Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 20, 2026

V. Joseph
E. Hamadeh
C. Zhang

February 20, 2026

Certificate Expectation Assertions (CEA)
draft-joseph-cea-00

Abstract

This document specifies Certificate Expectation Assertions (CEA), a DNS-based mechanism enabling domain owners to publish expected certificate authority identities for their domains. Clients can compare observed TLS certificates against these published expectations to identify potential unauthorized interception.

CEA uses DNSSEC-signed DNS TXT records, supports multiple certificate authorities, and employs fail-open transparency rather than blocking connections. This approach differs from HPKP's fail-secure model and provides an optional mechanism for interception detection.

This document is published as Informational to document the CEA protocol for experimental deployment. It does not modify TLS or PKI infrastructure and is not an IETF standard. This is an experimental specification that has not been endorsed by any IETF Working Group and does not represent IETF consensus.

Note: This document does not modify the TLS protocol (RFC 8446), certificate validation procedures (RFC 5280), or PKI infrastructure. CEA is an optional client-side transparency overlay that provides additional information to users. Implementations that do not support CEA continue to function normally with no changes to TLS behavior.

Status of This Memo

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

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

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

This Internet-Draft will expire on August 20, 2026.

Copyright Notice

to this document.

Table of Contents

1. Introduction

TLS [RFC8446] provides confidentiality and integrity for Internet communications. TLS interception by intermediaries creates transparency challenges. Existing mechanisms have limitations: HPKP [RFC7469] deprecated due to operational brittleness, Certificate Transparency [RFC6962] provides post-issuance auditing but not real-time detection.

This document specifies Certificate Expectation Assertions (CEA), a DNS-based mechanism enabling domains to publish expected CA SPKI hashes and clients to identify mismatches, providing transparency signals that MAY indicate interception.

CEA is complementary to CT [RFC6962] and DANE [RFC6698]. CEA does not replace PKIX validation (RFC 5280) or modify TLS (RFC 8446). CEA operates as an application-layer overlay AFTER standard TLS validation succeeds. All standard TLS validation (chain verification, hostname matching, expiration) continues unchanged.

CEA does not define a new authentication mechanism, does not introduce new trust anchors, and does not alter TLS handshake processing; it provides an advisory transparency signal evaluated after successful PKIX validation.

This specification is submitted as an experimental Independent Stream document. It does not claim to be the optimal solution for TLS transparency and welcomes feedback from the IETF community. Implementers should carefully evaluate whether CEA fits their specific deployment requirements.

1.1. Requirements Language

Key words per RFC 2119 [RFC2119]. Since this is Informational, normative language applies only to implementations that elect to support CEA.

1.2. Terminology

CA: Entity that issues certificates [RFC5280].
CEA Record: DNS TXT record with certificate expectations.
Interception: Intermediary terminates TLS, inspects plaintext.
SPKI: DER-encoded public key from CA certificate [RFC5280].
Expected Issuer: CA in CEA record authorized to issue certificates.
Observed Issuer: CA in certificate presented during TLS handshake.

1.3. Key Design Features

1. SPKI-based expectations via DNS
2. Multi-path consensus (resolver diversity, 75% threshold)
3. Semantic categories (Financial, Healthcare) for compliance
4. Tiered confidence levels (DNSSEC, DoT/DoH, multi-path, TOFU)
5. Fail-open transparency (warnings, not blocking)

1.4. Scope and Non-Scope

CEA is NOT: a TLS protocol modification (RFC 8446), a change to certificate validation (RFC 5280), a PKI replacement, or mandatory.

Clients that implement CEA are RECOMMENDED to: continue standard TLS validation per RFC 5280, avoid rejecting connections based solely on CEA failures, allow users to proceed despite mismatches.

1.4.1. Explicit Non-Goals

CEA does NOT protect against: state-level DNS infrastructure compromise (attacker controls authoritative DNS and DNSSEC), CA compromise listed in CEA record, client-side compromise, fail-open bypass (attacker blocks all verification paths).

Mitigation for state-level DNS: Multi-path consensus with geographically diverse resolvers provides partial protection.

1.4.2. Relationship to IETF Working Groups

This document does not claim consensus from any IETF Working Group. CEA is an experimental mechanism submitted through the Independent Submission stream (RFC 8729). This specification does not represent IETF consensus and does not modify or supersede any IETF standards-track work. Implementers should be aware that CEA has not been reviewed or endorsed by the TLS Working Group, DNSOP Working Group, or any other IETF body.

1.5. Relationship to Existing Work

CEA builds upon lessons learned from prior certificate validation mechanisms. This section clarifies CEA’s relationship to existing IETF specifications and explains design choices informed by deployment experience.

1.5.1. Relationship to DANE (DNS-Based Authentication of Named Entities)

DANE (RFC 6698, RFC 7671) is the most similar existing mechanism, using DNS to publish certificate expectations. Key differences:

Design Philosophy:

- DANE: Fail-secure enforcement (MUST reject mismatches)
- CEA: Fail-open transparency (MAY warn users)

DNSSEC Dependency:

- DANE: DNSSEC required for all deployments
- CEA: Tiered model works without DNSSEC (Levels 1-3)

Deployment Status for HTTPS (2026):

- DANE HTTPS: Limited browser adoption despite RFC 6698 publication in 2012
- DANE has seen deployment primarily in the email/SMTP domain
- CEA: Experimental proposal informed by DANE deployment experience

DANE Deployment Challenges for HTTPS:

- Fail-secure enforcement creates operational challenges for enterprise TLS inspection scenarios
- DNSSEC deployment remains limited in some regions
- Operational complexity similar to challenges faced by HPKP

CEA Design Approach:

- Transparency-focused model accommodates enterprise inspection scenarios
- Tiered validation model functions without DNSSEC dependency
- Fail-open behavior reduces operational risk

Relationship: CEA does not replace DANE. Both serve complementary roles for different use cases (DANE for mail servers with fail-secure needs, CEA for HTTPS with transparency focus).

