Mode-Aware Object-Level Security for EtherNet/IP Devices Using CIP Security in Industrial Network

Shyam Sundar Pal, Firmware Architect
Riya Rahul Shah, Senior Functional Safety Lead
Utthunga Technologies

Presented at the ODVA
2026 Industry Conference & 24th Annual Meeting
October 21, 2026
Vilamoura, Portugal

1. Abstract

Securing EtherNet/IP endpoints cannot rely only on perimeter controls. Firewalls, segmentation, VPNs, and (D)TLS protect communication channels, but they do not by themselves decide whether a specific CIP object, service, or operational command is authorized for a given requester in the device's current state or mode.
This paper proposes a fail-closed, post-(D)TLS endpoint enforcement pipeline for EtherNet/IP devices that composes with CIP Security rather than replacing it. After session establishment, the contribution is to enforce header or metadata inspection (L1), session binding validation (L2), operational-mode gates (L3), and CIP object or service authorization (L4) before the CIP Object Handler executes.

2. Keywords

EtherNet/IP, Common Industrial Protocol (CIP), CIP Security, industrial device security, device-level authentication, session validation, state validation, access control, mutual agreement, Volume 8, object-level policy, operational-mode gates, interoperability.

3. Scope and Non Goals

Paper contribution: This paper's primary contribution is the fail-closed, post-(D)TLS endpoint enforcement pipeline described in Level Details Flow and the L1–L4 sections. It composes existing CIP Security transport and profile capabilities with endpoint-side header/metadata inspection (L1), session binding (L2), optional operational-mode gating (L3), and CIP object or service authorization (L4) before the CIP Object Handler executes.

This work is scoped as an endpoint enforcement extension to CIP Security. It does not define a new secure transport, certificate model, cryptographic mechanism, or replacement for existing CIP Security profiles. It assumes CIP Security profiles provide authenticated and protected communication where configured. The proposed extension adds device-local policy enforcement for header/metadata inspection, session binding, operational-mode validation, and CIP object/service authorization.

4. Definition of terms

Session context: The requester identity, authenticated transport association, assigned role, timeout state, and agreed authorization scope tracked for an active session.

 ©2026 ODVA, Inc.

CIP Object Handler: The device-side implementation that executes an allowed CIP object or service request after endpoint policy checks succeed.

Device Policy Engine: The local endpoint enforcement logic that evaluates header/metadata inspection (L1), session binding (L2), operational-mode gates (L3), and CIP object or service authorization (L4) before control reaches the CIP Object Handler.

EtherNet/IP: An industrial Ethernet network that uses the Common Industrial Protocol (CIP) for device communication and services.

Common Industrial Protocol (CIP): An object-oriented industrial protocol used to exchange control, status, configuration, and diagnostic data between devices.

CIP Security: The ODVA-defined security extension for CIP and EtherNet/IP that adds endpoint authentication, message integrity, confidentiality, and trust management.

Requestor: A client such as a PLC, engineering workstation, HMI, or software application that initiates communication with the device.

Session validation: Verification that a request belongs to an active, authenticated session and satisfies session-specific checks such as identity, timeout, and key matching.

Application-layer CIP policy: Device-side checks after transport security header or metadata sanity, session binding, operational state, and object or service ACL

State validation: Verification that a requested operation is permitted in the device's current operational state.

CIP object access control: Restriction of access to industrial protocol objects, attributes, and services based on requester role, authenticity, and permitted operations.

Device based firewall: L3/L4 traffic filtering at the device (host firewall). Not a substitute for CIP object or service authorization which is post DTLS application policy.

CIP Security: The ODVA-defined security extension for CIP and EtherNet/IP that adds endpoint authentication, message integrity, confidentiality, and trust management.

Authorization profile (CIP Security direction): Evolving highly configurable authorization described in the CIP Networks Library[3] [3] . In Volume 8 Edition 1.17 (November 2023) the CIP Authorization Profile is listed as reserved for a subsequent edition.. This paper’s object-level mutual agreement or ACL is practical enforcement toward that profile. They are not a replacement for or redefinition of the ODVA Authorization Profile specification.

Related Work and Positioning

CIP Security[4] ODVA, CIP Security / Volume 8 of the CIP Networks Specification. Public overview: already provides endpoint authentication, confidentiality, integrity protection, and profile-based security mechanisms for EtherNet/IP. Prior EtherNet/IP security work, including ODVA deep packet inspection guidance and broader industrial control system literature also addresses traffic inspection and device compromise scenarios.

Urbina et al. [1] and Yu, Wang, and Song [2] are included because they represent two common gaps in prior EtherNet/IP security work. It demonstrated fieldbus compromise without sufficient endpoint command validation and detection-oriented approaches that do not guarantee inline prevention. They are cited as motivating background, not as the proposed solution.

Additional references in this paper include ODVA CIP Security, ODVA EtherNet/IP DPI guidance, IEC 62443, and NIST OT security guidance. The related-work scope is intentionally focused on endpoint

authorization composition for EtherNet/IP rather than on general IT security or unrelated industrial protocols.
This paper does not claim to redefine transport security, certificate handling or the ODVA Authorization Profile as per specification. Its contribution is a **practical endpoint composition pattern that operates with CIP Security**.

Threat Model and Scope
The scope of this paper is limited to endpoint authorization decisions after an authenticated communication context already exists. The representative attack chain is that a requester has network reachability and a valid or accepted security context, yet still attempts a CIP object access or operational command outside its allowed role, session scope, or device mode.
Typical attack chains include: (1) a compromised but ***trusted device attempting commands outside its assigned authorization*** or (2) an adversary with a ***valid certificate or security context attempting object or service access*** beyond configured policy. This paper does not address transport compromise, certificate issuance policy, or replacement of existing device safety logic.

