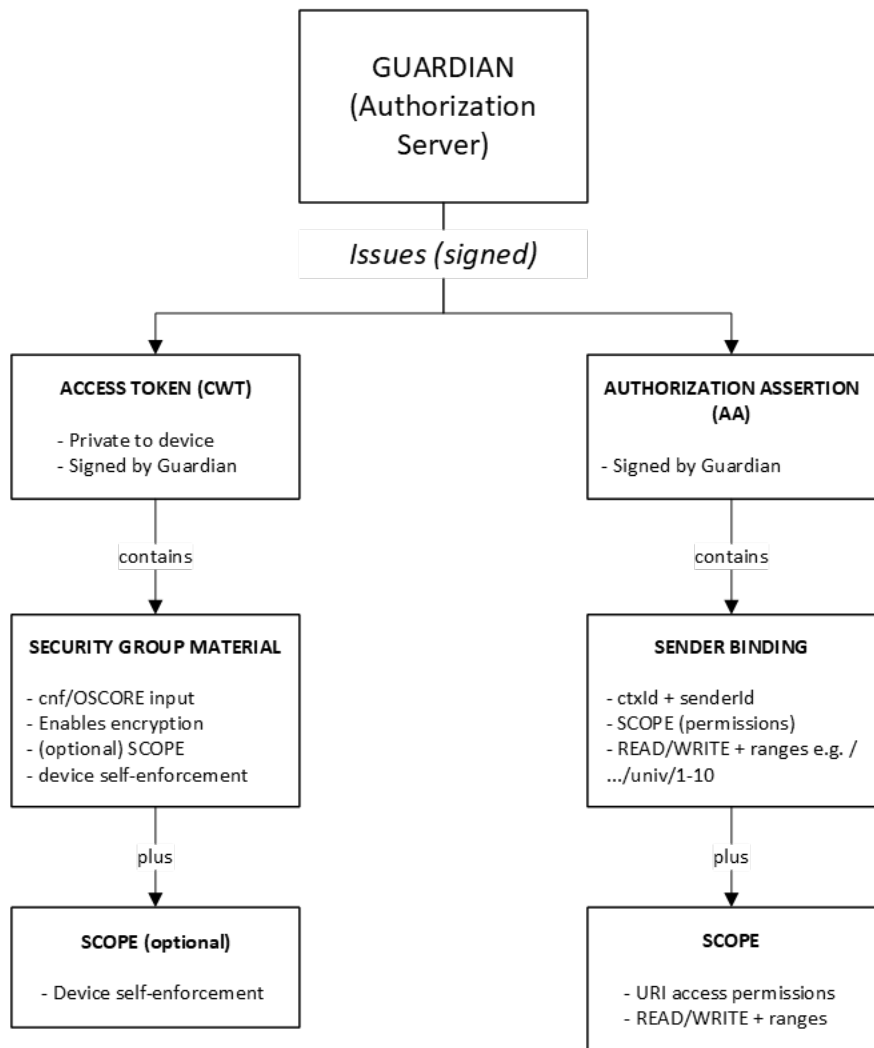
These two credentials fulfill distinct roles in the data plane:

- Internal Configuration (via Access Token): The device uses its private Access Token to configure its own security context. It derives the necessary keys to encrypt and decrypt data plane messages and understands its own permissions (e.g., a Responder uses the scope from its token to know it is authorized to process data for Universe 1).
- Peer Verification (via Authorization Assertion): Devices use the AAs of their peers to make authorization decisions. A Responder caches a Controller's AA to verify, on a per-message basis, that the Controller is permitted to send data to a specific destination resource.

Example:

- A Console (Controller) receives an Access Token granting it the keys and permission to WRITE to Universes 1–10. It also receives an AA publicly stating these permissions.
- A Moving Light (Responder) receives an Access Token granting it the keys and permission to READ Universe 1.
- When the Light receives an encrypted message for Universe 1 from the Console's Sender ID, it first decrypts the message using keys derived from its own Access Token. It then validates the Console's cached AA to confirm the Console is authorized to WRITE to Universe 1 before processing the payload.

Figure 4 Authorization Credential Architecture



Runtime use:

Access Token: device derives Group OSCORE context (keys) and enforces its own limits.

AA : peers verify sender is authorized for the destination (cached by Responders).

## 8.2 CoAP URI-Path

FENCE uses a hierarchical resource directory defined by the CoAP Uri-Path option. Uri-Path is a Class E (Encrypted) option in OSCORE; therefore, it is carried inside the encrypted Inner CoAP message. An on-path observer cannot read or tamper with the path without detection.

This section defines the normative structure that all FENCE URI-Paths shall follow. The specific resource names, identifiers, and payload formats are defined in companion standards within the E1.88-x series.

### 8.2.1 URI Structure and Namespace

All FENCE URI-Paths shall begin with the following prefix:

`/esta/e1.88/<version>`

Where:

- `esta`: A literal string that identifies the ESTA standard namespace.
- `E1.88`: A literal string that identifies the FENCE namespace.
- `<version>`: The version of this document. The current pre-release document is "v0".

The remainder of the URI-Path following this prefix is defined by this standard or a referencing standard.

### 8.2.2 Resource Collections and Scoping

To enable fine-grained authorization, the FENCE authorization model (Section 8.6) operates on "resource collections." If a companion standard in the E1.88-x series defines resources that require instance-based access control (e.g., by universe, zone, or channel), the URI-Path for those resources shall follow this template:

```
.../<collection_name>/<instance_id>/...
```

Where:

- `<collection_name>`: A short ASCII string that identifies the type of resource collection (e.g., "univ"). This name is used as the resource family identifier within the scope claim for authorization.
- `<instance_id>`: A positive integer in decimal representation used to uniquely identify a specific resource within the collection. Leading zeroes shall not be used. This is the identifier that FENCE authorization policies (Section 8.7.2) will check against the permitted ranges in a device's scope. Implementations shall reject values that are not a canonical base-10 integer representation.

### 8.2.3 Informative Example

A hypothetical companion standard, E1.88-10, defines a set of resources for controlling lighting universes. It might define a URI as follows:

```
/esta/e1.88/<version>/univ/1/slot
```

This URI conforms to the FENCE structure:

- `<collection_name>` is `univ`.
- `<instance_id>` is `1`.

In this example, a Guardian could issue a scope granting a device `WRITE` permission to the `univ` collection for the integer range `[1, 100]`. The FENCE Responder would then use this generic rule to enforce that the sender is authorized to write to this specific path. The semantic meaning of `univ` and `slot` are defined entirely within E1.88-10.

### 8.2.4 Security classification

FENCE resources are classified into the following security categories. For a detailed explanation of the roles of OSCORE and Group OSCORE, see Section 4.4.4.

- **Unprotected Resources (NoSec)**: These resources, typically used for initial discovery, are transmitted in cleartext [CoAP] messages and do not require prior authentication.
- **Protected Resources**: These resources contain sensitive control, metadata, or [RDM] information and shall be carried in cryptographically protected messages. Access to protected resources requires that a device has been successfully authenticated and authorized. Protected resources are further divided based on the security protocol used:
  - **Control Plane Resources**: Resources used for communication between a device and a Guardian (e.g., token requests, diagnostics). These shall be protected using the Control Transport Class, utilizing the OSCORE security context established during the EDHOC handshake.

- Data Plane Resources: Resources used for communication between peer devices (e.g., lighting data, authorization assertions). These shall be protected using either the Streaming Transport Class or the Signaling Transport Class, utilizing the Group OSCORE security context derived from the Access Token. Authorization to access these resources shall be enforced as specified in Sections 8.7.

### 8.3 Token request

Upon successful completion of the onboarding and ownership proof process (Section 7.5), the device shall request an Access Token by sending a POST /token to the Guardian over the established pairwise OSCORE channel.

The token request shall be protected using the OSCORE context derived from the EDHOC handshake, ensuring confidentiality and integrity of the request parameters.

Request Parameters:

The device shall include the following parameters in the POST /token request body (CBOR-encoded):

1. `grant_type`: Shall be set to "client\_credentials" per RFC 9200 Section 5.8.2.
2. `scope`: A CBOR byte string describing the requested permissions. The scope shall specify the universe ranges and access rights (READ/WRITE) the device is requesting. The format is defined in Section 8.6.2.
3. `c_i`: The EDHOC connection identifier (C\_I) from the completed EDHOC handshake. This parameter enables a Guardian to validate session binding (Section 8.4).