Technical Comparison Table:

Aspect	CEA	DANE
DNSSEC Required	No (Levels 1-3) Optional (Level 4)	Yes (mandatory)
PKIX Replacement	No (overlay only)	Optional (DANE-TA mode)
Deployment Scope	Application policy layer	DNS + TLS infrastructure
Trust Anchor Change	No (uses OS/browser CA store)	Potentially yes (DANE-TA)
Failure Mode	Fail-open (warnings)	Fail-secure (blocks)
Enterprise TLS Inspection	Compatible (warns)	Incompatible (blocks)
Browser Support	Experimental	Limited browser adoption
Primary Use Case	HTTPS transparency	SMTP/Mail authentication
TLS Validation Semantics	None (advisory overlay only)	May augment or replace PKIX validation

DANE is appropriate for environments where fail-secure enforcement is acceptable (e.g., mail servers). CEA is designed for environments requiring transparency without blocking (e.g., enterprise web browsing).

Architectural Distinction:

CEA and DANE have complementary architectural approaches. DANE can modify TLS certificate validation (replacing or augmenting PKIX trust), while CEA operates as a transparency layer AFTER standard TLS validation succeeds. CEA provides transparency signals without blocking connections. This architectural difference allows both mechanisms to serve complementary roles in different deployment contexts.

1.5.2. Relationship to HPKP (HTTP Public Key Pinning)

HPKP (RFC 7469, deprecated 2018) provides the historical lesson that informed CEA’s design.

What HPKP Got Right:

- SPKI-based pinning (CEA uses same approach)
- Recognized need for CA-level expectations

Why HPKP Failed:

- Operational Brittleness: Misconfigurations locked users out permanently
- Chicken-and-Egg: Delivered over HTTPS, couldn't validate first connection
- No Recovery: Once pinned incorrectly, domain became inaccessible
- Weaponization: Attackers could pin to malicious certificates

CEA Design Improvements:

- DNS Delivery: Out-of-band, can validate first connection
- Fail-Open: Misconfigurations don't break sites
- TTL-Based Expiry: Errors automatically heal after TTL
- Multiple CAs: Supports smooth CA rotation
- Transparency Not Prevention: Users retain agency

Specific HPKP Problem CEA Solves:

HPKP Scenario: Domain owner sets pin for CA-A, later switches to CA-B, forgets to update pin. Result: All users locked out until max-age expires (months).

CEA Equivalent: Domain owner publishes CEA for CA-A, switches to CA-B. Result: Users see warning but can proceed. CEA record updates via DNS TTL (hours), or domain owner can pre-publish both CAs.

Historical Context: Chrome removed HPKP support in 2018 citing "limited legitimate usage and high risk of misconfiguration." CEA addresses these exact concerns through fail-open design and DNS-based updates.

1.5.3. Relationship to Certificate Transparency (CT)

Certificate Transparency (RFC 6962) and CEA are complementary, not competitive.

CT's Role:

- Detects unauthorized issuance after the fact
- Monitors: "Did a CA issue a certificate it shouldn't have?"
- Timeline: Post-issuance detection (hours to days)

CEA's Role:

- Detects unauthorized usage at connection time
- Monitors: "Is this certificate being used as expected?"
- Timeline: Real-time detection during TLS handshake

Complementary Defense:

- CT: Detects unauthorized CA issuance (post-hoc monitoring)
- CEA: Detects when legitimate CAs are used for interception (real-time transparency)

Example Scenario:

1. Enterprise installs corporate CA in employee devices
2. CT logs show no unauthorized issuance (CA is legitimate)
3. CEA identifies connection uses corporate CA instead of expected public CA (SPKI mismatch)
4. User informed of potential interception

CT Cannot Detect:

- Enterprise proxy using locally-trusted CA
- State-level interception with compelled CA
- TLS interception using legitimate intermediate CAs

CEA Cannot Detect:

- Compromised CA issuing fraudulent certificates (CT's job)
- Zero-day CA key compromise before CT logging

Layered Security: Organizations should deploy both CT monitoring AND CEA validation for comprehensive protection.

1.5.4. Relationship Summary

CEA occupies a unique position in the certificate security ecosystem:

Mechanism	Enforcement	Scope	Browser Support
DANE (RFC 6698)	Fail-secure	HTTPS	Not deployed
HPKP (RFC 7469)	Fail-secure	HTTPS	Deprecated 2018
CT (RFC 6962)	Observability	CA	Universal
CAA (RFC 8659)	Enforcement	CA	N/A (CA-side)
CEA (This doc)	Transparency	HTTPS	Experimental

Design Principle: CEA is transparency-focused and fail-open, differing from the fail-secure approaches of HPKP and DANE. CEA complements CT's post-issuance monitoring by detecting active interception in real-time.

Key Feature: CEA provides user awareness through transparency signals rather than connection blocking, allowing deployment without the operational risks associated with fail-secure mechanisms.

2. Motivation and Problem Statement

2.1. TLS Interception Landscape

TLS interception modifies the end-to-end security model of encrypted communications. Interception occurs in various contexts:

Malicious Activity: Attackers use interception techniques to eavesdrop on communications, steal credentials, inject malware, or perform man-in-the-middle attacks.

Surveillance: Government entities may intercept TLS traffic for intelligence gathering purposes, raising privacy concerns.

Enterprise Monitoring: Organizations deploy SSL/TLS inspection appliances for security analysis, though these systems create transparency and compliance considerations.

Content Filtering: Educational institutions and public networks use interception for policy enforcement, often without explicit user notification.

The common characteristic of all interception scenarios is that the client application receives a certificate issued by an entity other than the domain owner's chosen certificate authority. When the intercepting CA's root certificate is trusted by the client (either through the system trust store or manual installation), the client has no mechanism to detect that interception has occurred.

This creates fundamental security and privacy problems:

Privacy Violation (PRIMARY CONCERN): Users are unaware when their supposedly private communications are being read by third parties. This violates the reasonable expectation of privacy that TLS encryption is intended to provide. Users cannot give informed consent when interception is invisible.