5. Core Problem

Network-level security and authenticated transport alone cannot guarantee that only authorized and context-appropriate CIP operations are executed by a device.
Even when firewalls, segmentation, VPN, and TLS are correctly deployed the remaining concern is endpoint-side misuse of an otherwise accepted communication path. The following categories summarize that narrower problem space:

5.1.1.Encrypted malicious payload:
CIP Security and (D)TLS address authentication and transport protection for this category. The remaining gap is endpoint-side authorization: deciding whether the authenticated requester may perform the requested CIP object access or command in the current device context.
TLS/VPN protects confidentiality and channel integrity but does not validate whether a CIP command or object access is authorized for that requester.

5.1.2.Detection without prevention:
This category motivates inline fail-closed endpoint rejection rather than alert-only detection.
IDS/NIDS primarily detects or alerts on suspicious behavior; unless deployed inline with blocking enforcement, it does not guarantee that malicious CIP requests are rejected before reaching the device handler.

5.1.3.No end-to-end command trust:
Existing approaches often lack authentication, packet integrity enforcement, and command validation at the device application protocol layer.

5.1.4.Stateful process abuse:
CIP Security authenticates the requester but does not by itself define complete operational-mode authorization for every device implementation.A requester may be network-authenticated but attempt an operation incompatible with the device's current operational state (e.g., configuration change and other non-permitted operation during runtime mode)

5.2. Gaps in existing methodology

5.2.1. Insufficient device-layer security
Current published approaches that focus on network access control do not adequately address the following four device level capabilities:

After CIP Security established authenticated transport, the endpoint still requires explicit authorization checks for

- Authentication of the requester at the protocol level

- Packet integrity enforcement beyond transport checksums
- Command verification & validation: Verifying that the requested operation is policy-wise permitted

As a result, a packet that reaches the device may be accepted and processed even if the command is unauthorized or malicious.

5.2.2. Detection-oriented but not prevention-oriented **[2]** monitors traffic patterns and raise alerts but typically do not stop the device from:

- Parsing malicious CIP or EtherNet/IP messages
- Allowing to change device state in violation of safety

5.3. Problem Statement Summary

In industrial networks traditional security mechanisms including firewalls, network segmentation, VPN, and TLS wrappers are applied at the network and transport layers. Firewalls, segmentation, VPN and DTLS secure and filter the channel but do not fully decide CIP command legitimacy at the endpoint device. Therefore, it cannot fully prevent compromised or malicious packets from reaching and being processed by devices. Existing literature does not consider extensive device-level authentication and relies on intrusion detection without prevention allowing malicious data to be executed on the device.

CIP Security Spec Volume 8[3] [3] [3] (Edition 1.17, November 2023) improves authentication and channel protection. In that edition the CIP Authorization Profile is identified as reserved for a subsequent Volume 8 release; this paper does not claim it is already defined there. This paper composes with existing CIP Security transport and profiles. What remains for practical deployments is endpoint composition: binding authenticated identity to session context and enforcing object-level, mode-aware authorization inline before the CIP handler—toward the Authorization Profile without claiming to define its standardized handshake.

Accordingly, the practical gap addressed by this paper is endpoint-side enforcement after CIP Security authentication: a fail-closed sequence of header or metadata validation, session binding, operational-mode checks where implemented, and CIP object or service authorization under explicit policy.

5.4. Impact on the network device if not correctly addressed

The impacts below follow directly from the endpoint authorization gap described in the threat model and problem summary:

If the endpoint enforcement gap described above is not addressed, industrial devices remain exposed to the following impacts even when perimeter controls and (D)TLS are correctly deployed:

- Malicious firmware deployment or configuration changes on devices
- Denial-of-service through session expiration or state anomaly or mismatch
- Single compromised device to entire devices in the network
- Non-compliance with IEC 62443 requirements for least privilege at the device layer

6. Overview — What Problem proposed solution solves !

Industrial devices (sensors, actuators,) expose data and configuration through CIP objects. Without security, any client on the network could read process data, change device parameters, send unauthorized control commands, or impersonate a legitimate PLC, controlling device or engineering tool.

The mechanism addresses this with four security layers forming a defense-in-depth on the CIP endpoint. Each layer must pass before the CIP service executes. If there is any failure it will reject (fail-closed).

Problem: EtherNet/IP devices expose process data, configuration, and control through CIP objects. Even when a requester is accepted at the transport and authentication layers, the endpoint still needs to determine whether the specific requested object access or command is allowed for that requester in the current device context.

Without endpoint policy, an authenticated or otherwise accepted requester could still read process data, change parameters, or issue operational commands outside its intended authorization scope.

Security Level	Purpose
1. Header and Metadata Inspection	Reject malformed or out-of-context requests before deeper handling
3. Session Binding per Requester	Validate requester identity, session context and timeout
2. Operational-Mode/State Gate	Allow only operations valid for the current device state or mode
4. CIP Object/Service Authorization	Restrict object, attribute, and service access by requester role and explicit policy

7. Proposed Endpoint Enforcement Architecture

This section is organized in three parts: (1) endpoint enforcement principles, (2) the proposed per-request user pipeline that is the main contribution of the paper, and (3) differentiation, supporting operational recommendations, and mapping to existing CIP Security objects and profiles. The pipeline is presented before comparison so the reader knows what is being proposed before seeing how it relates to prior work and standards.