### 8.4 Session binding

A Guardian shall validate that the EDHOC session identifier (`c_i`) used for the /token request corresponds to a currently authenticated session. The Guardian shall use the cryptographic identity (the X.509 Identity Key) associated with that session to determine the device's assigned permissions in the Registry.

If the session is not valid, or if the authenticated identity does not map to a valid entry in the Guardian's registry, the Guardian shall reject the request.

### 8.5 Scope validation (requested vs assigned)

A Guardian shall compare the requested scope against the scope assigned to the Device ID in the registry.

If the requested scope is a subset of the assigned scope, the request may proceed.

If the requested scope exceeds the assigned scope, a Guardian shall return a 4.01 Unauthorized error.

To ensure consistent subset checking, the scope encoding defined in Section 8.6.2 shall be canonicalised (sorted and merged ranges).

### 8.6 Credential issuance

When a device is authorized to participate in a Security Group on the Data Plane, a Guardian shall issue:

- an Access Token, private to the device and
- an Authorization Assertion, safe to share with peers.

#### 8.6.1 Access Token (CWT)

For devices using Signalling Transport Class or Streaming Transport Class, a Guardian shall issue an Access Token encoded as a CBOR Web Token (CWT). This token contains the cryptographic material required to join the Security Group and contains at least the following claims:

- iat: issued at time
- exp: expiry time
- cnf: confirmation including Group OSCORE input material for the assigned Security Group and a Guardian-assigned Sender ID
- scope: the granted scope of permissions, encoded as defined in section 8.6.2

The CWT structure of the Access Token is defined below using CDDL.

```
access-token = {
  1: tstr      ; iss (Issued by)
  6: uint      ; iat (Issued at time)
  4: uint      ; exp (Expiry time)
  8: cnf-claim ; cnf (Confirmation)
  9: access-scope ; scope (Access scope - see section 8.6.2)
}
```

```
cnf-claim = {
  5: group-secret
}
```

```
group-secret = {
  0: bstr,      ; id (Sender ID)
  2: bstr,      ; ms (Group master secret)
  6: bstr      ; context_id (Group context ID)
}
```

The Access Token provides the following security resources:

- Cryptographic group membership (Group OSCORE keying material)
- Authorization policy (permissions scope)

Informative note: this token request combines the functionality of the Group Manager entity defined in [Group OSCORE] and the Authorization Server entity defined in [ACE-OAuth]. The cnf claim shall not be extracted; only the Authorization Assertion is distributed to peers.

The Access Token shall be signed by the Guardian using the Profile Guardian Signature Algorithm specified in Appendix B

Implementations shall protect the Access Token with the same security measures used for private keys.

Time Synchronization Requirement:

Upon receipt of the Access Token, the device shall accept the iat claim value as the current authoritative time and update its internal system clock (or software offset) to match.

Note: This mechanism provides devices without hardware RTCs with a valid time reference relative to a Guardian, enabling correct validation of exp claims.

The cnf claim shall convey OSCORE input material sufficient for the device to derive the Group OSCORE context for the assigned Security Group, including:

- the group secret input (e.g. Master Secret or equivalent)
- a context identifier; and
- the Sender ID assigned by a Guardian

The Sender ID shall be unique per device within the Security Group.

The Access Token contains sensitive keying material and shall not be distributed to other devices.

### 8.6.2 Access Scope encoding (CBOR)

Resource access authorization is encoded in the form of an access scope structure. This structure is included in both the Access Token and Authorization Assertion in the form of a CBOR map and defines access permissions per resource, using a pattern encoding that evaluates to one or more URI paths. The structure contains one or more scope sections, which are composed of the following components:

- An optional root URI pattern, which defines the root URI of all resources defined in this section's access rules.
- A set of access rules, which map an access level per device class to one or more URI paths. Different access privileges can be defined for resources owned by devices of different classes.

The structure of the CBOR map is defined below using Concise Data Description Language (CDDL).

```

access-scope = {
  0: [+ scope-section]
}

scope-section = {
  1: [* pattern-element],      ; Root URI of section
  2: [+ rule]                  ; Access rules for URI endpoints of
}

rule = {
  0: [+ pattern-element],      ; URI matching pattern
  1: access                    ; Access level
}

pattern-element = literal-pattern / numeric-range-pattern / any-pattern

literal-pattern = {
  0: 0,                        ; Pattern type = literal
  1: tstr                      ; URI segment value
}

numeric-range-pattern = {
  0: 1,                        ; Pattern type = numeric range
  1: uint,                    ; lower bound of range
  2: uint                      ; upper bound of range
}

any-pattern = {
  0: 2                          ; Pattern type = any
}

access = {
  ? 0: access-level,          ; Permission level for accessing resource on Responders
  ? 1: access-level,          ; Permission level for accessing resource on Controllers
  ? 2: access-level,          ; Permission level for accessing resource on any device
}

access-level = 0 / 1          ; 0 = read only, 1 = read/write

```

For sender policy enforcement, resource servers shall treat a client as authorized to access a resource if and only if the Authorization Assertion scope contains a rule that:

- contains a URI pattern that matches the URI of the target resource, and
- includes access permission level that matches the CoAP method used in the request

### 8.6.3 Authorization Assertion (AA)

A Guardian shall issue an Authorization Assertion, encoded as a COSE\_Sign1 object that includes a CBOR payload. The payload includes:

- The Security Group context identifier (binding the assertion to a specific cryptographic group)
- The Sender ID assigned to the device receiving permissions to access resources within that group
- The device's Identity Public Key
- The granted access scope rules
- Expiry time

The AA shall be integrity protected using a COSE signature such that any device possessing a Guardian trust anchor can verify it. The alg parameter in the protected header shall be the Profile Guardian Signature Algorithm specified in Appendix B. The AA shall not contain any secrets, including Group OSCORE input material.

The AA shall be a COSE\_Sign1 structure.

- The COSE\_Sign1 payload shall be the CBOR encoding of the AA Claims Map defined below.
- The COSE\_Sign1 protected header shall include alg.
- The COSE\_Sign1 unprotected header should include kid.

The AA payload shall be encoded using deterministic CBOR.

The internal structure of the Authorization Assertion is defined below, using CDDL.

```
authorization-assertion = {
  1: uint      ; ver (Version number - shall be 1)
  2: bstr      ; ctx (Group context ID)
  3: bstr      ; sid (Device Sender ID within group)
  4: access-scope ; scope (Access scope - see section 8.6.2)
  5: uint      ; exp (Expiry time)
  ? 6: uint    ; iat (Issued-at time, optional)
  ? 7: tstr    ; iss (Optional Guardian identifier)
  8: bstr      ; pubk (Device Identity Public Key)
}
```

## 8.7 Runtime enforcement on the data plane

To ensure zero-trust operation, enforcement shall be applied bi-directionally. Controllers shall perform egress filtering to prevent the transmission of unauthorized data, and Responders shall perform ingress filtering to reject unauthorized commands.

### 8.7.1 Controller Egress Enforcement

Prior to encrypting and transmitting a protected message, a Controller shall perform a local authorization check against its current Access Token.

- **Scope Validation:** The Controller shall verify that the target URI-Path of the message is explicitly permitted by a rule within its granted Access Scope (Section 8.6.2) with a permission level of WRITE.
- **Token Validity:** The Controller shall verify that its Access Token has not expired (exp claim).

If either check fails, the Controller shall not generate the message. This prevents the transmission of encrypted "garbage" that would waste bandwidth and processing power on the network.

### 8.7.2 Responder Ingress Enforcement

Upon receipt of a protected message, a Responder shall enforce security in the following strict order of operations. Failure at any step shall cause the message to be discarded immediately without further processing.