Loss of Expected End-to-End Security: When TLS interception occurs, the security model changes from end-to-end encryption to two separate TLS sessions (client-to-proxy and proxy-to-server). While this remains valid TLS behavior when the proxy CA is trusted, users may have a reasonable expectation of direct end-to-end encryption that is not met. This occurs without user awareness or explicit consent in many deployments.

Attack Surface Expansion: Interception increases the attack surface by adding additional points where plaintext is exposed. Inspection appliances and their credential stores become high-value targets for attackers.

Compliance Risk: Organizations performing inspection may inadvertently violate regulations (e.g., GDPR, HIPAA, PCI-DSS) by intercepting sensitive data flows. Current systems lack mechanisms to prove non-interception of regulated traffic.

2.2. Limitations of Existing Solutions

2.2.1. HTTP Public Key Pinning (HPKP)

HPKP [RFC7469] allowed websites to instruct browsers to only accept specific public keys for a given domain. While conceptually sound, HPKP suffered from critical operational issues:

- Rigidity: Websites could not easily change certificate providers without risking widespread connection failures.
- No Recovery Mechanism: Misconfigured pins could render websites inaccessible for extended periods.
- Cache Persistence: Pins were cached in browsers for extended periods with no ability for immediate updates.
- Deployment Complexity: Backup pins and careful operational procedures were required but often not implemented correctly.

These limitations led to HPKP’s deprecation in 2018.

2.2.2. Certificate Transparency (CT)

Certificate Transparency (CT) [RFC6962] provides append-only public logs of issued certificates. While valuable for detecting unauthorized certificate issuance, CT does not address interception detection:

- CT logs record certificate issuance events, not certificate usage
- CT does not provide real-time validation during TLS handshakes

- CT does not inform users when interception is occurring

2.2.3. Certification Authority Authorization (CAA) Records

Certification Authority Authorization (CAA) [RFC6844] enables domain owners to restrict which CAs may issue certificates. However:

- CAA restricts issuance authorization, not certificate usage
- CAA is checked during certificate issuance, not during client connections

- CAA does not detect when a validly issued certificate is being used by an intercepting proxy

2.2.4. Application-Level Pinning

Some applications implement certificate or public key pinning internally. This approach:

- Requires per-application implementation
- Does not benefit web browsers or other TLS clients
- Suffers from the same operational challenges as HPKP

2.3. Design Requirements

Based on the limitations of existing mechanisms, CEA addresses the following design requirements:

R1: MUST provide a mechanism for domain owners to publish expected certificate authorities

R2: MUST enable clients to validate observed certificates against published expectations

R3: MUST support multiple authorized certificate authorities to enable CA rotation without service disruption

R4: MUST provide a mechanism for rapid updates to published expectations

R5: MUST be cryptographically protected against tampering

R6: SHOULD provide user transparency when interception is detected rather than blocking connections

R7: SHOULD be deployable without requiring changes to certificate authorities

R8: SHOULD leverage existing infrastructure where possible

R9: MAY provide semantic categorization to enable policy automation

3. CEA Mechanism Overview

3.1. Publishing Certificate Expectations

Domain owners publish certificate expectations by creating DNS TXT resource records under the "_cea" subdomain. For example, to publish expectations for "example.com", the domain owner creates a record at "_cea.example.com".

The CEA record contains:

- A version identifier
- A list of authorized certificate authority names
- Optional semantic trust categories
- Optional cache duration parameters

The CEA record SHOULD be DNSSEC-signed to provide cryptographic assurance of authenticity and integrity. DNSSEC provides the highest security guarantee (confidence Level 4), but CEA deployment does NOT require universal DNSSEC adoption. Clients MAY use alternative verification methods (DoT/DoH, multi-path consensus, registry) for domains without DNSSEC, as specified in Section 4.5.

3.2. Client Validation Process

When a client establishes a TLS connection, it performs validation in two phases:

Phase 1 - Standard TLS Validation (REQUIRED):

1. Perform complete RFC 5280 certificate validation: chain verification, hostname matching, expiration, revocation checks
2. If RFC 5280 validation fails, reject connection per standard TLS procedures. CEA validation does NOT proceed.

Phase 2 - CEA Transparency Check (OPTIONAL, only after Phase 1 succeeds):

3. Query DNS for CEA record for the target domain
4. If available, validate DNSSEC signature chain
5. Extract expected CA SPKI hashes from CEA record
6. Compute SPKI hash of the issuing CA certificate (the certificate that directly signed the end-entity certificate, typically an intermediate CA)
7. If SPKI matches any expected hash: CEA check passes, no action
8. If SPKI does not match: CEA check fails, MAY display transparency warning. Connection remains established (fail-open).

Important: CEA is a post-validation transparency overlay. RFC 5280 validation is unchanged and takes precedence.

3.3. User Experience and Transparency

CEA is designed to provide transparency and user empowerment. The primary goal is to restore informed consent by making interception visible to users.

When a CEA validation failure occurs (indicating potential interception), the recommended behavior prioritizes user agency:

For Interactive Clients (e.g., Web Browsers):

- Display a clear warning that interception may be occurring
- Provide information about the expected cryptographic identity (SPKI hash) and the observed identity
- Explain the implications: "Your organization or network may be reading this connection"
- Present options with clear privacy implications:
  - Continue (user accepts interception for this session)
  - Always trust this network (user consents to monitoring)
  - Use alternative connection (VPN, cellular, etc.)
  - Do not connect (preserve privacy)
- Log the event for user’s own security awareness
- Recommended: Warnings SHOULD be user-facing. Implementations MAY allow enterprise policy control, but SHOULD ensure users have visibility into policy-suppressed warnings through logging or notification mechanisms where operationally feasible.

For Automated Clients (e.g., API Clients, IoT Devices):

- Log the validation failure
- Optionally alert administrators
- Proceed or abort based on local policy configuration

This approach balances security with operational flexibility, particularly in enterprise environments where legitimate SSL inspection may be employed.

4. CEA Record Specification

4.1. DNS Resource Record Format

CEA records are published as DNS TXT resource records [RFC1035] under the "_cea" subdomain label.