- Fail & Deny: Rejection of any validation failure
- Header-first: Inspect metadata before payload handling
- Endpoint enforcement: Decisions occur on the device
- Least privilege: Minimum object permission per requester
- Stateful sessions: Independent session context per requester after (D)TLS.
- State-aware context: Operation must match device mode
- Mutual agreement: Explicit object-access contract between device and requester

7.1. Level Details Flow

Figure 1 provides an overview of the proposed per-request enforcement order. For consistency, the paper refers to four enforcement levels: Level 1 (L1) Header and Metadata Inspection, Level 2 (L2) Session Binding per Requester, Level 3 (L3) Operational-Mode Gate, and Level 4 (L4) CIP Object or Service Authorization. Figure 1 is shown first as a roadmap; the sections below explain session-establishment prerequisites and then each enforcement stage in detail.

These enforcement levels are sequential checks in a single endpoint policy pipeline. L1–L4 are not OSI network layers and are not independent architecture layers; they may share policy knowledge and are not presented as fully independent levels in the traditional layered-design sense. The contribution is the fail-closed composition order, not the claim that each decision source is independent. L2 session binding and L3 operational-mode gating require device-side knowledge of CIP object semantics, allowed services, and operational state—not merely network-layer rules; the Authorization policy model below describes representing that knowledge once for reuse across stages.

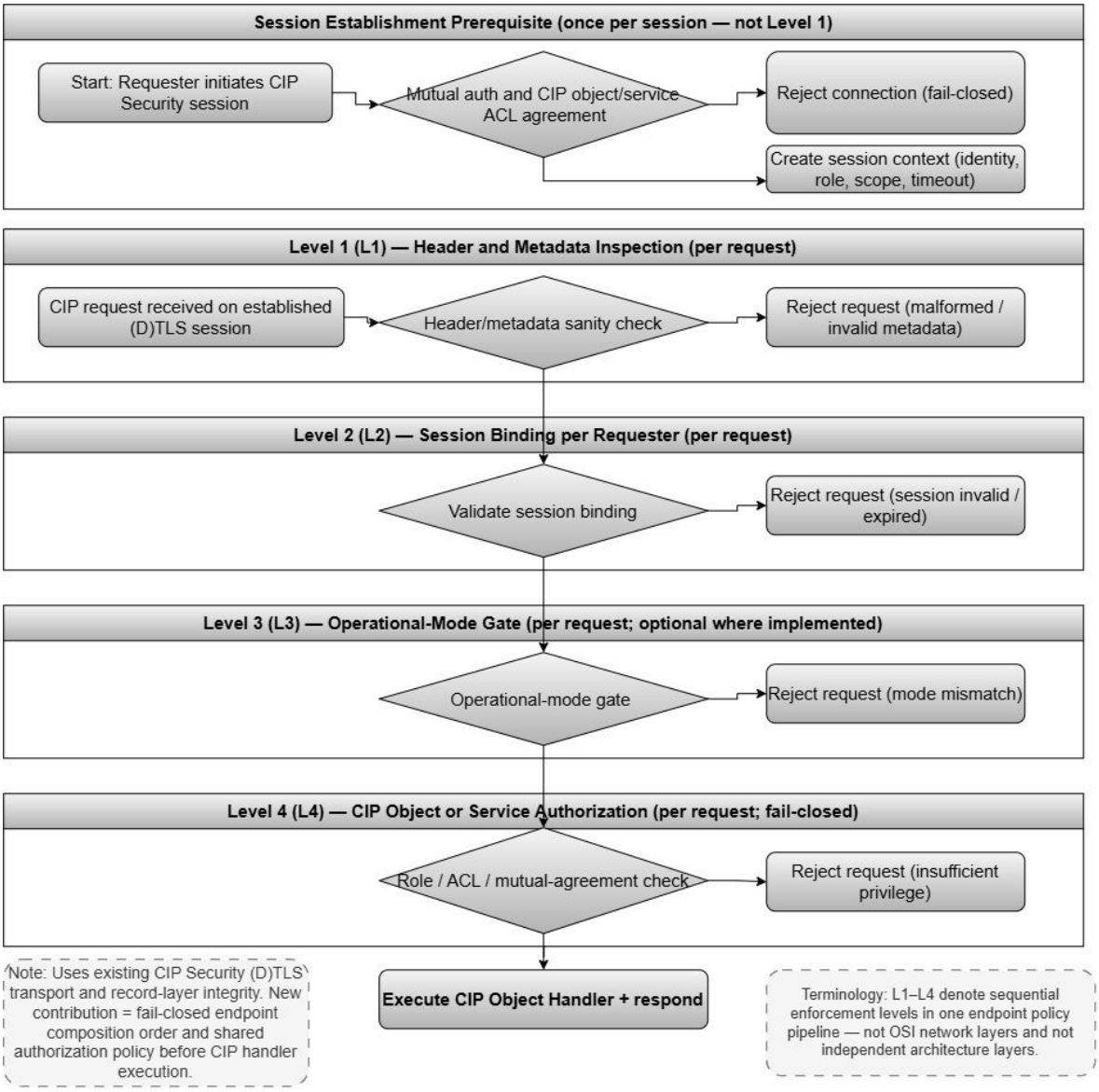


Figure 1 — Multi-Level CIP Endpoint Enforcement Pipeline

7.2. Session Establishment Prerequisite: Mutual Agreement Between Device and Requestor

Before any data exchange, both sides must agree that the other party is trusted and that the security context is acceptable. This is mutual authentication, which is not just the device verifying the client but also the client verifies the device.

