

Defensible 10

Annex I (Normative): D09-Cryptography, Encryption & Key Management

Technical Standards

Standards Committee
1-12-2026

© **2026 ISAUnited.org**. Non-commercial use permitted under CC BY-NC. Commercial integration requires ISAUnited licensing.

Obsolete and withdrawn documents should not be used; please use replacements.

Copyright 2026. The Institute of Security Architecture United. All rights reserved

About ISAUnited

The Institute of Security Architecture United is the first dedicated Standards Development Organization (SDO) focused exclusively on cybersecurity architecture and engineering through security-by-design. As an international support institute, ISAUnited helps individuals and enterprises unlock the full potential of technology by promoting best practices and fostering innovation in security.

Technology drives progress; security enables it. ISAUnited equips practitioners and organizations across cybersecurity, IT operations, cloud/platform engineering, software development, data/AI, and product/operations with vendor-agnostic standards, education, credentials, and a peer community—turning good practice into engineered, testable outcomes in real environments.

Headquartered in the United States, ISAUnited is committed to promoting a global presence and delivering programs that emphasize collaboration, clarity, and actionable solutions to today's and tomorrow's security challenges. With a focus on security by design, the institute champions the integration of security into every stage of architectural and engineering practice, ensuring robust, resilient, and defensible systems for organizations worldwide.

Obsolete and withdrawn documents should not be used; please use replacements.

Disclaimer

ISAUnited publishes the ISAUnited Defensible 10 Standards Technical Guide to provide information and education on security architecture and engineering practices. While efforts have been made to ensure accuracy and reliability, the content is provided “as is,” without any express or implied warranties. This guide is for informational purposes only and does not constitute legal, regulatory, compliance, or professional advice. Consult qualified professionals before making decisions.

Limitation of Liability

ISAUnited - and its authors, contributors, and affiliates - shall not be liable for any direct, indirect, incidental, consequential, special, exemplary, or punitive damages arising from the use of, inability to use, or reliance on this guide, including any errors or omissions.

Operational Safety Notice

Implementing security controls can affect system behavior and availability. First, validate changes in non-production, use change control, and ensure rollback plans are in place.

Third-Party References

This guide may reference third-party frameworks, websites, or resources. ISAUnited does not endorse and is not responsible for the content, products, or services of third parties. Access is at the reader’s own risk.

Use of Normative Terms (“Must”, “Should”)

- Must: A mandatory requirement for conformance to the standard.
- Must Not: A prohibition; implementations claiming conformance shall not perform the stated action.
- Should: A strong recommendation; valid reasons may exist to deviate in particular circumstances, but the full implications must be understood and documented.

Acceptance of Terms

By using this guide, readers acknowledge and agree to the terms in this disclaimer. If you disagree, refrain from using the information provided.

For more information, please visit our [Terms and Conditions](#) page.

Obsolete and withdrawn documents should not be used; please use replacements.

License & Use Permissions

The Defensible 10 Standards (D10S) are owned, governed, and maintained by the Institute of Security Architecture United (ISAUnited.org).

This publication is released under a Creative Commons Attribution–NonCommercial License (CC BY-NC).

Practitioner & Internal Use (Allowed):

- You are free to download, share, and apply this standard for non-commercial use within your organization, departments, or for individual professional, academic, or research purposes.
- Attribution to ISAUnited.org must be maintained.
- You may not modify the document outside of Sub-Standard authorship workflows governed by ISAUnited, excluding the provided Defensible 10 Standards templates and matrices.

Commercial Use (Prohibited Without Permission):

- Commercial entities seeking to embed, integrate, redistribute, automate, or incorporate this standard in software, tooling, managed services, audit products, or commercial training must obtain a Commercial Integration License from ISAUnited.

To request permissions or licensing:
info@isaunited.org

Standards Development & Governance Notice

This standard is one of the ten Parent Standards in the Defensible 10 Standards (D10S) series. Each Parent Standard is governed by ISAUnited's Standards Committee, peer-reviewed by the ISAUnited Technical Fellow Society, and maintained in the Defensible 10 Standards GitHub repository for transparency and version control.

Contributions & Collaboration

ISAUnited maintains a public GitHub repository for standards development.

Practitioners may view and clone materials, but contributions require:

- ISAUnited registration and vetting
- Approved Contributor ID
- Valid GitHub username

All Sub-Standard contributions must follow the Defensible Standards Submission Schema (D-SSF) and are peer-reviewed by the Technical Fellow Society during the annual Open Season.

Obsolete and withdrawn documents should not be used; please use replacements.

Abstract

The ISAUnited Defensible 10 Standards provide a structured, engineering-grade framework for implementing robust and measurable cybersecurity architecture and engineering practices. The guide outlines the frameworks, principles, methods, and technical specifications required to design, build, verify, and operate reliable systems.

Developed under the ISAUnited methodology, the standards align with modern enterprise realities and integrate Security by Design, continuous technical validation, and resilience-based engineering to address emerging threats. The guide is written for security architects and engineers, IT and platform practitioners, software and product teams, governance and risk professionals, and technical decision-makers seeking a defensible approach that is testable, auditable, and scalable.

This document includes a series of Practitioner Guidance, Cybersecurity Students & Early-Career Guidance, and Quick Win Playbook callouts.



Practitioner Guidance- Actionable steps and patterns to apply the technical standards in real environments.



Cybersecurity Student & Early-Career Guidance- Compact, hands-on activities that turn each section's ideas into a small, verifiable artifact.



Quick Win Playbook- Immediate, evidence-driven actions that improve posture now while reinforcing good engineering discipline.

Together, these elements help organizations translate intent into engineered outcomes and sustain long-term protection and operational integrity.

Obsolete and withdrawn documents should not be used; please use replacements.

Foreword

Message from ISAUnited Leadership

Cybersecurity is at a turning point. As digital systems scale, reactive and checklist-driven practices do not keep pace with adversaries. The ISAUnited position is clear: security must be practiced as engineered design, grounded in scientific principles, structured methods, and defensible evidence. Our mission is to professionalize cybersecurity architecture and engineering with standards that are actionable, testable, and auditable.

ISAUnited Defensible 10 Standards: First Edition is a practical framework for that shift. The standards in this book are not theoretical. They translate intent into measurable specifications, controls, and verification, and enable teams to design and operate resilient systems at enterprise scale.

About This First Edition

This edition publishes 10 Parent Standards, one for each core domain of security architecture and engineering. Sub-standards will follow in subsequent editions, contributed by ISAUnited members and reviewed by our Technical Fellow Society, to provide focused, technology-aligned detail. Adopting the Parent Standards now positions organizations for seamless integration of Sub Standards as they are released on the ISAUnited annual update cycle.

Why “Defensible Standards”

Defensible means the work can withstand technical, operational, and adversarial scrutiny. These standards are designed to be demonstrated with evidence, featuring clear architecture, measurable specifications, and verification, so that practitioners can confidently stand behind their designs.

Obsolete and withdrawn documents should not be used; please use replacements.

Contents

Section 1. Standard Introduction.....	10
Section 2. Definitions	12
Section 3. Scope.....	16
Section 4. Use Case	18
Section 5. Requirements (Inputs)	21
Section 6. Technical Specifications (Outputs)	24
Section 7. Cybersecurity Core Principles.....	29
Section 8. Foundational Standards Alignment.....	31
Section 9. Security Controls	35
Section 10. Engineering Discipline	39
Section 11. Associate Sub-Standards Mapping.....	44
Section 12. Verification and Validation (Tests)	47
Section 13. Implementation Guidelines	53
Appendices.....	59
Appendix A: Engineering Traceability Matrix (ETM).....	59
Appendix B: Evidence Pack Matrix	63

Obsolete and withdrawn documents should not be used; please use replacements.

Annex I (Normative): D09-Cryptography, Encryption & Key Management

Obsolete and withdrawn documents should not be used; please use replacements.

ISAUnited's Defensible 10 Standards

Parent Standard: D09-Cryptographic, Encryption, and Key Management

Document: ISAU-DS-CEK-1000

Last Revision Date: January 2026

Peer-Reviewed By: ISAUnited Technical Fellow Society

Approved By: ISAUnited Standards Committee

Obsolete and withdrawn documents should not be used; please use replacements.

Section 1. Standard Introduction

Cryptography, encryption, and key management form the engineering foundation for protecting the confidentiality, integrity, authenticity, and non-repudiation of enterprise data and transactions. As organizations operate across on-premises, public-cloud, and edge environments, cryptographic responsibilities have evolved from isolated code libraries into platform-wide services that must be architected, instrumented, and validated throughout their lifecycle. Without disciplined design and governance, weak randomness, incorrect cipher modes, certificate sprawl, expired or mis-issued certificates, and uncontrolled key material create systemic risks that bypass traditional network and endpoint defenses. The increasing demand for cryptographic agility and post-quantum readiness adds architectural complexity and drives the long-term cost of cryptographic debt.

This standard provides the authoritative foundation for designing, implementing, and operating secure and resilient cryptographic architectures. It applies to cybersecurity engineers, security architects, platform and SRE teams, and technical leaders responsible for delivering measurable, defensible encryption outcomes for data at rest, in transit, and in use. The guidance defines algorithm and parameter baselines, module validation expectations, transport security profiles (TLS and mTLS), enterprise PKI architecture and certificate lifecycle automation, secrets management practices, and key lifecycle operations using HSM and KMS technologies. Each capability must produce verifiable evidence of protection. The goal is to establish and validate interoperable, auditable, and sustainable cryptographic services across diverse platforms with clear ownership, separation of duties, and evidence-producing controls.

Objective

This standard defines foundational engineering principles for Cryptography, Encryption, and Key Management and guides practitioners through a structured approach for safeguarding data and enabling trustworthy communications.