For domain "example.com", the CEA record is at "_cea.example.com". For subdomain "www.example.com", it is at "_cea.www.example.com".

4.2. CEA Record Syntax

The CEA record is a semicolon-separated string of tag-value pairs:

```
cea-record = "v=CEA1" *(";" tag "=" value)
```

Supported tags:
- v: Version (REQUIRED, must be "CEA1")
- pins: SPKI hashes (REQUIRED)
- cat: Categories (OPTIONAL)
- max_age: Cache lifetime in seconds (OPTIONAL, default: 86400)

Example:

```
v=CEA1;pins=sha256/X3pGTSOuJeEVw989IJ/cEtXUEmy52zs1TZQrU06KUKg=;cat=Financial
```

4.3. Field Definitions

4.3.1. Version (v)

REQUIRED. Must be "CEA1". Clients MUST ignore unknown versions.

4.3.2. Pins (pins)

REQUIRED. Comma-separated list of expected CA SPKI hashes.

Format: hash-algorithm/base64-encoded-hash

Supported algorithms:
- sha256 (REQUIRED)
- sha384 (RECOMMENDED)
- sha512 (OPTIONAL)

SPKI Calculation: Extract SubjectPublicKeyInfo from the issuing CA certificate, compute SHA-256 hash, Base64 encode [RFC4648].

Chain Level Clarification: In a typical PKI chain (Root CA -> Intermediate CA -> End-Entity), the "issuing CA" is the certificate that directly signed the end-entity certificate (typically the intermediate). Domain owners MAY pin the intermediate, the root, or both. Pinning the intermediate provides stronger binding; pinning the root provides operational flexibility during intermediate rotation.

Example: pins=sha256/X3pGTSOuJeEVw989IJ/cEtXUEmy52zs1TZQrU06KUKg=

4.3.3. Categories (cat)

OPTIONAL. Comma-separated semantic trust categories (see Section 6).

Example: cat=Financial,Healthcare

4.3.4. Maximum Age (max_age)

OPTIONAL. Cache lifetime in seconds. Default: 86400 (24 hours).

Example: max_age=3600

4.4. DNSSEC Requirements

CEA records SHOULD be signed with DNSSEC [RFC4033] when available. DNSSEC provides cryptographic authenticity and prevents cache

poisoning.

Clients MAY require DNSSEC for high-security deployments. When DNSSEC validation fails or is unavailable, clients SHOULD use alternative verification methods (Section 4.5).

4.5. Alternative Verification Methods

When DNSSEC is unavailable, clients can verify CEA records using:

4.5.1. Trust on First Use (TOFU) - Level 1

On first connection, client caches the observed CA SPKI. On subsequent connections, client compares presented certificate against cached SPKI. Warns on mismatch.

Limitation: Vulnerable to MITM on first connection.

4.5.2. DNS-over-TLS / DNS-over-HTTPS - Level 3.5

Query CEA records via encrypted DNS (DoT [RFC7858] or DoH [RFC8484]) to prevent passive DNS eavesdropping and active modification.

Recommended DoH resolvers: Cloudflare (1.1.1.1), Google (8.8.8.8), Quad9 (9.9.9.9).

4.5.3. Multi-Path Consensus - Level 3

Query CEA record from n >= 8 geographically diverse DNS resolvers. Accept result if t >= 75% of resolvers agree.

Provides increased resistance to localized DNS manipulation through resolver diversity. Assumes independent, honest resolver selection. Does NOT provide formal Byzantine fault tolerance due to: lack of resolver coordination, potential upstream correlation, and no protection against coordinated attacks. See Section 7.8.3 for limitations.

Example: Query 8 resolvers, require 6 to agree (75% threshold). Increases resistance to single-resolver compromise.

4.5.4. Public Registry Lookup - Level 2

Query CEA records from a public registry of verified CEA assertions (e.g., certificate transparency-style log).

Registry provides independent verification but introduces centralization.

5. Client Validation Algorithm

This section describes the validation algorithm for clients that implement CEA. Implementation is OPTIONAL.

5.1. CEA Record Discovery

Clients discover CEA records by querying "_cea.<domain>" TXT records.

Verification methods (in order of preference):
1. DNSSEC validation (Level 4) - highest confidence
2. DNS-over-HTTPS/TLS (Level 3.5) - encrypted DNS
3. Multi-path consensus (Level 3) - resolver diversity
4. Registry lookup (Level 2) - independent verification
5. TOFU (Level 1) - first-use caching

If no method succeeds, proceed without CEA validation (fail-open).

Parse the TXT record per Section 4.2. Cache the result using max_age or DNS TTL.

5.2. Certificate Validation

After obtaining the CEA record, validate the presented certificate:

1. Extract the issuing CA certificate (the certificate that directly signed the end-entity certificate, typically an intermediate)
2. Compute SHA-256 hash of the issuing CA's SubjectPublicKeyInfo
3. Compare against pins in CEA record
4. If any pin matches, validation succeeds
5. If no match, validation fails

SPKI is the DER-encoded ASN.1 structure from [RFC5280] Section 4.1.2.7. Hash includes SEQUENCE tag and length octets.

5.3. Error Handling

CEA validation results:

- PASS: Certificate SPKI matches a pin in CEA record. Proceed with connection.

- FAIL: CEA record exists but SPKI does not match. Warn user of potential interception.

- ERROR: CEA record unavailable (DNS timeout, parse error). Fail-open: proceed with connection, optionally inform user.

- NONE: Domain does not publish CEA record. Proceed with standard TLS validation.

5.3.1. User Warnings

On CEA validation failure (FAIL state), clients SHOULD present a warning that indicates:

- Unexpected certificate detected
- Possible TLS interception
- Option to proceed or abort connection
- Option to report the incident

Warning UX should differentiate CEA failures from standard TLS errors (invalid certificate, expired, hostname mismatch).

5.3.2. Fail-Open vs Fail-Closed

CEA implements fail-open behavior: if CEA record is unavailable (ERROR state), connection proceeds.

Rationale: Fail-closed (block connection on CEA unavailability) would enable DoS attacks by suppressing CEA records. Domains can enforce fail-closed via semantic categories (Section 6.2).