Purpose : Mutually authenticate requester and device using CIP Security, then agree the allowable CIP object/service set (role/ACL) for the session.
Flow : Authenticate via (D)TLS → bind identity/role to session → negotiate object-access agreement → require session binding on subsequent requests → apply operational mode gates and ACL → execute or reject.

Failed Reason : Not enrolled, Authentication failed, Key expired, Key exchange rejected.

7.2.1. Mechanisms Used

The mechanisms below reuse standard CIP Security authentication methods (X.509 certificates, PSK, User Authentication, and TLS/DTLS handshake). The paper does not define a new transport protocol. It specifies how those existing mechanisms establish the session context used by the endpoint policy pipeline.

7.2.2. Agreement conditions

Agreement conditions: Both device and requester must satisfy the following before session context is created: (1) successful mutual authentication via CIP Security (D)TLS (2) acceptable trust domain (CA or PSK), (3) compatible security profile and (4) explicit mutual agreement on the allowable CIP object and service permissions for the session. If any condition fails, the connection is rejected fail-closed with no object access.

On Failure: If mutual agreement fails, the connection is rejected entirely. No object access is possible.

7.2.3. L1 Header and Metadata Inspection

Purpose: Perform the first-pass decision using EtherNet/IP encapsulation and CIP header or metadata fields before deeper request handling.

Inspected fields : encapsulation command, connection ID, sequence count, service code, CIP path, message type.

Functions : validate structure, extract metadata, detect replay, forward normalized metadata to next stage L2-L4.

Failed Reason : Invalid length, Unsupported command, Malformed path, Invalid connection

7.3. L2 Session Binding per Requester

Once mutual agreement is established, the device creates a session record binding identity, role, timeouts. Subsequent packets are validated against this binding under the already-established CIP Security transport.

Purpose: Maintain and validate independent session context for each requester on every request after the authenticated transport context is established.

Checks : Session active, Key, Source identity matches for enrolled profile.

Failed Reason : Session not found, Key mismatch, Session expired, Session binding invalid.

7.3.1. Session Record Contents

7.3.2. User Per-Request Session Validation

The following conditions shall be checked for per-request session validation:

- Does a session exist for this Session ID?
- Is the session still active or expired?
- Does the unique key in the packet match the session's key mechanism?
- Is the requester authenticated?

If any of the check fails then packet is rejected.

7.3.3. Session Binding under CIP Security Transport

Existing mechanism: CIP Security already provides (D)TLS authentication, session keys, record-layer integrity, and confidentiality.

New in this paper: binding the authenticated requester identity and agreed authorization scope to each subsequent CIP request and enforcing application-layer policy before the CIP Object Handler executes.

CIP request on established (D)TLS session → header/metadata check → validate session binding → operational-mode gate → role/ACL → execute or fail-closed reject.

The device treats (D)TLS as the cryptographic channel and enforces session binding plus application policy on each request.

7.3.4. CIP Security Standard Roles

CIP Security Standard Roles referenced in this paper (Administrator, Engineer, Operator, Viewer, Auditor) follow the User Authentication profile role model. These roles define baseline privilege levels; the endpoint policy pipeline (L2–L4) enforces the session-specific ACL and mutual agreement established at setup.

7.4. L3 Operational-Mode Gate

Security purpose: This stage prevents authenticated but context-inappropriate commands from reaching the CIP Object Handler. It does not replace the device's safety or application state machine. It exposes that decision as an explicit authorization gate in the composed endpoint policy.
The device maintains an operational state that defines what it is currently doing and what operations are allowed. Where this logic already exists in firmware, the proposal is to reference that decision in the authorization pipeline rather than duplicate the entire state machine in separate security logic.

Purpose: Ensure the requested command is allowed in the device's current operational state. Where vendors already enforce mode restrictions locally, this stage formalizes the check as part of the composed endpoint policy rather than relying on ad hoc handler logic alone.

Many EtherNet/IP devices already implement mode-dependent restrictions in firmware (for example, blocking parameter writes while Running or requiring Configuration mode for certain downloads). This behavior is common vendor practice but is not universally specified by CIP Security Volume 8 beyond limited commissioning exclusivity via the CIP Security Object. This section describes how such mode gates compose with CIP Security transport and application-layer ACL as an explicit, fail-closed pipeline stage.
The contribution is enforcing them consistently in a single fail-closed pipeline—after (D)TLS, before the CIP Object Handler—together with session binding and mutual agreement on object permissions.

7.5. L4 CIP Object or Service Authorization

Authorization policy model: The endpoint stores a compact policy set containing requester role, allowed CIP object and service permissions, operational-mode and the session-specific agreement established at setup. L2, L3, and L4 read from this shared policy context rather than maintaining four independent authorization databases. One policy engine applies sequential checks with early rejection—not four independent authorization evaluations.

After authentication and state checks, the device evaluates whether the specific request is allowed on the targeted CIP object, attribute, and service for the requester's assigned role and agreed session scope.

Purpose : Restrict CIP object or service access by requester type and mutual agreement.
Decision rule : If requested object or service is not in requester ACL then reject. Otherwise allow and forward to CIP Object Handler.
Failed Reason: Object not permitted, Service not permitted, Agreement not found, Insufficient permission.

Fail-closed behavior: The device must not silently ignore the request, partially allow unauthorized actions, or fall back to a lower privilege without explicit policy.
Steps for per-request enforcement determination

- Requestor sends packet (with Session ID)
- Device performs header/metadata sanity checks (encapsulation/CIP fields).
- Device validates session binding
- Device evaluates operational-mode gate for the requested service
- Device evaluates role or ACL for object permissions

If all checks pass, forward to CIP Object Handler; otherwise fail-closed reject.