1. Define architectural requirements, trust boundaries, and lifecycle expectations for cryptographic modules and services across enterprise, cloud, and hybrid environments.
2. Standardize defensible cryptographic patterns for data protection and service identity, including envelope encryption and mutual TLS.
3. Establish measurable governance for key lifecycles, including generation, protection, rotation, revocation, escrow where authorized, and cryptographic erasure.

Obsolete and withdrawn documents should not be used; please use replacements.

4. Require secure defaults, dual control for sensitive operations, and telemetry that supports auditability and forensic reconstruction.
5. Institutionalize cryptographic agility as an ongoing capability, including a documented transition roadmap for post-quantum readiness aligned to enterprise risk.

Justification

Adversaries increasingly target cryptographic weaknesses rather than application logic or perimeter defenses. Credential and key theft, TLS downgrade and interception, certificate misissuance, side-channel leakage, and embedded secrets in code or images remain common failure modes that lead to silent data exposure and operational outages. Although foundational frameworks such as FIPS 140-3 and NIST/ISO guidance provide essential baselines, they alone do not deliver the architectural specificity, lifecycle rigor, or measurable criteria needed to secure distributed systems at scale.

This standard addresses that gap by applying a security-by-design methodology to cryptography and key management. It unifies transport security, PKI, secrets management, and key operations within a single architectural framework; integrates continuous conformance testing into delivery pipelines and runtime environments; and defines evidence-based success measures for rotation, coverage, and incident response. By adopting this standard, organizations and academic institutions can equip engineers and architects to design, verify, and defend trustworthy encryption, sustain secure interoperability, and prepare for future algorithmic transitions.

Evidence

Evidence Packs (EPs) provide the proof layer for adopting this Parent Standard. For Domain 09, the Evidence Pack repository is EP-09 (D09) and is organized to mirror the sections that drive traceability and adoption:

- EP-09.1 Requirements (Inputs)
- EP-09.2 Technical Specifications (Outputs)
- EP-09.3 Foundational Standards
- EP-09.4 Control Mappings
- EP-09.5 Verification and Validation activities.

This structure links architectural intent in Section 5 to measurable implementation in Section 6, and then to Verification and Validation in Section 12, enabling organizations

Obsolete and withdrawn documents should not be used; please use replacements.

to demonstrate conformance through repeatable, time-bound artifacts rather than declarations.

Section 2. Definitions

These definitions ensure a consistent understanding and interpretation across ISAUnited members, implementers, and peer reviewers, supporting defensible engineering and implementation practices. Where possible, definitions align with industry-recognized terminology from NIST, ISO, and ISAUnited's internal frameworks and methodologies.

Asymmetric Cryptography – Public key cryptography that uses mathematically linked key pairs for confidentiality, key establishment, or digital signatures. Examples include RSA, ECDSA, EdDSA, and ECDH/ECDHE.

Authenticated Encryption (AEAD) – Schemes that provide confidentiality and integrity in a single construction. Common examples include AES-GCM and ChaCha20-Poly1305.

Certificate Authority (CA) – Trusted issuer that signs certificates, including offline root CAs and online intermediate CAs constrained by certificate policies.

Certificate Revocation List (CRL) – Periodically published list of revoked certificates. Clients may check CRLs when required by policy.

Certificate Transparency (CT) – Public append-only logs of issued certificates that enable monitoring and detection of misissuance.

Cipher Suite – Named collection of algorithms and parameters used by Transport Layer Security. Organizational policy standardizes allowed suites and disables legacy algorithms.

Cryptographic Erasure – Rendering data unrecoverable by destroying or invalidating the keys that protect it; often preferable to media sanitization for encrypted data.

Cryptoperiod – Maximum recommended time or usage volume for a key before mandatory rotation or retirement based on risk and exposure.

Data at Rest, in Transit, and in Use – Taxonomy describing where protections apply: storage and backups, network communications, and processing within memory or enclaves.

Data Encryption Key (DEK) – Key used to encrypt application or storage data. Typically short-lived and rotated frequently.

Obsolete and withdrawn documents should not be used; please use replacements.

Deterministic Encryption – Encryption that produces the same ciphertext for identical plaintext and key, enabling limited equality matching while increasing information leakage. Apply only with documented risk trade-offs.

Downgrade Attack – Adversary-induced negotiation of weaker protocols or parameters. Enforce minimum versions and deny legacy algorithms to prevent.

Envelope Encryption – Pattern where data is encrypted with a DEK and the DEK is wrapped by a KEK under a separate trust boundary.

Entropy – Measure of unpredictability required for secure key generation and nonces. Insufficient entropy leads to predictable keys or repeated nonces.

Format Preserving Encryption (FPE) – Encryption that preserves the format of structured fields (for example, numeric strings). Use only when necessary and after conducting a risk assessment.

Hardware Security Module (HSM) – Tamper-resistant hardware that generates, stores, and uses keys within a validated boundary, often required for root-of-trust operations.

Hash Function – One-way function that maps input data to a fixed-size digest used for integrity, deduplication, and signing workflows. Modern choices include SHA-256 and SHA-384.

HTTP Strict Transport Security (HSTS) – Policy directing clients to use HTTPS only for a domain, reducing downgrade and stripping risks for public endpoints.

Hybrid Key Exchange or Signature – Combination of classical and post-quantum algorithms in a single operation to maintain compatibility while increasing quantum resistance.

Initialization Vector (IV) and Nonce – Unique per-message values required by many modes to ensure security. IVs and nonces must never repeat with the same key.

Key Attestation – Cryptographic proof that a key was generated and is stored within an approved hardware or service boundary, bound to workload identity.

Key Ceremony – Controlled, documented process for generating, activating, backing up, rotating, and retiring high-value keys with witnesses and evidentiary artifacts.

Key Derivation Function (KDF) – Function that derives one or more cryptographic keys from input keying material. HKDF is widely used with salts and context. Password-based KDFs should be memory-hard, for example, scrypt or Argon2.

Obsolete and withdrawn documents should not be used; please use replacements.

Key Encapsulation Mechanism (KEM) – Asymmetric primitive used to establish shared secrets without directly transmitting them. Used in both classical and post-quantum key exchange.

Key Encryption Key (KEK) – Key used to protect other keys through wrapping. Segregate KEKs by domain to reduce blast radius.

Key Escrow and Archival – Controlled retention of key material to satisfy recovery or regulatory requirements under dual control and strict audit.

Key Management Service (KMS) – Centralized service that manages key lifecycles and enforces key-usage policies, often backed by HSMs and integrated with workload identities.

Key Wrapping – Standardized method for encrypting keys while preserving integrity and binding metadata.

Message Authentication Code (MAC) – Integrity and authenticity protection using a symmetric key. HMAC with SHA-256 or SHA-384 is recommended when AEAD is not applicable.

Mutual TLS (mTLS) – Transport Layer Security mode where both client and server authenticate with certificates to implement strong service identity.

Online Certificate Status Protocol (OCSP) – Protocol for obtaining near real-time certificate revocation status. OCSP stapling improves reliability and performance.

Perfect Forward Secrecy (PFS) – Property that protects past sessions even if long-term keys are later compromised, typically achieved via ephemeral Diffie-Hellman.

Post-Quantum Cryptography (PQC) – Cryptographic algorithms designed to resist quantum adversaries, covering key encapsulation and signature schemes. May be deployed in hybrid modes during transition.

Public Key Infrastructure (PKI) – Policies, roles, software, and procedures for issuing, distributing, validating, and revoking certificates and managing trust anchors.

Random Number Generator (CSPRNG or DRBG) – Cryptographically secure generator used for keys, nonces, and salts. Must be seeded from high-entropy sources and follow approved constructions.

Registration Authority (RA) – Entity that performs identity vetting and approves certificate requests on behalf of a Certificate Authority.

Salt – Non-secret value that randomizes key derivation and hashing operations to resist precomputation attacks.

Obsolete and withdrawn documents should not be used; please use replacements.

Secrets – Confidential values such as passwords, API keys, tokens, private keys, and connection strings that must be issued, stored, rotated, and revoked under policy.

Secrets Management – Processes and systems for securely issuing, storing, delivering, rotating, and revoking secrets with auditability and least privilege.

Short-Lived Certificates – Certificates with reduced validity periods that lower the impact of key compromise and simplify revocation.

Side-Channel Attack – An attack that derives secrets from implementation artifacts such as timing, cache behavior, or power usage. Constant-time operations and isolation reduce risk.

Split Knowledge and Dual Control – Safeguards ensuring no single person possesses complete key material or can unilaterally perform sensitive actions, commonly enforced as M-of-N approvals.

Symmetric Encryption – Encryption that uses a single shared secret key for both encryption and decryption. Modern practice favors authenticated modes such as AES-GCM. ChaCha20-Poly1305 is preferred on devices without AES acceleration.

Transport Layer Security (TLS) – A protocol that provides confidentiality and integrity for data in transit. TLS 1.3 is preferred. TLS 1.2 is allowed only by exception with restricted cipher suites.

Trusted Execution Environment (TEE) – Hardware-backed isolated environment that protects code and data in use. Remote attestation proves enclave identity and state to verifiers.

Trust Store – Curated collection of trusted root certificates or keys used by clients and services to validate presented certificates.

X.509 Certificate – Standard object that binds a subject to a public key and attributes, signed by a Certificate Authority. Includes fields such as Subject Alternative Name and key usage.

Zeroization – Reliable clearing of sensitive key material from volatile memory or storage.

Obsolete and withdrawn documents should not be used; please use replacements.

Section 3. Scope

Cryptography, encryption, and key management encompass the engineering practices, services, and controls that protect the confidentiality, integrity, authenticity, and non-repudiation of enterprise data and communications. As organizations operate across interconnected environments, including on-premises, public and private clouds, SaaS, edge, and operational technology (OT) and industrial control systems (ICS), the complexity of selecting algorithms, managing keys and certificates, enforcing transport security, and validating conformance has expanded substantially. Modern enterprises now depend on cryptographic engineering as an integrated platform discipline. This parent standard defines the architectural expectations and technical guardrails required to build and sustain a defensible CEK posture across the enterprise. It helps practitioners eliminate plaintext secrets, standardize protocol versions and cipher suites, enforce key-lifecycle discipline, mitigate downgrade and side-channel risks, and maintain operational efficiency while aligning with regulatory obligations and enterprise risk tolerance.