- **Replay Protection:** The Responder shall validate the OSCORE Partial IV (Sequence Number) against the sliding window for that Sender ID to prevent replay attacks.
- **Cryptographic Integrity:** The Responder shall perform Group OSCORE decryption and verification. If the MAC fails, the message is invalid.
- **Resource Parsing and Validation:** The Responder shall parse the Inner CoAP URI-Path. If the URI is malformed the message is invalid.
- **Sender Authorization (AA Check):** The Responder shall retrieve the cached Authorization Assertion (AA) associated with the kid (Sender ID) of the message.
  - If no valid AA exists, the message is unauthorized (see Section 8.8.3).
  - If an AA exists, the Responder shall match the Inner URI-Path of the decrypted message against the Access Scope defined in the AA. The message is authorized if and only if a rule exists granting WRITE access to that specific URI.

### 8.7.3 Error Handling and Diagnostics

To prevent Denial of Service (DoS) amplification attacks and side-channel leakage, Responders shall not generate network error messages (e.g., CoAP 4.01 or 4.03) in response to failures in the checks defined in 8.7.2. The message shall be silently dropped.

However, to facilitate troubleshooting, Responders shall increment an internal diagnostic counter for each specific failure type (e.g., auth\_scope\_violations, replay\_failures). These counters shall be retrievable by an authorized user via the diagnostic interface defined in Section 9.3.

## 8.8 AA Distribution, Retrieval and Caching

### 8.8.1 Caching requirements

Devices shall cache verified AAs indexed by (ctx, sid) until expiry.

A cached AA is valid if:

- a Guardian signature verifies against a configured Guardian trust anchor; and
- the current time (synchronized via the device's Access Token) is strictly less than exp; and
- the AA ctx matches the Security Group context in which the sender's messages are being processed.

If a device's internal time is reset (e.g., due to power loss), it shall be considered "Unauthorized" and must perform a new Token Request (Section 8.3) to re-synchronize its clock before validating AAs.

Upon loss of replay state (e.g., power cycle), a device shall discard existing OSCORE contexts and re-establish security context via EDHOC before accepting protected messages.

A device shall not process protected data-plane messages until it has successfully obtained a valid Access Token and synchronized its clock to Guardian Time.

### 8.8.2 Peer-to-peer AA retrieval resource

Implementations shall support peer-to-peer AA exchange using the following protected function resource:  
`/esta/e1.88/<version>/auth`

This resource shall support the following methods within the relevant Security Group context:

Method: GET (Retrieval)

Used by a device to solicit an AA from a peer (Pull).

- Request Payload: A CBOR map containing:
  - Key 1: requested Sender ID (sid) as bstr
- Response Payload: The COSE\_Sign1 AA object belonging to the requested Sender ID.

Method: POST (Pre-emptive Push)

Used by a Controller to push its AA to a Responder (Push) as defined in Section 8.8.4.

- Request Payload: The Controller's valid COSE\_Sign1 AA object.
- Behaviour: Upon receipt, the Responder shall verify the signature and cache the AA.
- Response Payload: A CBOR map containing:
- Key 1: The Responder's Identity Public Key (bstr).

### 8.8.3 Behavior on missing AA

If a Responder receives a protected message from Sender ID sid in context ctx and does not possess a valid cached AA for (ctx, sid), the Responder shall:

- treat the message as unauthorized and ignore it; and
- attempt to obtain an AA using GET /esta/e1.88/<version>/auth where feasible.
- Before initiating peer retrieval of an AA, a Responder shall wait a randomized backoff interval <aa\_backoff\_interval> to reduce request synchronization.
- A Responder shall not send more than 4 AA retrieval requests per second per (ctx, sid).
- A Responder shall cache negative results (no response / not found) for a short interval to avoid repeated bursts.

### 8.8.4 Pre-emptive AA & Pairwise Setup

Upon successfully receiving an Access Token, and prior to transmitting the first protected data plane message to a specific Responder, a Controller shall perform a pre-emptive Authorization Assertion (AA) exchange to establish both authorization and the cryptographic inputs required for Pairwise encryption.

1. The Controller shall send a request using the Signaling Transport Class (CoAP CON POST) to the Responder's /esta/e1.88/<version>/auth resource. The payload of this request shall be the Controller's valid AA (which includes its pubk claim).
2. Responder Verification: Responders shall verify the Guardian signature on all received AAs prior to caching and shall silently discard invalid assertions.
3. The Response: Upon successful verification, the Responder shall cache the AA to enforce the Controller's scope and extract the Controller's Public Key. The Responder shall then reply with a CoAP 2.04 Changed acknowledgment. The payload of this response shall be a CBOR map containing the Responder's own Identity Public Key.

By completing this exchange, both the Controller and the Responder possess each other's Public Keys. They shall immediately use these keys, in conjunction with the Group Master Secret, to derive the unique Pairwise keys required for data plane communication (Section 9.2).

This pre-emptive distribution requirement shall also apply to any subsequent re-issuance of an Access Token due to a key rotation or re-key event. Upon receiving an updated Access Token, a Controller shall distribute the corresponding new AA to its intended Responders prior to transmitting data plane messages protected by the new security context. This ensures Responders are primed with the necessary authorization information before the cryptographic handover occurs.

A Guardian should optimize scope assignment to minimize fragmentation.

## 9 Transport Layer

### 9.1 General

This section defines the Runtime Data Plane. FENCE uses lightweight [Group OSCORE] for low-latency lighting data transmission, following the initial EDHOC handshake.

Runtime communication shall be secured using Group OSCORE in one of two modes, depending on the transport mechanism:

- Pairwise Mode: Shall be used for all unicast communication. In this mode, devices derive unique keys to secure point-to-point data exchanges between a specific Controller and Responder.
- Group Mode: Shall be used for all multicast communication (e.g., synchronization packets, timecode). In this mode, a device encrypts a message once using the group context, allowing it to be decrypted by all authorized members of the Security Group.

### 9.2 Key Derivation

Devices shall derive unique Pairwise keys for unicast communication in accordance with the [GROUP OSCORE] specification.

To perform the required HKDF Pairwise derivation, a device shall use:

- Its own private key.
- The Group Master Secret provided in its Access Token (Section 8.6.1).
- The peer device's Public Key, obtained during the Unicast AA exchange (Section 8.8.4).