5.3.3. Network Memory

Clients SHOULD implement network memory: if a domain previously published a CEA record, subsequent absence (transition from PASS/FAIL to ERROR/NONE) triggers a warning.

Memory retention: 90 days recommended.

This mitigates downgrade attacks where attackers suppress CEA records after initial publication.

Operational Considerations: Network memory introduces stateful behavior similar to mechanisms like HPKP that experienced operational challenges. However, CEA's fail-open design significantly reduces brittleness risk: memory failures trigger warnings rather than blocking connections. Implementations SHOULD provide mechanisms to clear memory state if operational issues arise (e.g., legitimate domain configuration changes, CA migration failures).

5.4. Multi-Path Consensus Algorithm

Multi-path consensus queries n >= 8 diverse resolvers and accepts results with t >= 75% agreement.

5.4.1. Resolver Selection

Recommended diversity:
- Geographic: Different jurisdictions
- Organizational: Different operators
- Infrastructure: Different anycast networks

5.4.2. Consensus Threshold

Threshold t = 75% increases resistance to resolver compromise.

For n=8: requires 6/8 agreement, tolerates up to 2 compromised resolvers (assuming independence).
For n=12: requires 9/12 agreement, tolerates up to 3 compromised resolvers (assuming independence).

The 75% threshold is a deployment trade-off between availability and compromise resistance. Lower thresholds (e.g., 50%) provide insufficient resistance to partial resolver compromise, while higher thresholds (e.g., 90%) significantly reduce availability under benign resolver disagreement. The 75% value tolerates limited resolver compromise while maintaining practical connectivity. Implementations MAY choose stricter thresholds based on local policy.

5.4.3. Tie Handling

If no result achieves threshold (e.g., 50/50 split), treat as ERROR and fail-open.

6. Semantic Trust Categories

The optional "cat=" parameter allows domains to signal semantic trust categories for policy automation.

6.1. Category Taxonomy

Standard categories include: Financial, Healthcare, Government, Military, Education, E-commerce, Social-Media, Communication, Cloud-Provider, Enterprise, IoT, Critical-Infrastructure.

Implementations MAY define additional categories. See IANA Considerations (Section 11.3) for registry.

6.2. Policy Automation

Categories enable policy signaling for enterprise compliance, but require independent verification due to self-attestation.

Security Warning: Categories are self-attested by domain owners. Malicious domains can claim any category to bypass security controls. Enterprises MUST NOT trust categories without verification:

- Maintain verified allowlists of known Financial/Healthcare domains

- Cross-reference with business registrations or industry databases
- Monitor for category abuse and require manual approval for category-based exemptions
- Log all category claims for audit

Example implementation with verification:

```
IF categories CONTAINS "Financial" OR "Healthcare" THEN
  IF domain IN verified_financial_allowlist THEN
    DO NOT intercept (verified compliance requirement)
    LOG exemption for audit
  ELSE
    LOG "Unverified category claim: " + domain
    APPLY normal inspection policy
  END IF
END IF
```

Categories provide a compliance signal but MUST NOT be trusted without independent verification. Unverified category exemptions create security bypass opportunities.

7. Security Considerations

This section provides security analysis including attack classification, threat modeling, resolver diversity properties, and explicit limitations.

Note: CEA provides detection signals, not cryptographic proof of interception. SPKI mismatches and consensus failures indicate potential interception but do not constitute unforgeable proof. CEA enables user transparency and informed decision-making.

7.1. Attack Classification and Threat Model

CEA addresses a specific threat class: network-level TLS interception by MITM attackers using locally-trusted CAs. This includes enterprise proxies, government MITM, compromised network equipment, and malicious software that installs root certificates.

7.1.1. Attacks CEA Mitigates

- Enterprise MITM with Self-Signed CA: Detection when DNS is not suppressed and domain publishes CEA record excluding the enterprise CA

- Nation-State MITM (Network-Level): Detection when attacker cannot control all DNS resolution paths

- Rogue Certificate from Authorized CA: Partial detection based on pinning granularity

- Cache Poisoning DNS Injection: Detection via DNSSEC validation or multi-path consensus

7.1.2. Explicit Limitations (Out-of-Scope)

CEA does NOT protect against:

- State-level DNS control: Attacker controlling authoritative DNS and DNSSEC infrastructure can modify CEA records. Mitigation: Multi-path consensus with geographically diverse resolvers.

- Compromised CA listed in CEA: If attacker compromises a CA that domain authorizes in CEA record, CEA validation passes. This is working as designed.

- Client-side compromise: If attacker controls the client (malware, OS modification), CEA validation can be bypassed.

- Fail-open bypass: If attacker blocks all CEA verification paths (DNSSEC + DoT/DoH + multi-path + registry), client fails open to prevent DoS. This is intentional.

7.1.3. Resolver Diversity and Consensus Limitations

CEA multi-path consensus uses resolver diversity to increase resistance to localized DNS manipulation. This approach has specific properties and important limitations:

Property 1 (Transparency Under Network-Level Attack): If an attacker has on-path DNS control and MITM capabilities but cannot compromise DNSSEC infrastructure, CEA's multi-path consensus MAY provide transparency signals under specific conditions.

Property 2 (Fail-Open Limitation): If an attacker blocks all verification paths, CEA fails open. This is an acknowledged limitation required for internet-scale deployment.

Property 3 (Resolver Diversity): Multi-path consensus (e.g., querying 8 resolvers with 75% threshold) increases resistance to single-resolver compromise. However, this is NOT formal Byzantine fault tolerance because: (a) resolvers do not coordinate, (b) resolvers may share upstream providers, creating correlation, (c) resolver selection assumes honest choices, (d) no protection against coordinated resolver compromise.

Property 4 (Jurisdictional Diversity): Querying resolvers across jurisdictional boundaries MAY provide partial protection against single-nation DNS manipulation, subject to the limitations in Property 3.

These properties assume independent, honest resolver selection. Correlated resolver infrastructure or coordinated attacks reduce effectiveness.