7.6. Figure 2 — Sequence Diagram:

This sequence diagram illustrates one request path through the proposed endpoint policy. Session context means the ***requester identity, authenticated transport association, assigned role, timeout state, and agreed authorization scope*** tracked for the active session.

Purpose of the diagram: It shows the decision order from received CIP request through Device Policy Engine enforcement levels L1–L4 checks to allowed execution by the CIP Object Handler or fail-closed rejection.
In this paper, the CIP Object Handler executes an allowed CIP object or service request, while ***the Device Policy Engine evaluates L1 header/metadata inspection, L2 session binding, L3 operational-mode gates, and L4 object-level authorization*** before control reaches that handler.

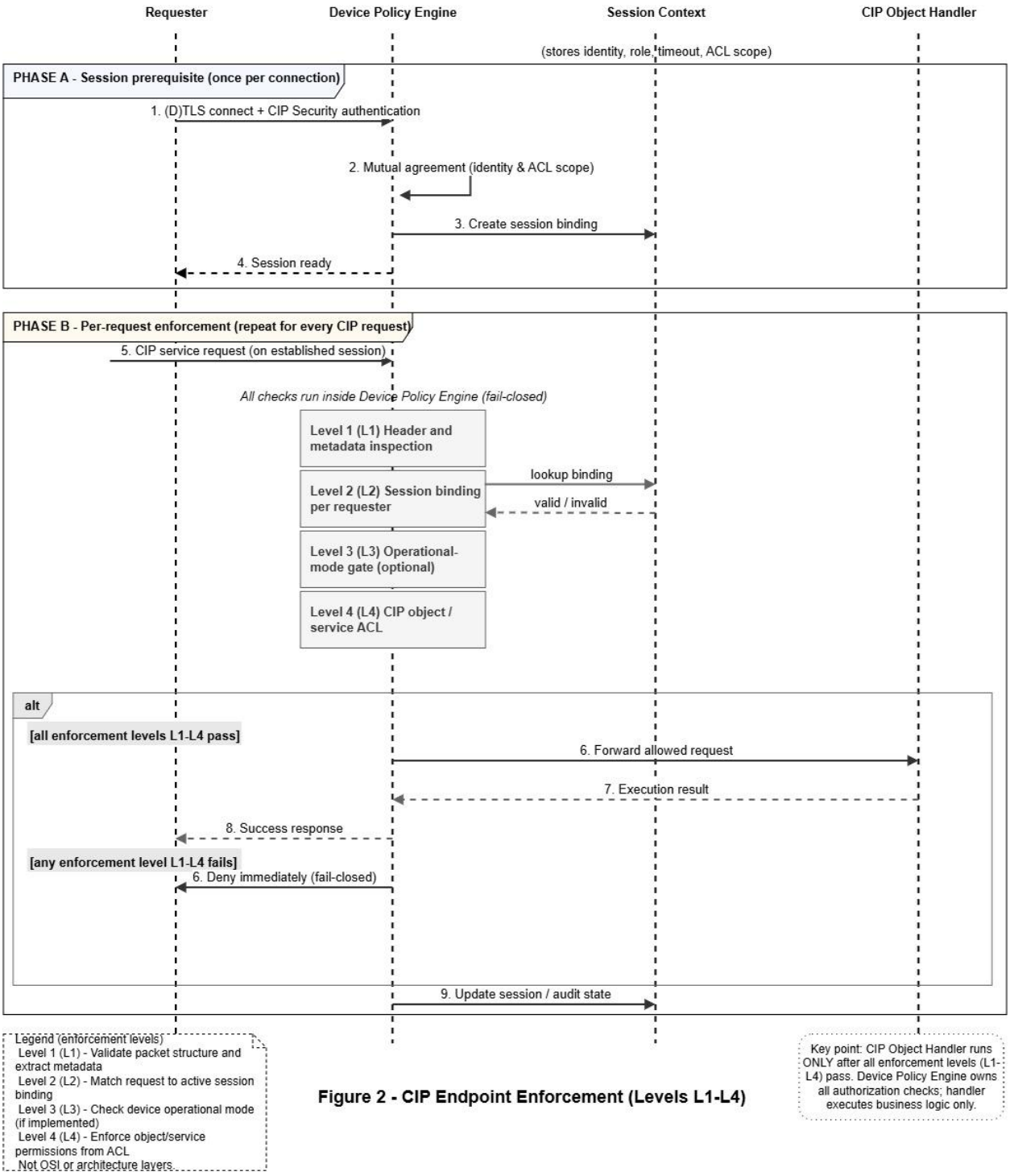


Figure 2 - CIP Endpoint Enforcement (Levels L1-L4)

7.7. Proposed Solution Fills the Gap Beyond Current CIP Security

Purpose: This subsection summarizes how the proposed pipeline extends existing CIP Security profiles without redefining them. Each item below maps one principle from the earlier design list to the concrete endpoint behavior introduced by this paper.

7.7.1. Mutual agreement on IP object access:

User Authentication Profile provides RBAC in existing method.
Proposed method adds device-requester bilateral agreement on specific CIP object permissions at session setup.

7.7.2. Header-first inspection pipeline:

Device-Based Firewall (existing CIP Security profile) filters L3/L4 traffic.
Proposed method adds structured metadata/header inspection as post-(D)TLS application policy, integrated with session binding, mode gates, and object ACLs

7.7.3. Session Bindings:

CIP Security already provides (D)TLS session keys and record-layer integrity or confidentiality.