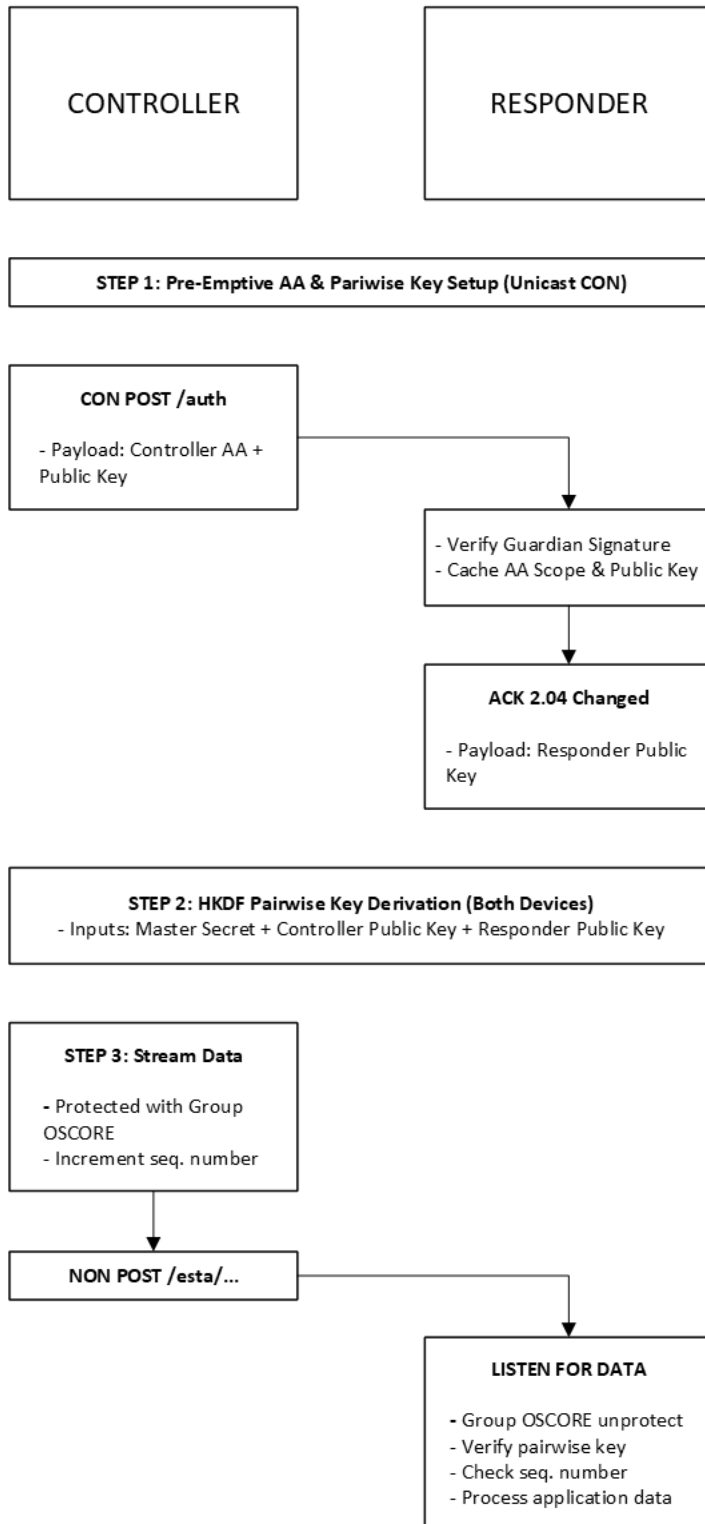
### 9.3 Security Processing

Replay Protection: Responders shall validate the OSCORE Sequence Number of every incoming packet against a sliding window.

Scope Enforcement: Responders shall validate that the target Universe in the CoAP URI-Path falls within the Scope granted to the Sender via the Authorization Assertion (Section 8.6.3). Packets targeting unauthorized Universes shall be silently dropped.

Responders shall maintain a counter of failed Authorization attempts (e.g., decryption failures or scope violations) and make this available to a Guardian via the protected resource `/esta/e1.88/<version>/diag`. As this resource facilitates direct communication between a device and a Guardian, it shall be protected using the pairwise OSCORE security context established during the EDHOC handshake. It shall not be accessible via the Group OSCORE context used for data plane communication.

Figure 5 Data Plane Communication Flow



### 9.4 Packet Format

This section defines the structure of a FENCE data plane message, which uses [Group OSCORE] to secure [CoAP] packets.

### 9.4.1 CoAP and Group OSCORE Relationship (Informative)

FENCE uses [Group OSCORE] to provide end-to-end security for [CoAP] messages. The Outer CoAP Header acts as the network transport and contains the necessary information for network devices to route the UDP packet, but it reveals nothing about the sensitive content.

The OSCORE Option and Encrypted Payload provides the secure layer. An observer on the network can see that a protected message is being sent but cannot read its contents or tamper with them without detection.

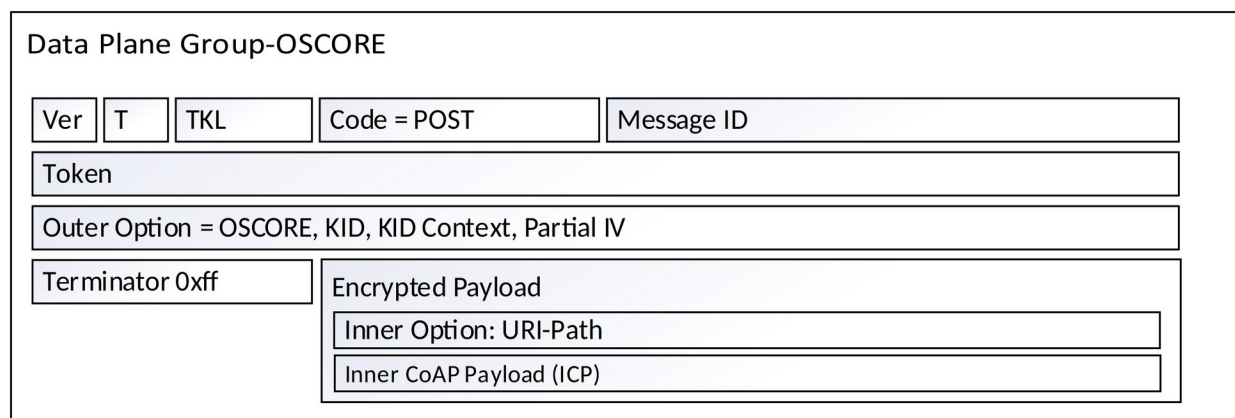
The Inner [CoAP] Message, which is contained within the Encrypted Payload, is the actual FENCE payload. It is a complete [CoAP] message with its own options (like the URI-Path) and payload, all of which are confidential and integrity-protected.

This layered approach, as described in Section 4.4.4, ensures that the high-latency negotiation on the control plane does not impact the low-latency requirements of the data plane, while providing robust security for all runtime communication.

### 9.4.2 Data Plane Packet Structure

The FENCE data plane packet format is a CoAP message containing an [OSCORE] Option, as shown in Figure 6. The following sections describe the fields and their function within FENCE.

Figure 6 Data Plane Group OSCORE Packet Format



### 9.4.3 Outer CoAP Header

The Outer [CoAP] Header contains unencrypted fields used by endpoints for message handling and routing.

- Ver (Version): The [CoAP] version number.
- T (Type): The [CoAP] message type shall be determined by the Transport Class being utilized (NON for Streaming, CON for Signaling). (Section 4.4.2).
- TKL (Token Length) and CoAP Token:

The Token is a value used to correlate requests and responses. The requirements for its use depend on the message's function:

For streaming data messages (e.g. POST), which are sent as Non-Confirmable messages and do not elicit a per-packet response, the CoAP Token serves no purpose. For these messages, TKL shall be 0, and the CoAP Token field shall be empty.

For request/response messages (e.g. GET), the CoAP Token is required for correlation. For these messages, TKL shall be 8 to provide the maximum token space for differentiating concurrent requests.

Under no circumstances shall the CoAP Token be used to encode any data that could be used to identify the payload, such as a universe number, as this would leak sensitive information in the unencrypted header.

- Code: The [CoAP] method. For sending streaming data, this is typically POST. Other methods like GET may be used for other functions, such as peer-to-peer AA retrieval (Section 8.8.2).
- Message ID: A 16-bit value used by [CoAP] endpoints to detect message duplication. This field shall not be used for replay protection or treated as a sequence number

#### 9.4.4 OSCORE Option and Encrypted Payload

This portion of the packet contains the security information and the protected application data.

- Outer Option = OSCORE: This [CoAP] Option signals that the message is protected with [OSCORE]. Its value contains the security context information required for decryption:
- KID (Key Identifier): In FENCE, this field carries the Sender ID of the transmitting device, as assigned by a Guardian in the Access Token (Section 8.6.1). This allows the Responder to identify the Controller, look up their cached Authorization Assertion (AA), and derive the correct pairwise key.
- KID Context: This field carries the Group Context ID, identifying the Security Group to which this message belongs. This ensures the correct Master Secret and security policies are applied.
- Partial IV (Initialization Vector): The core mechanism for replay protection (Section 9.3) is the [OSCORE] Sequence Number. This is a strictly increasing counter maintained as part of the security context by both the Controller and Responder. To ensure efficiency, the full Sequence Number is not transmitted in every packet.

Instead, the Partial IV field is sent, which carries a truncated portion of the full Sequence Number. This value acts as an efficient hint, allowing a responder to reconstruct the full Sequence Number for the incoming message and validate its freshness against a replay window. The reconstructed Sequence Number is then used to form the unique nonce required by the [AEAD] algorithm for decryption.

The [OSCORE] Sequence Number must not be confused with the unencrypted [CoAP] Message ID in the outer header.

- Terminator 0xff: A standard [CoAP] field that marks the end of the [CoAP] options and the beginning of the payload.
- Encrypted Payload: This is the ciphertext generated by the [Group OSCORE] process. It is an opaque block of bytes to any entity that does not possess the correct pairwise key. Decrypting this payload reveals the Inner [CoAP] Message.
- Inner Option: URI-Path: This is a [CoAP] option that is part of the inner, encrypted message. Its placement inside the encrypted payload is a critical security feature. It prevents network observers from performing traffic analysis, as the destination resource is confidential. The format of the URI-Path shall conform to the templates defined in Section 8.2.
- Inner [CoAP] Payload (ICP): This contains the actual application-level data. The ICP is not defined in this document. Please refer to the relevant E1.88-x document for details.