7.2. DNSSEC Requirements

CEA records SHOULD be signed with DNSSEC when available:

- DNSSEC provides cryptographic authenticity for CEA records
- DNSSEC-signed CEA records prevent cache poisoning attacks
- Clients MAY require DNSSEC for high-security deployments

When DNSSEC is unavailable or validation fails, clients SHOULD fall back to multi-path consensus or registry lookup (Section 5.1).

7.3. DNS Poisoning and Cache Attacks

CEA is vulnerable to DNS cache poisoning attacks if:
1) DNSSEC is not deployed, AND
2) Multi-path consensus is disabled, AND
3) Registry fallback is not used

Mitigation: Enable at least two of {DNSSEC, multi-path, registry}.

7.4. Downgrade Attacks and Fail-Open Behavior

Attackers may attempt to suppress CEA records via:

- Active suppression: Return NXDOMAIN for _cea TXT queries
- DNS timeout: Block all DNS queries to force ERROR state
- DNSSEC stripping: Remove DNSSEC signatures to bypass validation

CEA employs network memory to detect suppression: once a domain has published a CEA record, clients remember this for 90 days. Subsequent absence triggers warnings. This mitigates downgrade attacks against domains that previously published CEA.

Fail-open behavior is INTENTIONAL: blocking all internet connectivity (fail-closed) enables DoS attacks and violates deployment constraints.

7.5. Privacy Considerations

CEA's privacy properties are compared to existing DNS-based trust mechanisms (CAA, TLSA, OCSP).

7.5.1. Privacy Characteristics

CEA introduces no new DNS query patterns beyond domain-scoped TXT lookups and therefore does not materially alter DNS visibility characteristics. DNS observers learn the domain being accessed (same as A/AAAA queries), with DoT/DoH encryption available.

Note: Multi-path consensus (Section 4.5.3) queries 8+ resolvers, expanding the set of observers who learn about the connection. This creates additional privacy exposure compared to single-resolver DNS. Clients using multi-path SHOULD use DoH/DoT to all resolvers when possible.

Standard single-resolver CEA queries do not introduce privacy leaks beyond existing DNS infrastructure.

7.5.2. Privacy Requirements for DoT/DoH

CEA clients SHOULD use DNS-over-TLS (DoT) or DNS-over-HTTPS (DoH) when available to prevent passive DNS eavesdropping. This is a recommendation, not a requirement, to maintain deployment flexibility.

Enterprise networks may block DoT/DoH; CEA gracefully degrades to standard DNS in these environments.

7.5.3. Network Tagging Concern

CEA queries may reveal that a client is interception-aware. However:

- DNS queries are observable regardless (A/AAAA queries reveal domains)
- DoT/DoH usage already signals privacy-conscious behavior
- CEA adoption reduces tagging risk over time (normalization)

7.6. Certificate Authority Compromise

If an attacker compromises a CA that is authorized in a domain's CEA record, CEA validation will pass (certificate is "expected"). This is not a vulnerability--it is working as designed.

Mitigation: Domains should minimize the number of authorized CAs in their CEA record (principle of least privilege).

7.7. Interaction with Encrypted ClientHello (ECH)

ECH encrypts the TLS SNI field to prevent passive observers from learning the target domain. CEA and ECH are complementary:

- ECH prevents passive eavesdropping of SNI
- CEA detects active MITM attacks

CEA validation occurs AFTER TLS handshake completion, using the certificate presented by the server. CEA does not validate ECH public

keys or configurations--this remains the responsibility of the TLS stack.

ECH downgrade attacks (attacker stripping ECH extension) are outside CEA's scope; however, if the downgrade results in certificate mismatch, CEA will detect it.

7.8. Additional Security Considerations

7.8.1. CEA Does Not Replace TLS Validation

CEA is an ADDITIONAL check performed after standard TLS validation. Clients MUST continue to validate:

- Certificate chain to trusted root CA
- Certificate validity period
- Hostname verification
- Revocation status (OCSP/CRLSets)

CEA augments TLS validation; it does not replace it.

CEA does not alter certificate validation success or failure as defined by RFC 5280. A certificate that passes RFC 5280 validation MUST be treated as valid regardless of CEA outcome. CEA results are advisory transparency signals and MUST NOT modify the TLS state machine.

7.8.2. Enterprise Warning Fatigue

Users in enterprise environments with legitimate TLS interception may experience warning fatigue if CEA warnings are too aggressive.

Mitigation: Network memory implementation (Section 7.2.1) learns stable enterprise interception patterns and reduces warning frequency for known-stable environments.

7.8.3. Multi-Path Resolver Trust and Limitations

Multi-path consensus assumes that >= 75% of queried resolvers are honest. This assumption has known limitations:

- Works well against: Enterprise proxies, local MITM, single-resolver compromise
- Weak against: Jurisdictional capture (state controls multiple resolvers), coordinated resolver collusion
- Fails when: Attacker controls >= 25% of selected resolvers AND blocks remaining queries

Recommended resolver diversity (helps but doesn't eliminate risk): Geographic diversity (different jurisdictions), organizational diversity (different operators), infrastructure diversity (different anycast networks).

Important Limitation: Multi-path consensus is a best-effort resolver diversity mechanism, not a cryptographic guarantee or formal Byzantine fault tolerance system. Effectiveness depends on resolver independence and honest selection. Users should understand these limitations when relying on Level 3 validation.

7.8.4. Downgrade: CEA Record Stripping

Attackers may return NXDOMAIN for _cea queries to suppress CEA validation. Mitigation: Network memory (Section 7.2.1) detects suppression for domains that previously published CEA.

Initial deployment (domain first publishes CEA): No protection against

suppression. However, this is no worse than current state (no CEA).

7.8.5. Active DNS Suppression

State-level attackers may actively suppress CEA TXT queries. Multi-path consensus provides partial mitigation by querying diverse resolvers across jurisdictional boundaries.

Limitation: If all resolvers are suppressed, CEA fails open (Section 8.4).