Applicability

- **All Data States and Cryptographic Artifacts:** Applies to data at rest, in transit, and in use, and to keys, certificates, secrets, signatures, and integrity tags protecting applications, services, storage, backups, and code artifacts.
- **Enterprise and Academic Environments:** Intended for security architects, crypto officers, PKI engineers, platform and SRE teams, application-security engineers, and academic programs advancing cryptographic-engineering practice.
- **Hybrid and Multi-platform Architectures:** Addresses unifying CEK controls across data centers, multiple cloud providers, SaaS platforms, mobile and endpoint fleets, edge and IoT devices, and OT/ICS systems.
- **Environment Coverage:** Applies to production, staging, development, and test environments; exceptions for legacy or constrained systems require compensating controls and time-bound remediation.

Key Focus Areas

The following focus areas define the core engineering domains for CEK implementation:

- **Algorithm and Parameter Governance:** Establish and maintain an enterprise policy registry defining approved algorithms, modes, key sizes, and hash functions with deprecation timelines.

Obsolete and withdrawn documents should not be used; please use replacements.

- **Key Lifecycle Management:** Govern generation, distribution, storage, usage, rotation, escrow, and archival where authorized, revocation, and destruction with defined cryptoperiods, dual control, split knowledge, and auditable workflows.
- **PKI and Certificate Lifecycle:** Design rooted and intermediate CA hierarchies, automate issuance and renewal with short-lived certificates, implement OCSP/CRL and certificate-transparency controls, and standardize validation requirements.
- **Transport and Session Security:** Standardize TLS and mTLS profiles, SSH for administration, and, where appropriate, IPsec or QUIC; enforce minimum protocol versions, perfect forward secrecy, and HSTS for public endpoints.
- **Secrets Management:** Issue dynamic, short-lived credentials bound to workload identity; prohibit hard-coded secrets; integrate secret scanning in CI/CD and image supply chains.
- **Randomness and Entropy:** Utilize approved DRBGs and CSPRNGs, validate hardware and OS entropy sources, and confirm nonce and IV uniqueness.
- **Module Assurance and Library Hygiene:** Use validated modules where mandated, maintain version inventories, enable self-tests and known-answer tests, and apply compiler and memory-safety hardening to crypto-adjacent code paths.
- **Cryptographic Agility and PQC Readiness:** Define capability profiles, pilot hybrid KEMs and signature schemes, set migration triggers and rollback criteria, and maintain compatibility matrices.
- **Data Encryption Patterns:** Apply envelope encryption with domain-segregated KEKs; evaluate deterministic or format-preserving encryption only with documented trade-offs; encrypt backups with independent keys and enforce cryptographic erasure.
- **Observability and Evidence:** Generate signed and tamper-evident audit telemetry for key usage and administration, maintain accurate inventories of keys and certificates, define rotation and validity SLOs, and detect anomalies, downgrade attempts, and misuse.
- **Key Ceremonies, Dual Control, and Attestation:** Formalize ceremonies for high-value keys, require M-of-N approvals for sensitive operations, and use hardware or service attestation to prove key provenance and residency.
- **Downgrade and Side-Channel Mitigations:** Enforce minimum protocol versions and cipher suites, use constant-time operations, and isolate sensitive computations to minimize leakage.

Outcomes

By defining this scope, the standard ensures that cryptography, encryption, and key management are:

- **Define:** Establish cryptographic inventory and trust anchors.
- **Design:** Specify algorithm policy and key lifecycle design.
- **Deploy:** Implement key storage, rotation, and certificate baselines.

Obsolete and withdrawn documents should not be used; please use replacements.

- **Detect:** Monitor key usage anomalies and policy violations.
- **Defend:** Execute revocation, rotation, and compromise handling.
- **Demonstrate:** Produce cryptographic verification and rotation proof.

Together, these elements provide the foundation for resilient, secure, and auditable cryptographic services that protect critical assets, enable trustworthy communications, and sustain enterprise operations without compromising security.

Section 4. Use Case

Achieving resilient cryptographic operations requires more than implementing algorithms or rotating keys; it demands engineered lifecycle control across hybrid, distributed environments. The following consolidated use case represents a realistic enterprise scenario faced by organizations operating across on-premises, multi-cloud, SaaS, and edge platforms. It highlights common weaknesses in certificate management, key handling, and secrets governance, and demonstrates how these deficiencies propagate into systemic operational risk. Each element of the use case maps these weaknesses to targeted engineering countermeasures, including enterprise PKI automation, HSM/KMS-based key governance, TLS/mTLS profile standardization, and post-quantum agility. The result is a defensible cryptographic architecture where encryption, key management, and validation processes are measured, automated, and continuously verifiable against ISAUnited engineering standards.

Table I-1:

Use Case Name	Unified Enterprise PKI, mTLS, and Key Lifecycle to Eliminate Certificate Outages and Secrets Sprawl
Objective	Standardize and automate cryptographic services across hybrid environments to achieve reliable encryption for data in transit and at rest, eliminate plaintext secrets, prevent certificate-related outages, and establish cryptographic agility, including post-quantum readiness, through enterprise PKI, automated certificate lifecycle management, disciplined key governance (HSM/KMS), and continuous conformance testing.
Scenario	A global financial-services organization operating across two public clouds and an on-premises data center experienced recurring outages due to expired certificates, inconsistent TLS configurations, and hard-coded secrets in source repositories and container images. Keys were generated on developer laptops and exported in plaintext for backups. The enterprise lacked an authoritative inventory of keys and certificates, dual

Obsolete and withdrawn documents should not be used; please use replacements.

	control for KEK operations, and any roadmap for post-quantum transition — illustrating systemic cryptographic drift that the CEK Parent Standard is designed to eliminate.
Actors	Crypto Security Architect; PKI Engineer; Platform/SRE Engineer; Application Security Engineer; DevSecOps Engineer; SOC Analyst; Risk and Compliance Officer.
Adversary Mapping	<p>Design-time Threat Models: STRIDE categories – Information Disclosure, Tampering, Elevation of Privilege, and Repudiation.</p> <p>ATT&CK examples: T1552 Unsecured Credentials; T1555 Credentials from Password Stores; T1040 Network Sniffing; T1606.001 Web Cookies; T1606.002 SAML Tokens; T1588.003 Obtain Capabilities: Code Signing Certificates; T1588.004 Obtain Capabilities: Digital Certificates; T1587.003 Develop Capabilities: Digital Certificates.</p> <p>Kill Chain Phases: Weaponization (certificate forgery), Delivery (man-in-the-middle or supply-chain injection), Exploitation (downgrade attack or key reuse), Installation (secret implant), C2 (session hijacking via compromised certs), Actions on Objectives (data decryption or credential abuse).</p> <p>Failure Vectors Addressed: Certificate spoofing, key theft from source repos, TLS downgrade, side-channel leakage, and algorithmic obsolescence.</p>
Challenges Identified	Certificate expiry and drift from manual renewals, inconsistent validity periods; secrets sprawl across code repositories and pipelines; TLS cipher inconsistencies and downgrade risk; uncontrolled key generation outside HSM/KMS boundaries; limited observability and audit trails; and an absence of a cryptographic agility and PQC readiness plan.
Technical Solution	<p>1) Crypto Policy and Governance: Establish an enterprise cryptographic-policy catalog defining approved algorithms, modes, key sizes, and DRBGs with deprecation timelines. Enforce policy via CI/CD gates and runtime validation; monitor entropy and DRBG health.</p> <p>2) PKI and Certificate Lifecycle Automation: Implement a tiered PKI with an offline root and constrained intermediates; adopt ACME-compatible issuance; require short-lived (≤ 90-day) leaf certificates with automated renewal; enable OCSP stapling and certificate transparency for public endpoints.</p> <p>3) Transport Security Standardization: Define a single enterprise TLS/mTLS profile (TLS 1.3 preferred, 1.2 by exception); require PFS and strict hostname/SAN validation; apply HSTS for public services; centralize policy through service mesh or gateway.</p> <p>4) Key Management Operations: Generate and store keys within HSM/KMS boundaries; deny plaintext exports; enforce dual control and split knowledge for KEK operations; apply envelope encryption with domain-segregated KEKs; automate rotation ($\text{DEK} \leq 90$ days, $\text{KEK} \leq 12$ months); perform cryptographic erasure for decommissioned datasets.</p> <p>5) Secrets Management: Prohibit hard-coded secrets; integrate scanners in pre-commit hooks, CI pipelines, and image builds; issue dynamic, short-lived credentials bound to workload identity with automatic revocation and least-privilege policies.</p>

Obsolete and withdrawn documents should not be used; please use replacements.