## 10 Key Management, Revocation and Online Enhancements

### 10.1 Key Classes

FENCE utilizes distinct classes of cryptographic keys to separate hardware provenance from network operation. These consist of:

Factory-Provisioned Keys (See Section 6.2):

- Onboarding Key Pair: Used exclusively during initial discovery to prove physical proximity via an Out-of-Band credential (e.g., QR code).
- Identity Key Pair: The permanent hardware identity of the device, attested by a Manufacturer CA.

Operational Trust Keys (See Section 4.5 and 10.2):

- FENCE Trust Root & Guardian Identity Keys: Long-term keys used to establish the local operational authority and authenticate the Guardian.

Runtime Keys (See Sections 8 and 9):

- Protocol Keys: Short-lived, ephemeral keys (e.g., EDHOC session keys, OSCORE contexts, and Group OSCORE epoch keys) used to secure the control and data planes.

## 10.2 Guardian Trust Model

A device's operational trust in a Guardian is established through a FENCE Trust Root certificate. This certificate acts as the local Certificate Authority (CA) for a specific venue or system's operational trust domain.

To support seamless, operator-initiated onboarding without requiring manual pre-configuration, devices shall employ a "Trust on First Use" (TOFU) and pinning mechanism:

1. Initial Discovery: While in the UNAUTHENTICATED state, a device shall accept the Guardian Identity Certificate presented during the initial EDHOC handshake (Section 7.3).
2. Trust Pinning: Upon successful completion of the Ownership Proof (Section 7.5) and receipt of its first Access Token, the device shall persistently store (pin) the FENCE Trust Root certificate that signed the Guardian's identity.
3. Operational Enforcement: For all subsequent connections, key rotations, and HA failovers, the device shall only trust a Guardian whose identity certificate successfully validates against this pinned FENCE Trust Root.

This pinned trust anchor shall only be removed if the device undergoes a secure local reset (Section 7.8).

## 10.3 Protocol Key Rotation

A Guardian shall support rotation of Group OSCORE key material using the epoch update mechanism defined in [ACE GROUP OSCORE].

Protocol Key rotation (including Group OSCORE keys, OSCORE contexts, and Access Tokens) is performed over the authenticated control plane. A device accepts these updates because they are sent within the existing pairwise OSCORE security context that it shares with the Guardian. This context remains trusted because its creation was dependent on the successful verification of the Guardian's long-term identity certificate against the device's pre-configured FENCE Trust Root. This separation of long-term identity from short-term operational keys allows for secure key rotation without the disruption of a full re-pairing.

This architectural separation between a device's long-term identity and its short-term operational keys is what enables this seamless rotation. A device's identity is established by its persistent Identity Key, possession of which is proven during the onboarding process (Section 7.5). The Protocol Keys used for the OSCORE context are ephemeral and derived from that handshake. Therefore, rotating the Protocol Keys does not invalidate the device's underlying identity, preventing the need for a full re-onboarding. Devices shall support a migration window during which messages protected under the previous epoch remain valid.

## 10.4 Device Revocation

A Guardian shall support revocation of Asset IDs via its local registry.

### 10.4.1 Revocation and Certificate Lifetimes

Traditional Public Key Infrastructure (PKI) relies heavily on end-devices processing Certificate Revocation Lists (CRLs) or the Online Certificate Status Protocol (OCSP). However, these mechanisms introduce significant bandwidth, memory, and connectivity overheads that are unsuited for air-gapped, low-latency entertainment networks.

FENCE explicitly omits the requirement for individual devices to process CRLs or OCSP. Instead, the Guardian is responsible for managing CRLs. The protocol achieves rapid peer-to-peer revocation through its centralized authorization model and short-lived credentials:

- **Operational Revocation:** Devices do not check the revocation status of their peers' X.509 certificates. Instead, if a device is revoked, the Guardian enforces this by (a) denying future EDHOC handshakes or token requests from that device, (b) relying on the expiration of the device's short-lived Access Token (the exp claim), and (c) immediately initiating a Group OSCORE rekey for the affected Security Group. This locks the revoked device out of the Data Plane without requiring peer devices to download a CRL.
- **Token Lifetimes (Functional Equivalent to CRLs):** Because the system relies on the expiration of the Access Token to evict disconnected malicious devices, the maximum lifetime of the token is a critical security parameter. A Guardian should default to an Access Token validity period of no more than 24 hours. For critical deployments, a Guardian shall provide the administrative ability to configure significantly shorter token lifetimes (e.g., 1 to 4 hours) to force faster revocation, balancing the security posture against the network overhead of token renewal traffic.
- **Manufacturing Revocation:** Only the Guardian evaluates Manufacturing Attestation certificates. If a manufacturer's hardware batch is compromised, the Guardian consumes the industry CRLs or revocation lists (see Section 10.4.4, Online Enhanced Mode). The Guardian enforces this policy internally during the onboarding phase, preventing compromised devices from joining the network.

### 10.4.2 Revocation Procedure

A revoked Asset ID:

- Shall be denied future control plane sessions.
  - During an Application Layer Onboarding attempt (Section 7), specifically after the device has presented its X.509 Identity Certificate in the Identity Claim phase (Section 7.4), the Guardian shall consult its internal registry (Section 11.2). If the device's Identity Public Key, as verified from the certificate, is associated with an Asset ID that is marked as revoked, the Guardian shall immediately terminate the onboarding process.
  - Any subsequent POST /esta/e1.88/<version>/token (Section 8.3) from a device whose asset\_id is marked as revoked shall be rejected by the Guardian with a 4.01 Unauthorized error, regardless of the validity of its prior EDHOC session.
- Shall lose access upon Access Token expiry or Group OSCORE rekey, whichever occurs first

Upon revocation of an Asset ID possessing current Group OSCORE material, a Guardian:

- Shall treat the revocation as a condition requiring immediate rekey
- Shall initiate a Group OSCORE rekey for all affected Security Groups
- Should notify all devices in the affected Security Group(s) of the epoch change

To mitigate the load on a Guardian during a re-key event, devices shall wait for a randomized backoff interval after receiving a re-key notification before sending a POST /esta/e1.88/<version>/token. This interval shall be a randomly selected duration in a range defined by <rekey\_backoff\_interval>.

Note: Revoking a device requires all other devices in the same Security Group(s) to obtain a new Access Token containing updated OSCORE input material from the Guardian. This re-key is critical. Because all pairwise keys are derived from this same OSCORE input material, a revoked device possessing the old material would still be able to derive the keys for, and decrypt traffic between, any other two devices in the group

### 10.4.3 Responder Behavior During Security Rekey

Upon receiving new Group OSCORE epoch material:

- Devices shall create a new Group OSCORE context with the updated epoch
- Devices shall maintain the previous epoch context for a migration window (<security\_key\_migration\_window>). During this window, a Responder shall be capable of processing messages protected under either the previous epoch or the new epoch.
- To process messages from the new epoch, a Responder must have successfully received and validated the Controller's new Authorization Assertion (AA) corresponding to that epoch.
- The Responder shall stop accepting messages under the previous epoch after the migration window expires and shall then destroy the previous epoch context.

The migration window for security-related rekeys should be significantly shorter than operational key rotation to minimize exposure.

### 10.4.4 Online Enhanced Mode

When operating in Online Enhanced Mode, a Guardian may enhance security by processing factory credential revocation information. This allows the system to reject the onboarding of devices whose factory attestation keys are known to be compromised.

Trusted sources for this information are typically the device manufacturer or an industry authority (such as ESTA). Such a source would publish a digitally signed, machine-readable revocation list. A Guardian shall only act upon such a list if it can cryptographically verify the signature against a pre-configured trusted source public key. To prevent replay attacks, this mechanism shall include rollback protection, such as version numbers or timestamps. Upon successful verification, the Guardian shall update its policies to deny future registration of devices whose attestation credentials match an entry in the revocation list.