7.7.4. Proposed method adds binds requester identity or role to the established security session and validates each request against that binding before mode and ACL checks. Inline prevention vs IDS:

CIP Security prevents unauthorized devices via authentication in existing methods.
Operational-mode gate:
CIP Security Object handles commissioning state in current method.
Proposed method extends to different operational modes (RUN/CONFIG) for command validation and explicitly blocks malicious payload processing.

7.8. Differentiation from Existing Approaches

'Prior academic work' refers to published ICS attack research (e.g., Urbina et al. [1]) that demonstrates compromise of fieldbus communications without device-level command validation. It does not refer to ODVA CIP Security.
What is new in this paper: Not individual mechanisms in isolation, and not replacement of CIP Security transport, DTLS session protection, or certificate handling already defined by Volume 8. The contribution is the explicit composition order on the CIP endpoint after (D)TLS, the fail-closed inline decision before the CIP Object Handler, and the practical binding of authenticated identity to session, mode, and object-level authorization policy.

Approach	Layer	Prevents Processing?	Command/Object Validation?
Firewall / Segmentation	Network	Partial	No
VPN / TLS wrapper	Transport	No (secures channel)	No
IDS [2]	Network/Device	No (alerts only)	Limited
Prior work [1]	Varies	Partial	Shows missing device auth, integrity, and command validation at fieldbus layer
Proposed method	Device + Network	Yes (inline reject)	Yes (multi-layer)

7.9. Supporting Operational Recommendations (Not Claimed as Novel)

These recommendations support secure deployment of the endpoint pipeline but are not presented as research contributions. They are included because practical endpoint authorization still depends on sound certificate lifecycle, key protection, and DTLS session management in real OT systems.

7.9.1. Certificate Revocation & Compromised Device Handling

- Define a revocation policy for compromised OT devices.
- Enforce hard-fail on revoked certificates for safety-critical connections.
- Automate certificate rotation and trust anchor updates when a device is decommissioned or breached.

7.9.2. TLS Configuration Hardening

- Prefer TLS 1.2
- Use AES-256-GCM or ChaCha20 for higher security

7.9.3.Private Key Protection

- Store private keys in a TPM, HSM, or secure element especially on controllers.
- Use key wrapping during provisioning.
- Implement secure boot mechanism, so that only authenticated firmware can use the device key.

7.9.4. DTLS-Specific Protections

- Set appropriate session timeouts and rekey intervals for long-running I/O connections.
- Handle packet loss and reordering without weakening security (DTLS retransmission policy).

7.9.5.Reduce Operational Burden Beyond PKI

PKI solves the key-distribution problem, but operations still can be poor.

- Automate certificate lifecycle (issue, renew, revoke) via an OT-aware gateway.
- Use short-lived certificates with automated renewal to limit exposure window time.

7.10. Alignment with ODVA CIP Security (Volume 8)

The mapping below shows where existing CIP Security objects and profiles stop and where this paper adds endpoint composition for session binding, mode-aware checks, and CIP object or service authorization. The proposed solution aligns with ODVA CIP Security as defined in Volume 8 of the CIP Networks Library and related ODVA training material. The paper does not replace CIP Security profiles. Rather it composes authenticated transport and existing profile concepts with endpoint policy checks for session binding, operational mode, and CIP object or service authorization.

Proposed Mechanism	ODVA CIP Security Equivalent	Volume 8 / Training Reference
Mutual authentication and agreement setup	Device Authentication via X.509 certificates or PSK. TLS/DTLS mutual authentication (Certificate Request/Verify)	EtherNet/IP Confidentiality Profile; TLS handshake (Fig. 3-2.4, Vol. 8 Ed. 1.17, ~p. 50); Digital certificate (Glossary, Ed. 1.17, ~p. 21)
Metadata/header inspection	Proposed pipeline Level 1 / L1 (post-(D)TLS application policy; not the Device-Based Firewall profile)	Endpoint header/metadata inspection composed with session binding, mode gates, and ACL
Session binding per requester	TLS/DTLS session management; session keys via PRF master secret	TLS handshake key derivation (§3-2, Vol. 8 Ed. 1.17, ~pp. 50–55); CIP-required cipher suites (§3-2.3, Ed. 1.17)
Packet integrity enforcement	HMAC for data authenticity; TLS/DTLS record layer encryption & authentication (AES)	TLS Record Protocol—integrity and optional confidentiality (§3-3, Vol. 8 Ed. 1.17, ~pp. 55–65); HMAC/AEAD per negotiated cipher suite
Operational state validation	CIP Security Object — exclusive commissioning access	CIP Security Object (Ed. 1.17, ~p. 108): commissioning exclusivity, profile state
CIP object or service authorization	User Authentication RBAC and Authorization Profile direction	Volume 8 user authorization direction; endpoint policy composition in this paper

7.11. CIP Security Objects Mapping

L4 CIP object or service authorization is implemented as endpoint application policy after CIP Security transport and User Authentication context are established; it is not a replacement for the Authorization Profile specification itself.

7.11.1. CIP Security Object:

- Maps to operational state validation and commissioning exclusivity.
- Ensures single-client exclusive access during security configuration. It provides profile or state query attributes.

7.11.2. EtherNet/IP Security Object:

- Maps to DTLS or CIP Security transport configuration—not to a proprietary per-packet application key.
- Configures TLS/DTLS settings: certificates vs PSK, trusted CAs, client certificate verification, expiration checks.