7.8.6. Semantic Categories Privacy

The "cat=" parameter may reveal domain's security posture. This is intentional: domains explicitly signal their policy to enable automated client responses (Section 6).

8. Operational Considerations

8.1. Certificate Authority Rotation

CEA supports smooth CA rotation by listing multiple CA SPKIs simultaneously.

Recommended procedure:
1. Expand CEA record to include both old and new CA SPKIs
2. Wait for DNS propagation (TTL + safety margin)
3. Obtain and deploy certificate from new CA
4. After verification, remove old CA SPKI from CEA record

Emergency migration: If immediate CA change is required, update CEA record first, then deploy new certificate. Some clients may experience warnings during propagation delay.

8.2. Key Rollover

DNSSEC key rollover follows RFC 6781 procedures. CEA records should use sufficiently long TTL (>= 3600 seconds) to avoid excessive re-querying during normal operation.

8.3. Monitoring and Incident Response

Domain operators should monitor for:
- CEA record availability (DNS query success rate)
- DNSSEC validation success rate
- Client-reported validation failures

Incident response: If unexpected validation failures occur, verify CEA record accuracy, check for certificate chain issues, and investigate potential DNS poisoning or unauthorized certificate issuance.

8.4. CDN and Multi-Certificate Scenarios

Domains using multiple CAs (e.g., CDN with different CAs per region) should list all CA SPKIs in the CEA record. Example:

```
pins=sha256/CA1_SPKI,sha256/CA2_SPKI,sha256/CA3_SPKI
```

Important Limitation (Shared CA Risk): CEA validates the CA SPKI, not the end-entity certificate. In shared hosting/CDN scenarios, an attacker with a valid certificate from the same CA (even for a different domain) will pass CEA validation. Example: If bank.example.com lists "CloudFlare CA" and attacker.example.net has a CloudFlare certificate, MITM using attacker's certificate passes CEA. Mitigation: RFC 5280 hostname validation prevents this attack. CEA

provides CA transparency; RFC 5280 provides hostname verification.

9. Deployment Guidance

9.1. For Domain Owners

Deploying CEA requires publishing a DNS TXT record and maintaining it during CA rotation.

Steps:
1. Identify authorized CAs: List all CAs that issue certificates for your domain
2. Compute SPKI hashes: Extract CA certificates, compute SHA-256 hash of SPKI
3. Publish CEA record: Add "_cea.<domain>" TXT record with pins
4. Enable DNSSEC: Sign CEA records with DNSSEC when possible
5. Monitor: Track CEA query success rates and validation failures

Example:
```
_cea.example.com. TXT "v=CEA1;pins=sha256/X3pGTSOu...;cat=Financial"
```

Update during CA rotation: Add new CA SPKI before deploying new certificate, remove old CA SPKI after migration completes.

9.2. For Client Implementers

Client implementation options:
- Browser extension
- System-level proxy
- TLS library integration
- Enterprise gateway

Recommended features:
- Multi-path consensus for resolver diversity
- Network memory for downgrade detection
- Clear user warnings on validation failure
- Category-based policy automation

9.3. For Enterprise Networks

Enterprises performing TLS inspection should:
1. Publish CEA records for internal services listing inspection CA
2. Respect category exemptions (Financial, Healthcare)
3. Provide transparency to users about interception
4. Maintain clear documentation of interception policy

CEA enables transparent and auditable enterprise inspection while protecting privacy-sensitive categories.

10. IANA Considerations

10.1. DNS TXT Record Prefix Registration

IANA is requested to register the "_cea" DNS subdomain label prefix.

Prefix: _cea
Purpose: Certificate Expectation Assertions for TLS validation
Reference: This document

10.2. CEA Version Registry

IANA is requested to create a CEA Versions registry.

Initial registration:
- Version: CEA1
- Status: Current

- Reference: This document

Registration Procedure: Specification Required [RFC8126]

10.3. CEA Category Registry

IANA is requested to create a CEA Categories registry.

Initial registrations: Financial, Healthcare, Government, Military, Education, E-commerce, Social-Media, Communication, Cloud-Provider, Cloud-Storage, Entertainment, News-Media, Enterprise, IoT, Critical-Infrastructure (as defined in Section 6.1).

Registration Procedure: Expert Review [RFC8126]

10.3.1. Expert Review Criteria

Designated Experts MUST evaluate registration requests against:

1. Legitimate regulatory or compliance purpose: Category must correspond to recognized legal/regulatory framework (e.g., Healthcare for HIPAA, Financial for PCI-DSS).

2. Objective verifiability: Must be possible to verify domain belongs to category (business registration, industry certification, government designation).

3. Non-circumvention purpose: Category must not be designed to circumvent legitimate security monitoring.

Categories designed primarily for convenience without regulatory basis SHOULD be rejected.

10.3.2. Registration Template

Requesters must provide:
- Category name
- Regulatory/compliance basis
- Verification method
- Use case justification
- Reference documentation

10.3.3. Designated Expert Appointment

The Independent Stream Editor will appoint Designated Experts for the CEA Category Registry.

11. Acknowledgments

The author thanks the IETF TLS Working Group for their input and feedback.

This work builds upon the lessons learned from HTTP Public Key Pinning (HPKP) and Certificate Transparency (CT). The author acknowledges the contributions of those who developed and deployed these earlier mechanisms.

12. References

12.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

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

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, DOI 10.17487/RFC4033, March 2005, <https://www.rfc-editor.org/info/rfc4033>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC4035] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Protocol Modifications for the DNS Security Extensions", RFC 4035, DOI 10.17487/RFC4035, March 2005, <https://www.rfc-editor.org/info/rfc4035>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

12.2. Informative References

[RFC6781] Kolkman, O., Mekking, W., and R. Gieben, "DNSSEC Operational Practices, Version 2", RFC 6781, DOI 10.17487/RFC6781, December 2012, <https://www.rfc-editor.org/info/rfc6781>.

[RFC6844] Hallam-Baker, P. and R. Stradling, "DNS Certification Authority Authorization (CAA) Resource Record", RFC 6844, DOI 10.17487/RFC6844, January 2013, <https://www.rfc-editor.org/info/rfc6844>.