	<p>6) Observability and Evidence Production: Emit signed audit telemetry for key generation, usage, and administrative actions; maintain authoritative inventories of keys and certificates; integrate with SIEM for anomaly detection and alerting.</p> <p>7) Cryptographic Agility and PQC Readiness: Maintain capability profiles and compatibility matrices; pilot hybrid KEMs and signature schemes in pre-production; define migration triggers and rollback criteria aligned to enterprise risk and vendor readiness.</p>
Expected Outcome	<p>Zero P1 outages from certificate expiry (100 % automatic renewal; median ≤ 5 minutes). 100 % TLS coverage for north-south traffic and ≥ 98 % mTLS adoption for east-west and administrative channels with no downgrades. ≥ 95 % reduction in secrets-in-code findings within 90 days and 100 % dynamic secret usage in production. ≥ 99 % rotation SLA adherence (DEK ≤ 90 days, KEK ≤ 12 months); revocation on compromise ≤ 15 minutes; zero plaintext private-key exports. FIPS 140-3 validated modules deployed where required; key ceremonies audited with M-of-N controls. Documented PQC migration plan with successful pre-production hybrid pilot and defined rollback path.</p>
Evidence Artifacts	<p>PKI hierarchy and certificate-transparency logs; ACME automation records; HSM/KMS audit trails; rotation and revocation reports; DRBG and entropy health dashboards; CI/CD policy-as-code validation outputs; secrets-scanner findings and remediation tickets; SIEM correlation alerts on key usage and certificate events; PQC pilot performance and rollback reports.</p> <p>Evidence Pack ID: EP-09.5 (supporting implementation artifacts cross-linked to EP-09.2).</p>

Key Takeaways

- **Cryptographic Discipline Requires Automation:** Manual certificate renewals and unmanaged key operations create systemic fragility. Automating PKI issuance, rotation, and validation is crucial for maintaining the reliability of encryption.
- **Centralized Trust and Policy Reduce Drift:** A single enterprise cryptographic-policy catalog—enforced by CI/CD and runtime checks—eliminates divergence in cipher suites, key sizes, and algorithm use.
- **Lifecycle Governance Is the Control Plane:** Key ceremonies, dual control, and cryptoperiod adherence transform encryption from a static control into an auditable engineering process.
- **Visibility Is Verifiability:** Signed telemetry, HSM/KMS audit logs, and certificate-transparency data provide measurable evidence that cryptographic operations are functioning and defensible.
- **Agility Must Be Engineered Early:** Preparing for post-quantum transitions through hybrid KEM and signature pilots ensures the enterprise can evolve without disruption when new cryptographic standards mature.
- **Security Debt Is Cumulative:** Weak randomness, expired certificates, and secrets sprawl accumulate silently—engineering rigor and automation are the only scalable countermeasures.

Obsolete and withdrawn documents should not be used; please use replacements.



Practitioner Guidance:

For Implementation Teams: Begin with an authoritative inventory of certificates, keys, and secrets across all environments. Use that inventory as the baseline before enabling automation. Integrate certificate issuance and renewal automation, HSM and KMS key rotation, and secrets scanning into CI/CD pipelines so assurance is continuous rather than periodic. Store implementation evidence under EP-09.2 and attach validation proof under EP-09.5.

For Security Architects: Define enterprise cryptographic SLOs for rotation adherence, renewal latency, and mTLS coverage, then instrument them in dashboards sourced from PKI, SIEM, and HSM telemetry. Review results quarterly and tie each metric to its Evidence Pack location, including change records, test outputs, and corrective actions.

For Leadership and Compliance Teams: Treat defensible encryption as an engineering service with measurable performance, not a compliance statement. Require evidence-producing controls and time-bound metrics such as renewal latency and revocation interval, and verify that supporting artifacts are recorded under the EP-09 structure during audit sampling.

Section 5. Requirements (Inputs)

To implement the Cryptography, Encryption & Key Management (CEK) Architecture defensibly, organizations shall maintain the following baseline architectural and environmental conditions. These prerequisites define the minimum engineering posture from which all technical specifications can be validated and enforced.

Foundational Requirements

5.1 Enterprise Cryptographic Policy and Governance

A formally approved cryptographic policy catalog Must exist, defining approved algorithms, modes, key sizes, deterministic random bit generators, protocol profiles, and deprecation timelines. The catalog Must align with NIST, ISO/IEC, and ISAUnited standards and include version control, ownership, and periodic review.

Obsolete and withdrawn documents should not be used; please use replacements.

5.2 Hardware Security Module and Key Management Service

An enterprise-grade hardware security module or cloud key management service Must be operational to generate, store, and manage cryptographic keys, enforce key-usage policies, and support dual control and split knowledge for key-encryption-key operations. Audit logs from these services Must be tamper-evident and centrally collected.

5.3 Public Key Infrastructure

A public key infrastructure hierarchy Must be established with an offline root certificate authority and one or more constrained intermediate certificate authorities. Certificate issuance Must support automated workflows for internal services and include online certificate status protocol and certificate revocation list publication. Certificate transparency Must be implemented where applicable.

5.4 Secrets Management Platform

A centralized, policy-driven secrets management platform Must issue, rotate, and revoke credentials, API keys, and tokens. The platform Must integrate with workload identities and enforce short-lived, dynamically generated secrets.

5.5 Time Synchronization Service

Authenticated time sources Must be enforced across all systems participating in cryptographic operations to preserve certificate validity, signature accuracy, and log correlation.

5.6 Secure Software Supply Chain Controls

Only approved, actively maintained, and vetted cryptographic libraries and modules Must be used. Build systems Must verify provenance, apply compiler hardening, and forbid custom or unvalidated cryptography.

5.7 Network and Transport Readiness

All in-scope network paths Must support TLS 1.3. TLS 1.2 Must be treated as an exception with restricted cipher suites. Mutual TLS Must protect service-to-service and administrative channels, and certificate-validation policies Must be enforced.

5.8 Audit-Ready Logging Infrastructure

Key lifecycle events, certificate issuance and revocation, and secrets access Must be centrally logged using digitally signed, tamper-evident records. Retention Must meet compliance requirements and support audit and incident response.


5.9 Entropy Sources and Randomness Assurance

Systems Must use reliable, high-entropy sources to seed deterministic random bit generators and cryptographically secure random number generators. Monitoring Must detect entropy degradation or failure. Entropy health reports Must be recorded as Evidence Pack readiness artifacts.

Obsolete and withdrawn documents should not be used; please use replacements.

5.10 Post-Quantum Readiness Assessment

A baseline post-quantum readiness assessment Must be completed, documenting systems, protocols, and dependencies requiring migration. The assessment Must include risk ranking and transition prioritization.

	<p>Practitioner Guidance:</p> <p>When validating readiness for CEK implementation, practitioners should prioritize enterprise cryptographic governance and key-generation boundaries before advanced work such as post-quantum pilots or hybrid key exchanges. Gaps in certificate automation, secrets lifecycle enforcement, or HSM and KMS integration undermine downstream controls.</p> <ul style="list-style-type: none"> • Use a one-page readiness gate: List Requirements 5.1–5.10 with owner, current status, and Evidence Pack link. Do not proceed until each row is green and dated. • Baseline before change: Record current metrics for certificate-expiry incidents, secrets-in-code findings, mutual TLS coverage, rotation success rate, and clock skew. Use these as the control group for § 6 SLO validation. • Fail fast on blockers: If Requirement 5.2 or 5.3 is missing, pause downstream work and log a tracked risk; § 6 cannot be implemented defensibly without them. • Validate evidence continuously: Require each prerequisite to generate signed artifacts (HSM audit logs, PKI topology, entropy reports) stored under the active Evidence Pack ID.
---	--

Evidence Pack

Record evidence Must be collected for Section 5 prerequisites in EP-09.1 (Requirements). Each requirement in 5.1 through 5.10 Must have at least one dated artifact that identifies the owner, the current status, and the enforcement boundary. Evidence Must be version-controlled and retained according to organizational audit requirements.

Minimum evidence expectations for EP-09.1 include:

- Policy and governance artifacts: cryptographic policy catalog, algorithm and parameter registry, deprecation timelines, and approval record.
- Platform readiness artifacts: HSM or KMS boundary documentation, PKI topology and certificate policy, secrets platform configuration baseline, and authenticated time synchronization configuration.

Obsolete and withdrawn documents should not be used; please use replacements.

- Supply chain and transport artifacts: approved cryptographic library allowlist, provenance or bill of materials policy, and transport profile baseline.
- Audit and assurance artifacts: logging and retention configuration baseline, entropy health baseline, and the post-quantum readiness assessment with dependency inventory and risk ranking.

EP-09.1 entries Must link forward to implementation proof in EP-09.2 (Technical Specifications) and to test results in EP-09.5 (Verification and Validation) where applicable.

Section 6. Technical Specifications (Outputs)

Technical specifications define the engineered outputs required to realize this standard. Each specification represents a distinct architectural domain that translates cryptographic policy into measurable, auditable results. Together, these specifications establish a resilient foundation for enterprise cryptography, encryption, and key management across on-premises, cloud, and hybrid environments, producing verifiable artifacts that demonstrate assurance and accountability.

Outputs must be:

- **Measurable:** validated by scans, logs, audits, or tests
- **Actionable:** implementation-ready, not policy slogans
- **Aligned:** traceable to §5 Requirements and sub-standards

6.1 Algorithm & Parameter Baselines

- Approved Symmetric Algorithms: AES-GCM (128 / 256) as the primary AEAD mode; ChaCha20-Poly1305 where AES acceleration is unavailable. Non-AEAD patterns shall use AES-CTR with HMAC-SHA-256.
- Approved Asymmetric Algorithms: ECDSA P-256 / P-384 or Ed25519 / Ed448 for signatures; RSA 3072 minimum for new deployments. ECDHE (P-256 / P-384) preferred for key exchange.
- Approved Hash Functions: SHA-256 / 384 (primary); SHA-512 for specialized use. MD5 and SHA-1 are prohibited.
- Randomness Requirements: Use NIST-approved DRBGs seeded from high-entropy sources; forbid non-CSPRNG PRNGs for cryptographic operations.
- Parameter Registry: Maintain an enterprise registry of approved algorithms, parameters, and deprecation timelines under change control.

6.2 Transport Security Profiles

- TLS Version Enforcement: TLS 1.3 everywhere feasible; TLS 1.2 allowed only by exception with restricted cipher suites; disable TLS 1.0 / 1.1.
- Cipher Suites: Limit to TLS_AES_128_GCM_SHA256, TLS_AES_256_GCM_SHA384, and TLS_CHACHA20_POLY1305_SHA256.

Obsolete and withdrawn documents should not be used; please use replacements.