To address security flaws discovered after deployment, manufacturers and Guardian implementers should establish a process for responsible disclosure. This process should include:

- A secure, private channel for receiving vulnerability reports from third parties.
- An internal procedure for assessing the severity and impact of confirmed vulnerabilities.
- A mechanism for distributing mitigations to end-users.

In the context of FENCE, mitigations may include, but are not limited to:

- Signed firmware updates to patch implementation flaws in a device.
- Publication of factory credential revocation lists to address the compromise of hardware keys.
- Software patches to correct security bugs in a Guardian implementation.

## 10.5 Connectivity and Trust Freshness

FENCE does not require continuous connectivity, global certificate revocation lists, or real time external validation for protocol correctness.

External connectivity allows faster detection and response to compromised credentials but is not required for secure operation.

When external connectivity is available, a Guardian may periodically retrieve:

- Updated Manufacturer CA trust information
- Manufacturer and batch revocation information

- Security advisories related to compromised credentials.

Retrieved information shall be cached locally and shall be usable during subsequent offline operation.

## 11 Guardian Implementation Requirements

The FENCE Guardian is the central authority and root of operational trust for a secure control network. It acts as both an Authorization Server and a Group Manager, responsible for validating the identity of all devices seeking to join the network, authorizing their permitted actions through the issuance of Access Tokens, and distributing the cryptographic material required for secure data plane communication. As the sole entity that establishes trust and defines security policy, the Guardian's own integrity and availability are paramount to the security of the entire system.

To fulfill this critical role, a Guardian implementation must itself be a secure and reliable system. This section therefore defines the normative requirements for a Guardian's internal security architecture, including the management of its trust stores and the protection of its device registry. Furthermore, it specifies the necessary functions for secure data management, high availability, and disaster recovery to ensure the operational resilience of the entire FENCE control plane.

### 11.1 Attestation Trust Store Management

A Guardian's trust store for Manufacturing Attestation is a security-critical component. Its management shall adhere to the following principles:

**Integrity Protection:** The trust store and any updates to it shall be integrity-protected. A Guardian should support the import of a list of trusted CAs that has been digitally signed by a trusted authority (such as ESTA).

**Administrator Control:** A Guardian shall provide a secure, authenticated interface for a local administrator to manually add or remove individual Manufacturer CAs from the trust store. This provides flexibility for closed systems and non-ESTA-member manufacturers. Additionally, an Administrator shall be able to configure the Guardian's policy regarding the onboarding of devices with un-attested identities (Section 4.5.1).

**Rollback Protection:** Any mechanism for updating the trust store (whether via an externally obtained list or manual edit) shall incorporate rollback protection, such as version numbering or timestamps, to prevent an attacker from forcing the Guardian to accept an older, potentially less secure configuration.

### 11.2 Registry Storage and Security

A Guardian shall maintain a Registry of all onboarded devices, their assigned Security Groups, and their cryptographic credentials (e.g., Public Keys and Sender IDs). To ensure the security of the system, this Registry:

- Shall be encrypted at rest.
- Shall be integrity-protected to prevent unauthorized modification.
- Shall be access-controlled to ensure only authorized administrators can modify device roles or scopes.
- Shall support a mechanism for state synchronization with one or more peer Guardians when deployed in a high-availability cluster. This synchronization channel shall be mutually authenticated, encrypted, and integrity-protected. The synchronized state shall include all device registry entries, Security Group definitions, and associated cryptographic material. The protocol for synchronization is implementation-defined.

**Informative Note:** While individual components of the registry, such as public keys, are not considered confidential, the registry as a whole is security sensitive. Mandating encryption for the entire registry at rest provides a robust defense-in-depth strategy.

### 11.3 Denial of Service Protection

A Guardian shall implement Denial of Service protections as specified in Section 7.9.

### 11.4 High Availability and Recovery

The FENCE Trust Root model facilitates both high-availability clusters and disaster recovery.

### 11.4.1 High Availability

Multiple Guardians can operate as a redundant cluster. To achieve this, all Guardians in the cluster shall be provisioned with identity certificates from a Certificate Authority whose certificate chain validates back to the FENCE Trust Root. Devices configured with this Trust Root will be able to validate the certificate of, and establish a trusted session with, any active Guardian in the cluster. Guardians in a cluster shall implement a mechanism to synchronize their device registries to provide seamless failover, as specified in Section 11.2.

### 11.4.2 Disaster Recovery

If all Guardians in a cluster fail permanently, a new replacement Guardian can be commissioned. The new Guardian obtains a unique identity keypair and obtains an identity certificate signed by a Certificate Authority in the same FENCE root of trust.

Because a device's trust is anchored in the FENCE Trust Root and not the specific key of an individual Guardian, it will accept a pairing attempt from this new Guardian. During the handshake, the device validates that the new Guardian's certificate was signed by the same trusted root, confirming its authority within the security domain. This allows a new trusted session to be established without requiring any re-configuration or reset of the device. An administrator can then restore a backup of the device registry to the new Guardian, allowing it to re-establish the control plane.

Informative Note: The private key of the FENCE Trust Root is the most critical security asset in an operational FENCE network. Its compromise would allow an attacker to impersonate a legitimate Guardian to all devices within that domain. System administrators should employ best practices for its protection, such as storing it on a dedicated hardware security module (HSM) or a secure hardware token, keeping it offline when not in use, and strictly limiting access.

## 11.5 Guardian Discovery

**This section is under construction.**

## 12 User Authentication and Authorization

**This section is under construction.**

To ensure proper separation of duties, a Guardian shall support at least two distinct user roles: Administrator and Operator.

### 12.1 Administrator

An Administrator shall have full control over security posture, trust stores, policy, and operator accounts.

### 12.2 Operator

An Operator shall have limited permissions for day-to-day operations, primarily onboarding devices and assigning them to pre-configured groups. An Operator shall not have permission to modify trust stores or system-wide policies.

## 13 Security Considerations

This section details the threat models considered during the design of FENCE and the rationale for specific architectural decisions.

### 13.1 Onboarding Attack Vectors

The FENCE onboarding model relies on a device discovering a Guardian and an operator with physical access authorizing the connection. This creates a window where an attacker on the same local network could attempt to interfere. The multi-step cryptographic verification process is designed to mitigate these threats.

**Vulnerability:**

An attacker on the same network segment can attempt two primary attacks:

- **Guardian Spoofing:** The attacker responds to the device's DNS-SD query with a fraudulent IP address, attempting to lure the device into connecting to the attacker's machine instead of a legitimate Guardian.
- **Device Claim Hijacking:** The attacker observes the device's unprotected POST to `/esta/e1.88/<version>/discover`, learns the device's IP and hash, and then races the legitimate Guardian to initiate an EDHOC handshake with the device.

**Mitigation:**

The FENCE protocol is secure against these attacks, which are reduced from a potential system compromise to a temporary and recoverable Denial of Service (DoS).

**Defense against Guardian Spoofing:** This attack is immediately defeated by the mandatory authentication of the Guardian during the EDHOC handshake. A device shall only trust a Guardian whose identity certificate successfully validates against a pre-configured FENCE Trust Root certificate (see Section 10.2). An attacker cannot produce a valid certificate signed by this trusted root. The device will detect the invalid certificate and shall abort the handshake, then continue its discovery process.

**Defense against Device Claim Hijacking:** This attack is defeated by the mandatory Proof of Ownership verification (Section 7.5). An attacker cannot complete the onboarding process because they do not possess the device's permanent, hardware-protected Identity Private Key, which is required to sign the Guardian's challenge. The device will not change its state to "onboarded," the attacker's connection will eventually fail, and the device will remain available for the legitimate Guardian to onboard.

**Residual Risk and Rationale:**

The residual risk is limited to a temporary Denial of Service. An operator may see a delayed or failed connection attempt if an attacker is actively interfering. The operational burden is minimal, and the device's cryptographic identity is never compromised. The ultimate mitigation for any persistent DoS remains the normative requirement for a physical factory reset mechanism, which ensures an operator with physical access can always recover a device.

## 13.2 Unattested Identity

During the onboarding process, if a device's X.509 Identity Certificate fails validation against the Guardian's Manufacturing Attestation Trust Store (Section 4.5.1), the device is considered to have an un-attested identity.