7.11.3. Certificate Management Object:

- Maps to authentication trust chain.
- Manages X.509 device certificates, CA certificates, trusted authorities
- Maps to X.509 trust chain management, certificate enrollment, and revocation handling. Does not perform network traffic filtering.
- Ingress/Egress (L3/L4) → (D)TLS via EtherNet/IP Security Object → session binding and CIP header checks → operational-mode gate *(application-layer extension; defense-in-depth; not universally specified by CIP Security)* → role/ACL object access (toward Authorization Profile).
-

7.12. CIP Security Profiles Relevant to Proposed Solution

CIP Security Profile	What It Addresses	Proposed Solution Extension
EtherNet/IP Confidentiality Profile	Unauthorized devices, MITM attacks	Binds each CIP request to the authenticated transport/session context and then apply application-layer policy
Mutual authentication & session binding	CIP Security DTLS	Provides the authenticated basis for per-request session binding before object-level authorization
Device based firewall	Simple traffic filter at device	Device-Based Firewall Profile (§8-2.5, Vol. 8 Ed. 1.17, ~p. 232); network-layer traffic filtering at the device; complements but does not replace L1 header inspection
Authorization Profile	Highly configurable user authorization	Application-layer endpoint policy aligned with the Authorization Profile direction which is complementary to User Authentication RBAC

7.12.1. Mechanisms Used

Method	Description
X.509 Certificates	Signed by a trusted Certificate Authority (CA) which proves device or client identity
Pre-Shared Keys (PSK)	Secret key configured on both sides before communication
User Authentication	Login with username/password, OpenID Connect

TLS/DTLS Handshake	Industry-standard secure transport (used in CIP Security)

7.12.2. Agreement conditions

Agreement Item	Description
Identity	Who is the requestor and who is the device
Trust Domain	Which CA or PSK is valid
Security Mode	Encrypted vs integrity-only, TLS vs plain
Session Context	Session ID, assigned roles, allowed scope. Here Session ID means the User Authentication Session ID
Protocol Version	Compatible security profile and capabilities

7.12.3. Session Record Contents

Field	Purpose
Session ID	Unique identifier for this connection
Requester ID / Type	Who connected (PLC, HMI, Engineer, Operator)
Assigned Role(s)	Permissions granted (Viewer, Operator, Engineer, Admin)
Unique Key Mechanism	Crypted key for this session
Creation Timestamp	When session started
Last Activity Timestamp	For timeout detection
Operational Context	Current allowed operations for this session
Authentication State	Fully authenticated / pending / expired

7.12.4. CIP Security Standard Roles

Role	Typical Access
Administrator	Full access; only role that can configure user auth
Engineer	Configuration, parameters, diagnostics
Operator	Operational control, limited config
Viewer	Read-only
Auditor	Audit logs

State	Description	Typical Allowed Operations
Running	Normal production operation	Read I/O, monitor, limited diagnostics
Idle	Powered on but not in production	Read config, some parameter changes
Configuration	Engineering or configuration mode	Parameter writing, network setup
Locked	Parameter lock active	Read-only; no writes
Safe / Fault	Safety or fault condition	Diagnostics only; no control commands
Firmware Update	Update in progress	No process I/O; update-related only
Program	Read only	Download or configuration allowed
Remote	Policy-defined	Remote stop/start if authorized

Fault	Read status only	Recovery actions only

7.12.5. Scenarios

Device State	Request	Result
Running	Read process value	Allowed
Running	Change IP address	Rejected (config not allowed while running)
Configuration	Set parameter	Allowed
Locked	Set any parameter	Rejected (device locked)
Safe or Fault	Send control command	Rejected (safety state)

7.13. IP Object Access Permission and Rejection

Scenario	Typical Response
Unauthenticated access to protected object	Connection refused or access denied
Set on Get-only attribute	Insufficient Privilege
Wrong role (e.g., Operator sets IP)	Access denied
Object/service not supported	Object/Service not supported
Expired session	Session invalid; reconnect required

8. Appendix:

8.1. Interoperability Note — Secure vs Non-Secure Mix

Mixed networks: secure-to-secure uses CIP Security DTLS and full post DTLS policy.
A non-secure peer if allowed is constrained by Ingress or Egress and device-based firewall and must not receive the same privileged CIP write config rights as a secure session.

8.2. Threat Coverage Mapping (illustrative vs CIP Security)

Representative threats vs pipeline layers (aligned to Volume 8):

Threat (representative)	Primary CIP Security / Vol 8 control	Proposed pipeline addition
Unauthorized device / spoofed endpoint	(D)TLS mutual authentication	Session binding of identity/role
MITM / tampering on wire	Confidentiality Profile record integrity	Fail-closed if binding invalid
L3/L4 unwanted traffic	Device-Based Firewall or Ingress–Egress	Keep as L3/L4 only
Unauthorized CIP object/service	User Authentication toward Authorization Profile	Mutual agreement and ACL role
Config/control while Running	Limited commissioning exclusivity	Operational-mode gates
Malicious command before alert	IDS Detect	Inline prevention before CIP handler
Secure or non-secure downgrade abuse	Profile and policy configuration	Deny privileged objects on non-secure paths

Deployment and Feasibility Considerations

This section examines whether the proposed endpoint policy pipeline is practical to deploy on industrial devices and proportionate to the authorization gap it addresses. The discussion covers where enforcement runs impact on resource-constrained embedded devices, performance and storage, the relationship between authentication and authorization and when a multi-stage pipeline is justified versus a simpler single-level design.

Deployment model: The device-based firewall (an existing CIP Security profile operating at OSI Layer 3/4) may filter network traffic, but the proposed pipeline stages L1–L4 application-policy checks execute entirely on the endpoint device after CIP Security transport is established. This is a device-resident enforcement model rather than a network-appliance-only model.