- Authentication: Require mTLS for service-to-service and administrative channels; enforce strict hostname/SAN validation; enable OCSP stapling.
- HSTS Enforcement: Apply HTTP Strict Transport Security for public endpoints.
- Perfect Forward Secrecy: All TLS connections shall use ephemeral key exchange.

6.3 PKI & Certificate Lifecycle Management

- Implement an offline root CA with policy-bound intermediate CAs per trust domain.
- Use ACME or equivalent protocols for certificate issuance and renewal; enforce ≤ 90 -day validity for internal leaf certificates.
- Maintain available and monitored OCSP / CRL endpoints; auto-revoke on key compromise.
- Publish to Certificate Transparency logs for public endpoints; log all issuance and revocation events.
- Define and enforce validation rules for usage constraints and SAN requirements.

6.4 Key Management Operations

- Key Generation: Perform exclusively within HSM/KMS boundaries using approved DRBGs; prohibit plaintext export of private keys.
- Key Rotation: Rotate DEKs every ≤ 90 days; KEKs every ≤ 12 months; immediately upon compromise or policy trigger.
- Dual Control & Split Knowledge: Enforce M-of-N approval for KEK creation/import and sensitive operations with immutable evidence records.
- Key Destruction: Use cryptographic erasure for decommissioned datasets by destroying KEKs; log and attest to destruction.
- Key Inventory: Maintain an authoritative inventory of all keys with ownership, purpose, cryptoperiod, and status.
- Ceremony Evidence: Key ceremonies and sensitive operations shall produce signed artifacts (attestations, witness logs, M-of-N approvals, HSM transcripts) stored immutably with unique Evidence Pack IDs.

6.5 Data Encryption Patterns

- At Rest: Apply full-disk or device encryption for endpoints and servers; use database/table/column encryption for structured data; object-level encryption for unstructured storage.
- In Transit: Require TLS/mTLS for all communications; encrypt replication and backup transfers.
- In Use: Use trusted-execution environments (TEEs) or hardware enclaves for sensitive operations; implement side-channel-resistant code.
- Envelope Encryption: Encrypt data using DEKs wrapped by KEKs in separate trust domains.
- Backup Encryption: Apply distinct KEKs for backups and maintain independent trust boundaries.

6.6 Secrets Management

- Prohibit Hard-coded Secrets: Implement pre-commit and CI/CD scanning to block commits containing secrets.

Obsolete and withdrawn documents should not be used; please use replacements.

- Dynamic Credentials: Issue short-lived credentials bound to workload identity; enforce automated rotation and revocation.
- Access Control: Apply least-privilege policies for secret retrieval; maintain immutable audit trails.
- Secrets Inventory: Track all issued secrets with metadata, rotation status, and last access timestamp.

6.7 Cryptographic Agility & PQC Readiness

- Maintain a current inventory of supported algorithms and parameters for all systems.
- Hybrid KEM and signature pilots **Should** be executed in non-production environments when justified by data longevity, threat model, or platform readiness.
- Define migration triggers based on NIST standardization milestones, vendor readiness, and performance budgets.
- Document rollback criteria and reversion steps if required.
- Maintain compatibility matrices to preserve interoperability during transitions.

6.8 Module Validation & Library Hygiene

- FIPS 140-3 Validation: Deploy validated modules where required; track validation certificates and versions.
- Library Management: Maintain an approved library list; apply patches regularly; forbid custom cryptography.
- Self-Tests: Enable known-answer tests (KATs) and startup self-tests; trigger alerts and quarantine on failure.
- Compiler & Language Hardening: Use memory-safe languages where practical; apply compiler hardening flags to crypto-adjacent code.

6.9 Observability, KPIs & SLOs

- KPIs: Track encryption coverage, key-rotation adherence, certificate-expiry incidents, secrets-in-code findings, mTLS coverage.
- SLOs: Certificate issuance/renewal ≤ 5 minutes; key-compromise detection-to-revocation ≤ 15 minutes; DEK rotation success $\geq 99\%$.
- Telemetry: Emit signed, immutable logs for all cryptographic operations and integrate with SIEM for anomaly detection.
- mTLS Coverage SLOs: $\geq 98\%$ for east-west and administrative channels; 100 % TLS for north-south.
- Secrets Lifetime SLOs: Production secret TTL ≤ 24 hours; rotation on compromise ≤ 15 minutes; 0 hard-coded secrets in protected branches.
- Revocation & Status SLOs: OCSP/CRL availability $\geq 99.9\%$; publication latency ≤ 5 minutes from event.
- Time Synchronization SLOs: Clock skew across CEK components ≤ 1 second (P95); monitored and enforced.
- Entropy/Nonce SLOs: DRBG health checks at startup and hourly; nonce reuse = 0 per key; alert on reuse attempts.

Obsolete and withdrawn documents should not be used; please use replacements.

Evidence Pack

Evidence Must be collected for Section 6 technical specifications in EP-09.2 (Technical Specifications). Each output in 6.1 through 6.9 Must include at least one dated artifact that demonstrates implementation, enforcement, and the applicable measurement point. Evidence Must be version-controlled and retained according to organizational audit requirements.

Minimum evidence expectations for EP-09.2 include:

- Algorithm and parameter enforcement artifacts (6.1): Approved algorithm and parameter registry export, deprecation timeline record, and CI or CD gate results showing blocked disallowed algorithms or parameters.
- Transport profile enforcement artifacts (6.2): TLS and mutual TLS policy definitions, scanner output confirming protocol and cipher-suite conformance, and evidence of hostname and subject alternative name validation rules.
- PKI and certificate lifecycle artifacts (6.3): PKI topology diagram, certificate issuance and renewal automation configuration, renewal logs, and revocation records showing propagation timing.
- Key management operations artifacts (6.4): HSM or KMS policy snapshots, rotation job definitions and results, blocked plaintext export events, M-of-N approval records, and key ceremony artifacts where applicable.
- Encryption pattern artifacts (6.5): Configuration evidence for encryption at rest, in transit, and in use, including envelope-encryption implementation records and backup encryption separation proof.
- Secrets management artifacts (6.6): Secrets scanning outputs, dynamic secret issuance policy, rotation and revocation logs, and access audit trails demonstrating least-privilege retrieval.
- Agility and post-quantum readiness artifacts (6.7): Capability inventory, compatibility matrix, migration trigger record, rollback procedure, and pilot evidence only where executed.
- Module validation and library hygiene artifacts (6.8): Approved crypto library allowlist, validation certificate references where required, patch records, and self-test and known-answer-test outputs with failure handling.
- Observability and SLO artifacts (6.9): Signed and tamper-evident audit telemetry samples, KPI and SLO dashboards, alert rules for downgrade attempts and key misuse, and metric snapshots demonstrating compliance with defined thresholds.

Entries in EP-09.2 Must link back to EP-09.1 (Requirements) to show prerequisite readiness and Must link forward to EP-09.5 (Verification and Validation) for test execution evidence, negative tests, and formal acceptance results.

Obsolete and withdrawn documents should not be used; please use replacements.



Practitioner Guidance:

Successful implementation requires continuous verification, evidence production, and strict enforcement of SLOs:

- Tie each change to evidence: Every TLS policy, key rotation, certificate issuance rule, and secret policy shall carry an Evidence Pack ID containing the IaC diff, validation output, and pass/fail artifact.
- Prove SLOs same day: Immediately verify § 6.9 metrics post-deployment: 100 % TLS north-south; ≥ 98 % mTLS east-west/admin; OCSP/CRL uptime ≥ 99.9 %; revocation ≤ 5 minutes; Secrets TTL ≤ 24 hours; rotation ≤ 15 minutes; clock skew ≤ 1 second (P95); DRBG health pass startup + hourly; nonce reuse = 0.
- Block merges on drift: CI pipelines shall fail builds that introduce disallowed algorithms, expired certificates, or hard-coded secrets; attach the rejected artifact to the Evidence Pack.
- Continuously instrument metrics: Feed key rotation, certificate renewal, and entropy health into central dashboards to demonstrate ongoing conformance during annual Evidence Pack reviews.



Quick Win Playbook:

Title: Automated Certificate Lifecycle and Key Boundary Enforcement

Objective: Eliminate certificate-expiration risk and prevent private-key exposure by enforcing automated certificate issuance and renewal, short certificate validity, and hardware-bound key generation with verifiable evidence recorded in EP-09.2 and EP-09.5.

Target: Eliminate manual certificate renewal and uncontrolled private-key export by enforcing automated ACME issuance, 90-day certificate validity, and HSM-backed key generation (§ 6.3, § 6.4).

Component / System: Enterprise PKI (internal certificate authorities, ACME service) and HSM and KMS key-generation endpoints.

Protects: Certificate integrity, service-to-service trust, and private-key confidentiality across production and administrative channels.

Stops / Detects: Expired or mis-issued certificates causing outages; plaintext key backups on developer systems; unauthorized certificate issuance; weak entropy during key generation.

Obsolete and withdrawn documents should not be used; please use replacements.

	<p>Action: Deploy ACME-compatible internal issuance with ≤ 90-day leaf-certificate policy.</p> <ol style="list-style-type: none"> 1. Enforce key generation inside HSM and KMS boundaries using approved deterministic random bit generators; deny plaintext export. 2. Integrate renewal automation with CI/CD pipelines and telemetry to record each issuance event. 3. Validate online certificate status protocol and certificate revocation list availability and log each issuance and revocation event. <p>Test: Attempt manual certificate signing request with local key export to confirm deny; attempt automated ACME issuance with HSM-backed key path to confirm allow and record.</p> <p>Proof: ACME configuration diff + issuance log + HSM transaction record + OCSP status report recorded in EP-09.2 (test results recorded in EP-09.5).</p> <p>Metric: 100 % of internal leaf certificates auto-renew within ≤ 5 minutes; 0 plaintext key exports; OCSP availability ≥ 99.9 %; revocation publication ≤ 5 minutes; all issuance events logged and verified.</p> <p>Rollback: Disable ACME automation and revert to prior PKI workflow (time-boxed). Retain the prior certificate chain and logs as superseded evidence in the active Evidence Pack.</p>
--	---

Section 7. Cybersecurity Core Principles

The following ISAUnited Cybersecurity Core Principles are foundational to the design, implementation, and ongoing management of secure Cryptography, Encryption & Key Management (CEK) architectures. Each principle guides architectural decisions, technical controls, and operational practices to ensure cryptographic systems are resilient, measurable, and engineered to withstand real-world threats.