**Vulnerability: Lack of Verified Provenance**

Devices with un-attested identities (e.g., those using self-signed certificates, or certificates from Manufacturer CAs not configured as trusted by the Guardian) cannot be cryptographically verified as genuine products from a trusted supply chain. An attacker could potentially introduce a counterfeit or malicious device that generates its own keys and claims an identity.

**Mitigation: Administrator-Enforced Trust**

The FENCE protocol accommodates such devices within the single onboarding flow. However, the Guardian's administrative interface shall require explicit administrator approval to proceed with onboarding for devices with un-attested identities (Section 4.5.1). This administrative action acts as the trust anchor for the device's provenance. The Guardian still enforces cryptographic Proof of Ownership (Section 7.5) for the device's claimed identity, verifying it possesses the private key for its self-asserted certificate.

**Residual Risk and Rationale:**

The primary residual risk for devices with un-attested identities is the inability to cryptographically verify their manufacturing origin and integrity. Trust is shifted from the supply chain to the integrity of the local administrator and the security of the environment where the device is physically introduced. This means the administrator bears

the responsibility for vetting the physical device and its source. Strong physical security measures for devices and robust administrator training are essential to mitigate risks of impersonation or unauthorized device introduction for such cases.

### 13.3 On-Path Attacker (Passive and Active)

An attacker with access to the same network segment may observe, replay, inject, delay, or modify packets.

#### Mitigation

- All protected resources are carried using OSCORE or Group OSCORE.
- Payload confidentiality, integrity, and replay protection are enforced at the application layer.
- CoAP Uri-Path is protected inside OSCORE (Class E option), preventing resource targeting attacks.
- Sequence numbers and replay windows prevent reuse of captured packets.

#### Residual Risk

- An on-path attacker can still drop packets or introduce latency (Denial of Service).
- This is an inherent limitation of UDP-based, real-time systems.

#### Rationale

FENCE explicitly prioritizes freshness and low latency over guaranteed delivery. Preventing DoS at the network layer is out of scope and must be addressed through physical security and network design.

### 13.4 Guardian Compromise

#### Threat

A Guardian device or software is compromised, allowing an attacker to issue credentials, scopes, or group keys.

#### Mitigation

- A Guardian is explicitly the root of trust for the system.
- Guardian registry data is required to be encrypted and access-controlled.
- Devices authenticate Guardian identity during onboarding and pairing.

#### Residual Risk

A fully compromised Guardian compromises the security of the system.

#### Rationale

This is an accepted risk consistent with other centralized trust models (PKI CAs, Kerberos KDCs). Operational security of a Guardian is critical and must be treated as such. As FENCE is designed to function in air-gapped environments without reliance on traditional revocation mechanisms like CRLs, the recovery from a compromised Guardian is handled as an operational procedure: the compromised instance is taken offline, replaced, and a system-wide rekey is initiated (see Section 11.4.2).

### 13.5 Stale Authorization Information

#### Threat

Responders rely on cached Authorization Assertions that are outdated or no longer reflect current policy.

#### Mitigation

- AAs contain explicit expiry times.
- Devices validate expiry against Guardian-synchronized time.
- Missing or expired AAs cause traffic to be treated as unauthorized.

#### Rationale

Short-lived, signed, non-secret credentials balance security with offline operability. The system fails closed when Authorization information is unavailable.

### 13.6 Time Manipulation Attacks

#### Threat

An attacker attempts to manipulate time to extend credential validity or invalidate Authorization checks.

#### Mitigation

- Guardian Time is authoritative.
- Devices accept time only from authenticated control plane messages.
- Time is refreshed during token issuance.

#### Residual Risk

A compromised Guardian can manipulate time.

#### Rationale

Time authority is inseparable from trust in a Guardian and is consistent with offline operational requirements.

### 13.7 Compromised Device / Insider Threat

Threat: A legitimately onboarded device is compromised or attempts to exceed its Authorization.

#### Mitigation:

- Scope-limited Authorization restricts blast radius
- Dual enforcement: Controllers self-enforce; Responders verify via AA
- Group OSCORE Pairwise Mode limits group key exposure impact
- Firmware integrity requirements reduce exploitability

#### Residual Risk:

A compromised device can misuse its own authorized permissions.

#### Rationale:

FENCE limits blast radius through least privilege and sender-level Authorization, rather than attempting runtime compromise detection.

### 13.8 Firmware Integrity

To comply with regulatory requirements regarding the integrity of the execution flow:

- Devices should implement Secure Boot. The device should verify the digital signature or cryptographic hash of the firmware image against a trusted root of trust (e.g., in hardware or immutable bootloader) upon every initialization.
- Devices should verify the digital signature of any firmware update package prior to applying the update.
- If verification fails, the device should prevent the execution or installation of the unauthorized code.

## 14 Implementation Guidelines (non-normative)

### 14.1 Design Principles

FENCE implementations should follow these core principles:

#### Offline-First Operation

Internet connectivity is not required. Guardian connectivity is needed during onboarding and pairing but not during data plane operation. Time synchronization uses a Guardian, not external NTP.

#### Zero Trust Data Protection

Security is cryptographically bound to each message. Responders validate Controller authorization independently. Network segmentation provides defense-in-depth but is not required for security.

## Device Identity Models

All devices use a single, mandatory factory-provisioned identity model rooted in a supply chain trust. This ensures a consistent and secure onboarding experience for all FENCE products.

### Graceful Degradation

Devices with expired credentials shall not process protected data. Loss of Guardian connectivity shall not disrupt established data flows.

Protocol key rotation does not require device re-onboarding.

## 14.2 Key Generation

Devices that generate ES256 keypairs shall use a CSPRNG with sufficient entropy. Implementations should ensure the random number generator is properly seeded before key generation to prevent weak key material.

## 15 Glossary (informative / non-normative)

### Glossary

#### Access Token (CWT)

A private credential issued by a Guardian containing OSCORE input material and scope. Enables a device to derive the Group OSCORE context and participate in a Security Group. See Section 8.6.1.

#### Administrator

A privileged user role responsible for managing a Guardian's core security configuration.

#### Attestation

The process by which a device proves its identity and provenance to a Guardian. It involves a multi-step verification where the Guardian first validates a device's physical proximity via an Onboarding Key and then validates its cryptographic ownership of a permanent, factory-signed Identity Certificate.

#### Attestation Trust Store

The collection of trusted X.509 root and intermediate Manufacturer CA certificates configured in a Guardian. It is used to validate the manufacturer's signature on a device's X.509 Identity Certificate during the onboarding process.

#### Asset ID

A unique identifier that is factory provisioned to a device. It is 128-bit random integer generated using a Cryptographically Secure Pseudo-Random Number Generator (CSPRNG). This identifier is static and does not change for the life of the device.

#### Authorization Assertion (AA)

A Guardian-signed, non-secret credential binding a Sender ID to specific permissions within a Security Group. Distributed peer-to-peer to enable Responders to verify Controller Authorization. See Section 8.6.3.

#### C\_I (Connection Identifier)

A session identifier used in EDHOC to distinguish between concurrent handshakes without maintaining full session state. Generated by the initiator.

#### Cipher Suite

A defined set of cryptographic algorithms (key exchange, encryption, integrity protection) negotiated during EDHOC handshake.

#### Control Plane

The communication channel between a Guardian and devices, secured using EDHOC, ACE-OAuth, and OSCORE. Used for authentication, Authorization, and key distribution. See Section 4.4.3.

### Control Transport Class

A formally defined network abstraction utilizing Authenticated [OSCORE] over [CoAP]/[UDP] with Confirmable (CON) messages. This class is mandated for all Control Plane communications between a Guardian and a Device. See Section 4.4.2.

### COSE (CBOR Object Signing and Encryption)

RFC 9052. A framework for signing and encrypting data using CBOR encoding.

### CSPRNG (Cryptographically Secure Pseudo-Random Number Generator)

A random number generator suitable for cryptographic key generation, providing sufficient entropy and unpredictability.

### CWT (CBOR Web Token)

RFC 8392. A token format for conveying claims between parties, using CBOR encoding. Used in FENCE for Access Tokens.

### Data Plane

The communication channel between devices, secured using Group OSCORE in Pairwise Mode. Used for real-time lighting control data. See Section 4.4.4.