[RFC6962] Laurie, B., Langley, A., and E. Kasper, "Certificate Transparency", RFC 6962, DOI 10.17487/RFC6962, June 2013, <https://www.rfc-editor.org/info/rfc6962>.

[RFC7469] Evans, C., Palmer, C., and R. Sleevi, "Public Key Pinning Extension for HTTP", RFC 7469, DOI 10.17487/RFC7469, April 2015, <https://www.rfc-editor.org/info/rfc7469>.

[RFC7858] Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., and P. Hoffman, "Specification for DNS over Transport Layer Security (TLS)", RFC 7858, DOI 10.17487/RFC7858, May 2016, <https://www.rfc-editor.org/info/rfc7858>.

[RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <https://www.rfc-editor.org/info/rfc8484>.

Appendix A. Examples

A.1. Example 1: Financial Institution

A bank publishes the following CEA record:

```
_cea.bank.example.com. 3600 IN TXT "v=CEA1;pins=sha256/X3pGTSOuJeEVw989IJ/cEtXUEmy52zs
1TZQrU06KUKg=,sha256/YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=;cat=Financial"
```

When a user connects from home:
- Browser queries _cea.bank.example.com via DoH (encrypted)
- Receives CEA record (DNSSEC validated, Level 4)
- TLS handshake provides certificate from DigiCert
- Browser extracts SPKI from DigiCert CA certificate
- Computes SHA-256 hash: X3pGTSOuJeEVw989IJ/cEtXUEmy52zs1TZQrU06KUKg=
- Matches first pin in CEA record
- Validation succeeds: [PASS] Direct connection

When a user connects from corporate network with TLS inspection:
- Browser queries _cea.bank.example.com (DNSSEC validated)
- Receives CEA record with bank’s expected SPKI pins
- TLS handshake provides certificate from "CorporateProxyCA"
- Browser extracts SPKI from Corporate CA certificate
- Computes SHA-256 hash: ZZh2eUS0a7Lka31SsAo8KLobRH/vFuMNlChGG3Gvjij=
- Does NOT match any pin in CEA record
- Validation fails: [WARNING] Warning displayed to user

A.2. Example 2: Cloud Service with Multiple CAs

A cloud service provider publishes:

```
_cea.cloud.example.com. 3600 IN TXT "v=CEA1;pins=sha256/AAAA...=,sha256/BBBB...=,sha25
6/CCCC...=;cat=Cloud-Provider"
```


Week 1: Service uses DigiCert certificate
- SPKI hash matches first pin (sha256/AAAA...=)
- Validation succeeds [PASS]

Week 2: Service switches to CloudFlare certificate
- SPKI hash matches second pin (sha256/BBBB...=)
- Validation succeeds [PASS]
- No service disruption, no user warnings

Week 3: Service uses Let’s Encrypt for new regions
- SPKI hash matches third pin (sha256/CCCC...=)
- Validation succeeds [PASS]

- Smooth multi-CA operation with cryptographic validation

A.3. Example 3: Enterprise with Policy Automation

Hospital publishes:

```
_cea.hospital.example.org. 3600 IN TXT "v=CEA1;pins=sha256/YLh1dUR9y6Kja30RrAn7JKnbQG/
uEtLMkBgFF2Fuihg=;cat=Healthcare,HIPAA"
```

Corporate firewall implements policy with verification:

```
IF cea_record.categories CONTAINS "Healthcare" OR "HIPAA" THEN
  IF hostname IN verified_healthcare_allowlist THEN
    BYPASS ssl_inspection
    LOG "Exempted " + hostname + " (verified HIPAA compliance)"
  ELSE
    LOG "WARNING: Unverified Healthcare claim: " + hostname
    PERFORM ssl_inspection
  END IF
ELSE
  PERFORM ssl_inspection
END IF
```

Result:
- Verified hospital traffic is exempted from inspection
- HIPAA compliance with security controls (verified allowlist)
- Audit log provides evidence
- Unverified category claims are logged and inspected normally

Appendix B. Frequently Asked Questions

B.1. Why DNS TXT Records Instead of HTTPS Resource Record?

CEA uses DNS TXT records for immediate deployability (100% DNS server compatibility) and scope separation. The HTTPS RR (RFC 9460) describes "how to connect" (ALPN, ECH), while CEA describes "what to expect when connected" (certificate validation). Future versions may explore HTTPS RR integration.

B.2. Why Not Extend CAA?

CAA (RFC 8659) specifies which CAs MAY issue certificates (issuance authorization). CEA specifies which CAs a domain CURRENTLY uses (expectation assertion). These are orthogonal: CAA is a policy for CAs, CEA is a signal for clients. Extending CAA would conflate these semantics.

B.3. Why Not Solve Via Certificate Transparency?

Certificate Transparency (CT) detects mis-issuance AFTER certificates are logged. CEA provides real-time detection DURING TLS handshake before any data is transmitted. CT and CEA are complementary: CT audits the CA ecosystem, CEA detects network-level interception.

B.4. How Does This Work with Enterprise TLS Inspection?

CEA respects enterprise environments: enterprises can publish CEA records listing their inspection CA, or clients can learn stable enterprise patterns via network memory (Section 7.2.1). CEA does not break legitimate enterprise inspection--it makes interception transparent and auditable.

B.5. What Is Novel Compared to Prior Work?

CEA combines ideas from HPKP (pinning), DANE (DNS-based trust), and resolver diversity. The novelty is the integration: multi-path DNS

consensus for increased attack resistance, semantic categories for policy automation, and network memory for warning fatigue mitigation. See Section 1.5 for detailed comparison to prior mechanisms.

Authors' Addresses

Vineeth Joseph
Palo Alto Networks
United States

Email: vinjoseph@paloaltonetworks.com, vintjoseph871@gmail.com

Ethan Hamadeh
Palo Alto Networks
United States

Email: ehamadeh@paloaltonetworks.com

Chen Zhang
Palo Alto Networks
United States

Email: tozhang@paloaltonetworks.com, tozh300@gmail.com.