Purpose and Function:

Security principles provide more than technical direction—they embed discipline, clarity, and foresight into every recommendation. By grounding technical specifications and implementation strategies in well-defined principles, ISAUnited ensures that sub-standards do not merely respond to threats tactically but are built to withstand architectural and systemic risk over time.

Obsolete and withdrawn documents should not be used; please use replacements.


Table I-2. Principles and CEK-Domain Applicability:

Principle Name	Code	Applicability to Cryptography, Encryption & Key Management
Least Privilege	ISAU-RP-01	Access to cryptographic keys, HSM/KMS functions, and secrets-management operations shall be restricted to the minimum personnel, services, and processes required for authorized activity.
Zero Trust	ISAU-RP-02	All cryptographic operations, key usages, and certificate validations require continuous verification of identity, integrity, and trust—regardless of network location or system state.
Complete Mediation	ISAU-RP-03	Every cryptographic request (key unwrap, signature generation, decryption) shall be validated, authorized, and logged; no operation relies on prior trust without re-evaluation.
Defense in Depth	ISAU-RP-04	Multiple layers of cryptographic controls (e.g., transport encryption, application-level encryption, HSM boundary protection, dual control) prevent any single point of compromise.
Secure by Design	ISAU-RP-05	Cryptographic protections are integrated at system design inception, with algorithms, key lengths, and protocols selected based on security requirements and lifecycle planning.
Minimize Attack Surface	ISAU-RP-06	Continuously reduce exposed cryptographic interfaces, deprecated algorithms, unused certificates, and over-privileged crypto-API access through governance and library hygiene.
Resilience & Recovery	ISAU-RP-14	Engineer cryptographic services for redundancy and rapid recovery, including geo-redundant HSM clusters, backup key escrow procedures, and failover for CRL/OCSP services to sustain availability during outages or attacks.
Evidence Production	ISAU-RP-15	Maintain immutable, signed audit logs for all cryptographic operations, key events, and administrative actions to prove provenance and support forensics, compliance, and assurance testing.
Make Compromise Detection Easier	ISAU-RP-16	Enhance monitoring and telemetry for cryptographic operations to detect abuse, misuse, and drift. Integrate alerts from HSM/KMS, PKI, and secrets platforms with SIEM for early breach visibility.

Obsolete and withdrawn documents should not be used; please use replacements.

Principle Name	Code	Applicability to Cryptography, Encryption & Key Management
Cryptographic Agility	ISAU-RP-17	Architect systems to support algorithm and parameter changes without major redesign, enabling planned migration to post-quantum or updated cryptographic standards.
Protect Confidentiality	ISAU-RP-18	Encrypt all sensitive data in transit, at rest, and in use using approved algorithms; store keys in secure hardware or services under strict access controls.
Protect Integrity	ISAU-RP-19	Use authenticated encryption and digital signatures to detect and prevent unauthorized modification of data, code, and cryptographic material.
Protect Availability	ISAU-RP-20	Design CEK systems for high availability, redundancy, and rapid recovery, ensuring that cryptographic services remain operational during infrastructure failures or cyber events.

Note: Organizations may include a matrix mapping each selected principle to its associated technical outputs or control mappings, further demonstrating traceability.

	<p>Practitioner Guidance:</p> <p>These principles Must be integrated into CEK architectural decisions and technical implementations. They form the engineering foundation for all sub-standards developed under this Parent Standard and ensure cryptographic designs remain defensible by design, not only compliant. Apply these principles during design reviews, change approvals, and Evidence Pack audits to maintain resilient, provably trustworthy cryptographic services.</p>
---	--

Section 8. Foundational Standards Alignment

Cryptography, Encryption & Key Management (CEK) aligns with globally recognized foundational standards to support interoperability, regulatory compliance, and consistent cryptographic risk management. ISAUnted Defensible Standards provide engineering depth and operational rigor. Foundational alignment preserves auditability, industry acceptance, and integration into existing security and compliance programs.

Obsolete and withdrawn documents should not be used; please use replacements.

Purpose and Function

This section defines the authoritative baseline from which CEK sub-standards derive cryptographic assurance. Alignment to NIST and ISO publications provides reference architecture for key lifecycle governance, algorithm selection, randomness validation, and cryptographic module assurance. These baselines support consistent measurement of cryptographic maturity across enterprise environments.

Table I-3. Applicable Foundational Standards:

Framework	Standard ID	Reference Focus
NIST	SP 800-57 Pt 1–3	Lifecycle management of cryptographic keys: generation, distribution, rotation, escrow, destruction, and governance.
NIST	SP 800-52 Rev. 2	TLS implementation guidance: protocol profiles, cipher-suite policy, certificate validation, interoperability, and revocation handling.
NIST	SP 800-56 A/B	Key establishment schemes: approved key agreement and key transport, including ECDH and ECDHE.
NIST	SP 800-130	Framework for cryptographic key management systems: architecture and operational requirements.
NIST	SP 800-38 Series	Block-cipher modes of operation: GCM, CTR, XTS, and authenticated-encryption constructs.
NIST	SP 800-90 A/B/C	Deterministic random bit generators: entropy-source validation, health testing, and seeding requirements.
NIST	SP 800-175B	Guidance for applying cryptographic standards: algorithm selection, parameter governance, and lifecycle control.
NIST	FIPS 140-3	Cryptographic module security requirements and validation process: boundary definition and certification.
NIST	FIPS 186-5	

Obsolete and withdrawn documents should not be used; please use replacements.

Framework	Standard ID	Reference Focus
		Digital Signature Standard: approved signature algorithms and parameter sets.
ISO/IEC	19790	Cryptographic module security requirements aligned to FIPS 140-3 for international applicability.
ISO/IEC	27040	Storage security: encryption and key management for data at rest.
ISO/IEC	29192	Lightweight cryptography for constrained and embedded environments.
ISO/IEC	18031	Random bit generation: entropy modeling, statistical testing, and RNG alignment.

NOTE: As detailed sub-standards are developed under this parent standard, specific references to NIST and ISO will be incorporated to provide control-level alignment and practical implementation guidance for CEK practitioners.

NOTE: ISAUnited Charter Adoption of Foundational Standards.

Per the ISAUnited Charter, the institute formally adopts the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) and the National Institute of Standards and Technology (NIST) as its foundational standards bodies, consistent with their public encouragement of organizational adoption. Parent Standards align with ISO/IEC and NIST for architectural grounding and auditability, and this alignment cascades down to Sub-Standards as invariant, minimum requirements that may be tightened but not weakened. ISAUnited does not restate or speak on behalf of ISO/IEC or NIST; practitioners shall consult the official publications and terminology of these organizations, verify scope and version currency against the latest materials, and implement controls in a manner consistent with ISAUnited security invariants and the requirements of this standard.

Sub-Standard Expectations

Sub-standards developed under ISAU-DS-CEK-1000 shall demonstrate direct lineage to one or more of the foundational NIST or ISO publications listed above. Each sub-standard shall extend those baseline expectations into domain-specific engineering controls, defining measurable outputs, validation methods, and required evidence

Obsolete and withdrawn documents should not be used; please use replacements.

artifacts. Any intentional divergence from a cited clause or model shall be fully justified through a documented compensating control, mapped citation, and associated Evidence Pack record to preserve architectural integrity and audit traceability.


Evidence Pack

Evidence for Section 8 foundational standards alignment is recorded in EP-09.3 (Foundational Standards). This Evidence Pack section captures the clause-level mappings that anchor CEK requirements, technical specifications, and validation activities to adopted NIST and ISO/IEC baselines. It also preserves revision history, allowing reviewers to confirm that mappings remain current as standards evolve.

Minimum evidence expectations for EP-09.3 include:

- Clause-level mapping sheet: A table mapping CEK sections and key outputs (for example, key rotation, certificate lifecycle, randomness assurance) to specific NIST or ISO/IEC clause references.
- Citation snapshots: Extracted references to the applicable sections of NIST and ISO/IEC publications used in the mapping, including revision identifiers and publication dates.
- Standards selection rationale: Short justification explaining why each baseline standard applies to the CEK scope and how it supports interoperability, assurance, and audit.
- Deviation and equivalence records: Documented cases where implementations diverge from a cited clause, including compensating controls and planned review dates.
- Change history: Version-controlled records showing when mappings were added or updated and what triggered the change.

Entries in EP-09.3 link forward to implementation evidence in EP-09.2 (Technical Specifications) and to test evidence in EP-09.5 (Verification and Validation) when foundational mappings are exercised through validation activities.

	<p>Practitioner Guidance:</p> <p>Practitioners use these mappings to demonstrate how CEK implementations inherit assurance from globally recognized authorities.</p> <ul style="list-style-type: none"> • Maintain clause-level mappings that connect CEK requirements and outputs to the referenced NIST and ISO publications. • When mappings change, update the citation and retain the revision history in EP-09.3 to support audit sampling and peer review.
---	--

Obsolete and withdrawn documents should not be used; please use replacements.

- | | |
|--|--|
| | <ul style="list-style-type: none"> Where multiple clauses apply, document the selected clause and rationale once and reuse it across implementations to reduce drift. |
|--|--|

Section 9. Security Controls

This section identifies the control families and external frameworks that the Cryptography, Encryption & Key Management (CEK) Parent Standard directly supports or enforces. These control mappings link ISAUnited's architectural requirements to recognized cybersecurity frameworks, enabling measurable validation, audit traceability, and consistency of implementation across enterprise environments.