For resource-constrained devices, feasibility depends on policy representation and reuse of existing state information. The practical design goal is not to evaluate authorization four unrelated times, but to apply a fail-closed sequence of checks that can cancel invalid requests early while reusing common policy context. Compared with a single-level authorization design, the proposed composition may add code structure and state management overhead. Its value is strongest where authenticated requesters need distinct object-level permissions and operational-mode limits.

Performance and storage: Implementers should budget for one session-context record per active requester and evaluation that stops at the first failed stage. The design intent is not four independent authorization engines, but one policy engine applying sequential checks with shared configuration and early rejection of invalid requests.

Authorization versus authentication: CIP Security User Authentication and the Authorization Profile direction already separate identity establishment from configurable authorization, consistent with Volume 8 User Authentication and RBAC material. This paper describes a practical endpoint composition that uses authenticated identity as input to local authorization decisions; it does not redefine that separation.

Cost-benefit view: The added complexity is justified primarily for authenticated endpoints that require distinct object-level permissions, session-scoped authorization, or mode-aware command control. The proposed pipeline is most useful where policy composition and fail-closed consistency matter.

Performance and storage trade-offs are discussed qualitatively in this section; quantitative benchmarks are not provided and may be requested for specific product implementations.

8.3. Abbreviations and Acronyms

Abbreviation	Definition
ACL	Access Control List
AEAD	Authenticated Encryption with Associated Data
CA	Certificate Authority
CIP	Common Industrial Protocol
CPF	Common Packet Format (EtherNet/IP)
CRA	Cyber Resilience Act (European Union)
CRL	Certificate Revocation List
DLR	Device Level Ring
DPI	Deep Packet Inspection
DTLS	Datagram Transport Layer Security
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
EE	End Entity (certificate)
ENIP	EtherNet/IP
EST	Enrollment over Secure Transport
GCM	Galois/Counter Mode
HMAC	Keyed-Hash Message Authentication Code
IACS	Industrial Automation and Control Systems
ICS	Industrial Control System
IDS	Intrusion Detection System
IEC	International Electrotechnical Commission
IP	Internet Protocol; also Industrial Protocol in CIP object context
JWT	JSON Web Token
MAC	Message Authentication Code
MITM	Man-in-the-Middle (attack)

NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology
OCSP	Online Certificate Status Protocol
ODVA	Open DeviceNet Vendor Association
OT	Operational Technology
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
PLC	Programmable Logic Controller
PRF	Pseudorandom Function
PSK	Pre-Shared Key
RBAC	Role-Based Access Control
SCEP	Simple Certificate Enrollment Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TPM	Trusted Platform Module
UDP	User Datagram Protocol
UCMM	Unconnected Message Manager
X.509	ITU-T standard for public key infrastructure certificates

8.4. Existing Security Mechanism Details:

8.4.1. ODVA CIP Security
Reference: ODVA, CIP Security / Volume 8 of the CIP Networks Specification. Public overview:
https://www.odva.org/technology-standards/distinct-cip-services/cip-security/

8.4.2. ODVA 2014 - Securing EtherNet/IP with Deep Packet Inspection
Reference: Byres, Schweigert, Thomas, Securing EtherNet/IP with DPI, ODVA Conference 2014.

8.4.3. EtherNet/IP Developer and CIP Object Model Guidance
Reference: ODVA EtherNet/IP Developers Guide and CIP family publications.

8.4.4. Urbina et al. - Attacking Fieldbus Communications in ICS: Applications to the SWaT Testbed
Reference: David Urbina, Jairo Giraldo, Nils Ole Tippenhauer, Alvaro Cardenas, SG-CRC 2016. DOI: 10.3233/978-1-61499-617-0-75

8.4.5. Yu, Wang, and Song - A Two Stage Intrusion Detection System for Industrial Control Networks Based on Ethernet/IP
Reference: Wenbin Yu, Yiyin Wang, Lei Song, Electronics 8(12):1545, 2019.
URL: https://www.mdpi.com/2079-9292/8/12/1545

9. References

[1]David URBINA a,1, Jairo GIRALDOa, Nils Ole TIPPENHAUERb, and Alvaro CARDENASa, Attacking Fieldbus Communications in ICS: Applications to the SWaT Testbed)
[2] Wenbin Yu,Yiyin Wang,Lei Song, A Two Stage Intrusion Detection System for Industrial Control Networks Based on Ethernet/IP
[3] ODVA, The CIP Networks Library, Volume 8, CIP Security, Ann Arbor: ODVA, Inc., 2015-2026.
[4] ODVA, CIP Security / Volume 8 of the CIP Networks Specification. Public overview:
https://www.odva.org/technology-standards/distinct-cip-services/cip-security/

****

The ideas, opinions, and recommendations expressed herein are intended to describe concepts of the author(s) for the possible use of ODVA technologies and do not reflect the ideas, opinions, and recommendation of ODVA per se. Because ODVA technologies may be applied in many diverse situations and in conjunction with products and systems from multiple vendors, the reader and those responsible for specifying ODVA networks must determine for themselves the suitability and the suitability of ideas, opinions, and recommendations expressed herein for intended use. Copyright ©2026 ODVA, Inc. All rights reserved. For permission to reproduce excerpts of this material, with appropriate attribution to the author(s), please contact ODVA on: TEL +1 734-975-8840 FAX +1 734-922-0027 EMAIL odva@odva.org WEB www.odva.org. CIP, Common Industrial Protocol, CIP Energy, CIP Motion, CIP Safety, CIP Sync, CIP Security, CompoNet, ControlNet, DeviceNet, and EtherNet/IP are trademarks of ODVA, Inc. All other trademarks are property of their respective owners.