### Device

A Controller or Responder that participates in a FENCE network after authentication and Authorization. See Section 4.1.

### DNS-SD (DNS-Service Discovery)

[RFC 6763]. A protocol that uses standard DNS and mDNS to allow devices to discover services on a network without manual configuration. FENCE uses DNS-SD for un-onboarded devices to discover Guardians.

### EDHOC (Ephemeral Diffie-Hellman Over COSE)

RFC 9528. An authenticated key exchange protocol optimized for constrained environments. Used to establish secure channels between devices and a Guardian.

### Entity ID

A human-readable name (e.g., "FOH-Console-1"), assigned by an operator to a device after onboarding. The Entity ID is a logical name and is distinct from hardware-specific identifiers like serial numbers, Asset ID or MAC addresses.

### Ephemeral Key

A temporary key pair generated for a single session and discarded after use. Provides forward secrecy by preventing compromise of long-term keys from affecting past sessions.

### ES256

ECDSA signature algorithm using P-256 curve and SHA-256 hash. Specified in RFC 9053 for use with COSE.

### FENCE Trust Root

The root X.509 certificate of a local Certificate Authority (CA) that establishes the single, authoritative source of trust for a specific FENCE operational security domain. Its sole purpose within the FENCE protocol is to sign the identity certificates of one or more Guardians, thereby conferring them with authority.

### Factory Attestation

Manufacturer-provided cryptographic evidence of a device's identity and provenance. In FENCE, this takes the form of an X.509 certificate containing the device's Identity Public Key, signed by a trusted Manufacturer CA. See Section 6.2.1.

### Group OSCORE (Group Object Security for Constrained RESTful Environments)

RFC 9594. Extension of OSCORE for group communication. FENCE uses Pairwise Mode for unicast data plane traffic.

**Guardian**

The Authorization server that controls authentication and Authorization for all devices in a FENCE network. Acts as the root of trust and time authority. See Section 4.1.

**Guardian Time**

The authoritative time source provided by a Guardian. All devices synchronize their clocks to Guardian Time during token issuance. See Section 4.2.7.

**Identity Key Pair**

A permanent, factory-provisioned asymmetric key pair representing a device's long-term identity.

**Master Secret**

The shared cryptographic material distributed within an Access Token that enables devices to derive Group OSCORE contexts and pairwise keys.

**Offline Mode**

Operational mode where a Guardian functions without external connectivity, relying solely on locally configured trust information. See Section 4.2.8.

**Onboarding Key Pair:**

A factory-provisioned asymmetric key pair used exclusively to establish the initial secure channel during onboarding.

**Online Mode**

Operational mode where a Guardian has external connectivity and may automatically retrieve updated trust and revocation information. See Section 4.2.8.

**OSCORE (Object Security for Constrained RESTful Environments)**

RFC 8613. Application-layer security protocol providing confidentiality, integrity, and replay protection for CoAP messages.

**Operator**

A user role with limited permissions for performing day-to-day tasks such as device onboarding.

**Out-of-Band (OOB) Credential**

A physically associated identifier (e.g., QR code, RFID tag) containing cryptographic information derived from a device's factory provisioning, used by an operator to initiate device onboarding and prove physical proximity.

**Pairing**

The successful establishment of a mutually authenticated, secure session between a device and a Guardian, which is the end result of the onboarding process.

**Pairwise Mode**

Group OSCORE operational mode where devices derive unique unicast keys for each Controller-Responder pair, providing Controller authentication within the group.

**Protected**

A term used to describe any data, message, or resource that is secured using the cryptographic mechanisms defined by FENCE, specifically OSCORE and Group OSCORE. A protected item benefits from confidentiality, integrity, and replay protection, and its access is subject to authentication and authorization.

**Responder**

A device that consumes lighting control data after authentication and Authorization. Examples: moving light, dimmer, DMX512 gateway. See Section 4.1.

**Scope**

The set of permissions defining which resources a device may READ or WRITE. Expressed as CBOR-encoded rules with universe ranges. See Section 8.1.1.

**Security Group**

A set of devices sharing common cryptographic material (Master Secret) and operating within a single Group OSCORE context. Typically represents a logical or physical boundary. See Section 8.1.1.

**Sender ID**

A unique identifier assigned by a Guardian to each device within a Security Group context. Used in Group OSCORE for source identification and key derivation.

**Signaling Transport Class**

A formally defined network abstraction utilizing [Group OSCORE] over [CoAP]/[UDP] with Confirmable (CON) messages. This class is mandated for Data Plane communications that require application-layer acknowledgement, such as the exchange of Authorization Assertions between peers. See Section 4.4.2.

**Streaming Transport Class**

A formally defined network abstraction utilizing [Group OSCORE] over [CoAP]/[UDP] with Non-Confirmable (NON) messages. This class is mandated for Data Plane communications that require low latency, such as real-time lighting control data. See Section 4.4.2.

**Time-To-Live (TTL)**

The validity duration of a credential. Access Tokens have longer TTLs (typically 24 hours).

**TPM (Trusted Platform Module)**

A hardware security component providing protected storage for cryptographic keys and secure cryptographic operations. Used for protecting a device's Onboarding and Identity private keys.

**Controller**

A device that generates lighting control data after authentication and Authorization. Examples: lighting console, house lights controller. See Section 4.1.

**Un-attested Identity**

A device identity (represented by its X.509 Identity Certificate) that fails to validate against a Guardian's configured Manufacturing Attestation Trust Store. Such an identity requires explicit administrator approval to be onboarded.

**UNAUTHENTICATED**

Device state before successful pairing with a Guardian. Devices in this state cannot process protected control data.

**Appendix A: Numeric Recommendations (Informative)**

<security_key_migration_window>	Recommended: 5 seconds for security rekeys
<port_coap>	5683
<aa_backoff_interval>	Recommended range 50ms to 300ms
<rekey_backoff_interval>	Recommended range 100ms to 2s
<announce_backoff_interval>	From Minimum of 1s
<edhoc_backoff_interval>	From Minimum of 1s

## Appendix B: Compliance Profiles (Normative)

To be compliant with this standard, an implementation shall implement the FENCE-2026-BASE profile.

### B.1 Profile: FENCE-2026-BASE

This profile defines the baseline set of cryptographic algorithms for interoperability.

- Factory Key Pair Algorithm: ES256 (ECDSA over P-256 with SHA-256).
  - This algorithm shall be used for both the Onboarding Key Pair (<onboarding\_keypair>) and the Identity Key Pair (<identity\_keypair>) defined in Section 6.2.1.
- Profile AEAD Algorithm: AES-CCM-16-64-128 (COSE Algorithm ID 10).
  - This algorithm shall be used for OSCORE (Control Plane) and Group OSCORE (Data Plane) encryption and integrity.
- Profile ECDH Algorithm: P-256.
  - This algorithm shall be used for Group OSCORE (Data Plane).
- Profile Hash Algorithm: SHA-256.
- Profile KDF Algorithm: HKDF-SHA-256.
  - This algorithm shall be used for Group OSCORE key derivation (deriving Pairwise keys from the Master Secret).
- Profile Guardian Signature Algorithm: ES256 (ECDSA over P-256 with SHA-256).
- Profile EDHOC Suite:
  - Key Agreement: ECDH-ES + HKDF-256
  - Application AEAD: AES-CCM-16-64-128
  - Hash: SHA-256
  - Supported Curve: P-256

### B.2 Ownership Proof Structure

For the Proof of Ownership step (Section 7.5), the device shall sign a deterministically encoded CBOR map that includes the challenge nonce and context. This structure ensures that the signature is unambiguously bound to the specific Proof of Ownership operation.

The OwnershipProofChallenge structure is defined using CDDL as:

```
ownership-proof-challenge = {
  1 => tstr ; challenge_type: Fixed string "fence-ownership-proof-v1"
  2 => bstr ; nonce: Cryptographically secure random nonce from Guardian
}
```

Implementations shall ensure that the challenge\_type field is set to the fixed string "fence-ownership-proof-v1" for all Proof of Ownership challenges under this profile. The device shall deterministically CBOR encode this ownership-proof-challenge map and sign the resulting byte string.