This alignment ensures that cryptographic implementation is verifiable not only within ISAUnited's defensible framework but also against industry control catalogs used for compliance assessments.

Purpose and Function

Security controls translate the architectural intent of this standard into actionable, measurable safeguards. They provide the tactical foundation to enforce confidentiality, integrity, availability, authentication, authorization, and auditability across CEK domains.

By mapping CEK technical specifications to frameworks such as the CSA Cloud Controls Matrix (CCM), CIS Controls v8, and OWASP ASVS, ISAUnited achieves:

- Clear alignment with recognized regulatory and assurance practices.
- Interoperability across enterprise and cloud environments.
- Consistency and reusability of control logic within CEK sub-standards, facilitating structured implementation and peer-review validation.

These mappings allow engineers, assessors, and auditors to measure and demonstrate the defensibility of CEK implementations.

Implementation Guidance

When defining CEK sub-standards or producing implementation evidence:

- Reference at least three technical controls from one or more authoritative cybersecurity frameworks.

Obsolete and withdrawn documents should not be used; please use replacements.

- Provide the framework acronym, control identifier, and concise control description.
- Align each selected control with the corresponding CEK technical specification (§ 6) and ISAUnited Core Principles (§ 7).
- Select implementation-level controls rather than policy statements to ensure measurable outcomes.

Table I-4. Control Mappings for Cryptography, Encryption & Key Management:

Framework	Control ID	Control Name / Description and Reference Focus
CSA CCM v4	CEK-01	Encryption and Key Management Policy and Procedures - Establish governance for approved algorithms, parameter baselines, key ownership, cryptoperiods, and deprecation timelines, enabling consistent enforcement and auditability across environments.
CSA CCM v4	CEK-03	Data Encryption - Require encryption for sensitive data states and prevent plaintext exposure by standardizing encryption at rest and in transit with measurable coverage targets.
CSA CCM v4	CEK-10	Key Generation - Ensure keys are generated using approved randomness sources and within controlled boundaries (e.g., HSM or KMS), reducing predictable key risk and unauthorized creation of key material.
CSA CCM v4	CEK-12	Key Rotation - Limit exposure windows by enforcing rotation schedules and cryptoperiod adherence, enabling measurable rotation success rates and automated renewal workflows.
CSA CCM v4	CEK-13	Key Revocation - Support rapid response to compromise by requiring revocation workflows, propagation timing targets, and evidence that revoked keys and certificates are rejected across clients and services.
CIS Controls v8	3.11	Encrypt Sensitive Data at Rest - Protect stored sensitive information by enforcing strong encryption and ensuring keys are protected and managed separately from encrypted data.
CIS Controls v8	3.10	Encrypt Sensitive Data in Transit - Prevent interception and downgrade attacks by enforcing secure transport protocols, a validated cipher suite policy, and consistent service identity verification.
CIS Controls v8	8.9	

Obsolete and withdrawn documents should not be used; please use replacements.

Framework	Control ID	Control Name / Description and Reference Focus
		Centralize Audit Logs - Enable forensic reconstruction and misuse detection by centralizing cryptographic event logs (key use, issuance, revocation, secret access) with integrity protections and retention.
OWASP ASVS v4	V6.2	Algorithms - Prevent weak cryptography by requiring modern algorithm choices, correct key lengths, and removal of deprecated ciphers across application and service implementations.
OWASP ASVS v4	V6.3	Random Values - Prevent nonce reuse, predictable keys, and token forgery by enforcing cryptographically secure randomness, entropy health validation, and correct use of nonces and salts.
OWASP ASVS v4	V6.4	Secret Management - Reduce credential leakage by prohibiting hard-coded secrets, enforcing short-lived credentials, and requiring audit trails and controlled access paths for secret retrieval.

NOTE: NIST and ISO are Foundational Standards in §8. Use CSA/CIS/OWASP here in §9 for control implementation. Adversary-technique mapping (e.g., ATT&CK) belongs in §12 and sub-standards' test plans.

NOTE: Use of External Control Frameworks.

ISAUnited maps to external control frameworks to provide alignment and traceability, but does not speak on behalf of those organizations. Practitioners shall consult and follow the official practices, recommendations, and implementation guidance of the Center for Internet Security (CIS), the Cloud Security Alliance (CSA), and the Open Worldwide Application Security Project (OWASP) when applying controls. Always verify control identifiers, scope, and version currency against the publishers' latest materials. Where wording differs, use the framework's official documentation while maintaining consistency with ISAUnited security invariants and this standard's requirements.

Sub-Standard Expectations

Sub-standards developed under ISAU-DS-CEK-1000 shall incorporate control mappings relevant to their technical scope. Each sub-standard shall extend these control references into measurable validation procedures, implementation guidance, and operational assurance criteria. Any deviation or exclusion of a referenced control

Obsolete and withdrawn documents should not be used; please use replacements.

must be documented with justification, compensating measures, and cross-reference to the related Evidence Pack to maintain transparency and defensibility.

Evidence Pack

Evidence for Section 9 control mappings is recorded in EP-09.4 (Control Mappings). This Evidence Pack area preserves the external control lineage for CEK and shows how each mapped control is applied through CEK technical specifications and verified through testing. The goal is not to restate frameworks, but to document traceable alignment that supports audit sampling, peer review, and consistent implementation across teams.

Minimum evidence expectations for EP-09.4 include:

- Control mapping sheet: A maintained table that maps each external control in Table I-4 to the related CEK technical specification (§ 6) and Core Principle (§ 7). Include control scope notes, ownership, and last review date.
- Implementation linkage: A short reference for how the control is enforced in practice (policy-as-code rule name, configuration policy identifier, or repository path). Link to the corresponding implementation artifact stored in EP-09.2.
- Clause selection rationale: A brief justification for why each control was selected and what risk it addresses in CEK context (certificate outage, key compromise, downgrade exposure, secrets sprawl, entropy failure).
- Cross-framework equivalence notes: Where two frameworks express the same intent, record a single equivalence note to prevent duplication and mapping drift.
- Versioning and change history: Record the framework version used for each control mapping and capture updates when control IDs, wording, or scope changes. Maintain revision history with date, editor, and change summary.
- Exception records: If a mapped control is not applicable to a specific environment, document the exception, compensating measure, and planned review date.
- Evidence pointers: For each control, include the pointer to the proof location in EP-09.5 where validation activities demonstrate enforcement (test IDs, scan results, negative test outcomes).

EP-09.4 entries link backward to EP-09.3 (Foundational Standards) when foundational baselines drive control interpretation, and link forward to EP-09.5 (Verification and Validation) when mapped controls are exercised through tests and operational validation.

Obsolete and withdrawn documents should not be used; please use replacements.



Practitioner Guidance:

For Security Architects and Engineers: Treat every mapped control in Table I-4 as a verifiable implementation checkpoint, not a documentation artifact. Confirm that each control can produce measurable evidence—configuration diff, log extract, or automation output—tagged to an active Evidence Pack ID. When creating sub-standards, embed these control IDs directly into CI/CD validation rules or IaC policy gates to prevent drift.

For Reviewers and Auditors: Verify that CEK implementations demonstrate both *alignment* and *evidence*. Each control must link to at least one cryptographic specification (§ 6) and one Core Principle (§ 7). During peer review, request the clause citation (e.g., NIST SP 800-57 § 5.3) and confirm that the Evidence Pack contains proof of enforcement, not only policy reference.

For Program Managers and Compliance Leads: Integrate these mappings into enterprise audit plans to replace checklist verification with evidence-driven validation. When a framework version updates (e.g., CIS v8 to v9), require sub-standard maintainers to update the Table I-4 citation in the same change request, preserving the “map-once, validate-always” discipline.

Outcome: Continuous mapping between CEK technical specifications, Core Principles, and external frameworks creates an auditable chain of custody for assurance. Practitioners who maintain this traceability can demonstrate to regulators and assessors that encryption, key management, and secrets governance are not only compliant but also defensible by design.

Section 10. Engineering Discipline

This section defines the architectural thinking, rigorous engineering processes, and disciplined operational behaviors required to implement the Cryptography, Encryption & Key Management (ISAU-DS-CEK-1000) standard.

ISAUnited’s Defensible Standards treat cryptography as an engineered system—grounded in systems thinking, lifecycle control, and Verification & Validation (V&V)—that produces measurable, auditable, and defensible outcomes across encryption, key management, certificate automation, secrets governance, and post-quantum readiness.

10.1 Purpose & Function

Purpose. Establish a repeatable, auditable engineering discipline that integrates systems thinking, cryptographic lifecycle management, assurance testing, and measurable outcomes for all CEK architectures.

Obsolete and withdrawn documents should not be used; please use replacements.

Function in D10S. Parent Standards define the engineering invariants and expectations.

Sub-standards translate them into policy-as-code and control-as-code, along with validation tests and evidence artifacts, which are embedded in delivery and operational pipelines.

10.2 Systems Thinking

Goal: Make the cryptographic system legible end-to-end—components, trust boundaries, key flows, dependencies, and safeguards—so that engineering and assurance activities bind precisely where cryptographic risk exists.

10.2.1 System Definition & Boundaries

- Declare scope, stakeholders, and in/out-of-scope components (PKI hierarchy, HSM/KMS clusters, certificate automation, secrets platforms, encryption services, DRBG/entropy modules, and PQC pilots).
- Model trust zones and boundary crossings (workload to KMS, KMS to HSM, application to PKI CA, CI/CD to secrets vault).
- Define boundary invariants—for example: no plaintext key export, no unsigned certificates, MFA + short-lived tokens for admin planes, and zero fail-open cryptographic operations.

10.2.2 Interfaces & CEK Contracts

- Maintain Interface Control Documents (ICDs) for key generation, wrapping/unwrapping, certificate issuance, secret retrieval, and telemetry exchange.
- For each interface, specify: identity type (human vs service), privileges, supported algorithm set, key length, nonce/IV requirements, latency SLOs, retention, time-sync tolerance, fail-closed behavior, and mandatory audit fields (e.g., key_id, cert_id, op_id, evidence_pack_id).

10.2.3 Dependencies & Emergent Behavior

- Map shared dependencies (entropy sources, directory/identity services, CI/CD systems, logging and SIEM, network orchestration).
- Identify emergent risk from composition (for example, entropy degradation leading to predictable keys or nonces; CA mis-issuance combined with weak revocation leading to trust collapse).

10.2.4 Failure Modes & Safeguards

- Document likely failure modes (entropy source failure, HSM quorum loss, OCSP timeout, certificate renewal drift, secrets scanner bypass).
- Engineer safeguards (dual-control thresholds, certificate-expiry alerts, entropy-health probes, auto-revocation, tamper-evident logging).

Required Artifacts (min): CEK context diagram with trust boundaries; key-flow map; PKI/HSM ICDs; invariants register.

Obsolete and withdrawn documents should not be used; please use replacements.

10.3 Critical Thinking

Goal: Replace assumption-based configuration with explicit, reviewable reasoning that withstands adversarial analysis and audit scrutiny.

10.3.1 Decision Discipline

- Maintain **Architecture Decision Records (ADRs)**: problem to options to constraints/assumptions to trade-offs to decision to invariants to test/evidence plan (who / when / how measured).
- Require ADR linkage to relevant ISAU-RPs (01–20), NIST/ISO clauses, and Evidence Pack IDs.

10.3.2 Engineering Prompts

- **Boundaries:** Where do key and trust boundaries exist and why? Which zones have explicit dual-control contracts?
- **Interfaces:** What invariants must always hold (auth, integrity, algorithm class)? How are they tested?
- **Adversary Pressure:** Which realistic attacks threaten confidentiality, key lifecycle, or cryptographic agility, and how are they mitigated or detected?
- **Evidence:** What objective signals prove the control works today and after change (key-rotation success %, certificate-renewal latency, entropy-health metrics)?
- **Failure:** When failure occurs, does it fail safe (e.g., auto-revoke, deny)? What is the operator response path?

Required Artifacts (min): ADRs; assumptions/constraints log; evidence plan per decision.

10.4 Domain-Wide Engineering Expectations

Secure System Design

- Define CEK trust boundaries (HSM / KMS / PKI / Secrets Platform / PQC Pilot).
- Validate boundaries and trust relationships via architecture reviews using § 10.2 artifacts.
- Apply least-privilege, separation of duties, and redundancy principles consistent with confidentiality, integrity, and availability objectives.

Implementation Philosophy — “Built-in, not Bolted-on”.

- Embed encryption, key management, and certificate automation during system design.
- Express controls as policy-as-code or control-as-code (e.g., “No plaintext key exports,” “All certificates \leq 90 days validity,” “Auto-revoke on compromise”).

Lifecycle Integration

- Embed CEK controls into design, build, deployment, and operational pipelines.
- Maintain version-controlled repositories requiring ADR and Evidence Pack updates on each change.

Verification Rigor (V&V)

Obsolete and withdrawn documents should not be used; please use replacements.

- Combine automated checks (key-rotation success, cert renewal latency, OCSP uptime, entropy health) with targeted red/purple tests and fault injection.
- Require continuous validation in pipelines and runtime schedules tied to § 6 SLOs.

Operational Discipline

- Monitor for cryptographic drift, expired certificates, deprecated algorithms, or unauthorized key usage.
- Maintain runbooks for key compromise, certificate mis-issuance, HSM failure, and algorithm deprecation; log outcomes to Evidence Pack.

10.5 Engineering Implementation Expectations

- **Cryptographic Controls as Code.** Store policies (e.g., TLS profiles, key rotation rules, certificate lifetimes, entropy monitors) as signed artifacts in version control.
- **Structured Enforcement Pipelines.** Automate validation and promotion with CI/CD gates, rollback plans, and peer-review records linked to Evidence Pack IDs.
- **Explicit Coverage Mapping.** Maintain dashboards for encryption coverage (by data state/platform), key rotation compliance, and certificate expiry metrics.
- **Automated Testing & Negative Validation.** Run simulated key rotations, certificate revocations, and entropy failure scenarios before production; verify fail-closed behaviors and rollback success.
- **Traceable Architecture Decisions.** Link each change (ADR ID, Test ID, Evidence Pack ID) for audit continuity and peer review.

Required Artifacts (min): policy/control-as-code repos; CI/CD gates; trust-boundary diagrams; rotation/renewal metrics; automated test logs; evidence ledger (see § 12).

10.6 Sub-Standard Alignment (Inheritance Rules)

Sub-standards operationalize this engineering discipline with CEK-specific detail. Each sub-standard documents how controls are expressed as code, how validation is performed, and where evidence is recorded within the EP-09 structure. Sub-Standards must operationalize this discipline with CEK-specific detail.

- ISAU-DS-CEK-1020 (Enterprise PKI Automation): Maintain certificate issuance and renewal as code, validate renewal latency, and retain peer-review records and test outputs.
- ISAU-DS-CEK-1030 (Key Management Operations and Ceremonies): Enforce dual-control workflows through HSM and KMS policy configuration, record M-of-N approvals, and preserve ceremony artifacts.
- ISAU-DS-CEK-1040 (TLS and mTLS Profiles for Services and APIs): Manage protocol and cipher-suite profiles as code, validate

Obsolete and withdrawn documents should not be used; please use replacements.

conformance through automated scanning, simulate downgrade attempts, and retain test logs.


- ISAU-DS-CEK-1050 (Secrets Management and Dynamic Credentials): Automate issuance, rotation, and revocation validation, verify TTL compliance, and retain scanner outputs and access audit trails.
- ISAU-DS-CEK-1060 (Post-Quantum Readiness and Hybrid Deployments): Record hybrid pilot evidence only when executed. Capture performance and interoperability results, along with rollback exercises, and where foundational alignment is cited.

10.7 Evidence & V&V (What Proves It Works)

Establish a CEK Evidence Pack for each environment containing:

- Design Evidence: Architecture diagrams, trust-boundary maps, PKI/HSM ICDs, invariants register, ADRs.
- Build Evidence: Key rotation rules, certificate automation scripts, policy-as-code repos, CI/CD validation results.
- Operate Evidence: Certificate renewal logs, entropy health metrics, HSM audit trails, OCSP/CRL availability, and rotation SLO reports.
- Challenge Evidence: Red/purple team key-theft tests, entropy degradation simulations, revocation and rollback drills.

Each control defines objective pass/fail criteria, test frequency, responsible owner, and retention period.

	<p>Practitioner Guidance:</p> <p>For cryptographic engineers and architects, treat CEK architecture as a living system. Begin design work by identifying trust boundaries, invariants, and measurable success criteria. Maintain ADRs that capture trade-offs and test plans.</p> <p>For implementation teams, express CEK logic as control-as-code with verifiable outputs. Favor fail-closed behavior, measurable telemetry, and auditable change records. Treat automation pipelines as test harnesses.</p> <p>For validation and operations, apply V and V practices aligned to § 12. Execute negative tests, record results, and tie evidence to the invariants register and EP-09 locations.</p> <p>For reviewers and leaders, look for traceability. Decisions map to assumptions, invariants, test IDs, and Evidence Pack locations. An engineering discipline becomes visible when design intent, operation, and audit evidence align without manual reconstruction.</p>
---	--

Obsolete and withdrawn documents should not be used; please use replacements.

Section 11. Associate Sub-Standards Mapping

Purpose of Sub-Standards

ISAUnited Defensible Sub-Standards are detailed, domain-specific extensions of the Cryptography, Encryption & Key Management Parent Standard (ISAU-DS-CEK-1000). Each Sub-Standard delivers:

- Granular technical guidance tailored to specialized CEK domains.
- Actionable engineering strategies that translate architectural intent into operational controls.
- Defined verification methodologies ensuring outputs are measurable, testable, and auditable.
- Alignment with the Parent Standard's § 6 technical outputs, § 7 principles, and Table I-3 foundational standards.

Sub-Standards translate architectural direction into the detailed technical precision required for robust engineering, continuous validation, and defensible auditing across PKI, TLS/mTLS, secrets management, cryptographic agility, and encryption patterns.

Scope and Focus of CEK Sub-Standards

PKI Architecture & Certificate Lifecycle Automation

Example Sub-standard: ISAU-DS-CEK-1020

- Defines PKI hierarchy, root/intermediate CA roles, and certificate policy OIDs.
- Prescribes ACME-based automation for issuance and renewal (\leq 90-day validity).
- Requires OCSP/CRL availability, certificate-transparency logging, and automated revocation workflows.
- Enforces continuous trust-chain validation.

TLS/mTLS Profiles for Services, APIs & Admin Channels

Example Sub-standard: ISAU-DS-CEK-1040

- Establishes approved TLS versions, cipher suites, and validation rules.
- Requires mTLS for service-to-service and administrative access.
- Enforces PFS and strict hostname/SAN validation.
- Integrates automated TLS configuration testing into CI/CD pipelines.

Key Management Operations & Ceremonies

Example Sub-standard: ISAU-DS-CEK-1030

Obsolete and withdrawn documents should not be used; please use replacements.

- Details procedures for secure key generation, wrapping, rotation, and destruction.
- Requires dual control, split knowledge, and M-of-N approvals for KEK operations.
- Defines cryptographic erasure and attestation requirements.
- Establishes key inventory metadata and audit retention rules.

Secrets Management & Dynamic Credential Issuance

Example Sub-standard: ISAU-DS-CEK-1050

- Prohibits hard-coded secrets; enforces pre-commit and CI/CD scanning.
- Issues short-lived, identity-bound secrets with automated revocation.
- Mandates audit logging and telemetry for secret access.
- Integrates workload-identity providers with revocation and rotation mechanisms.

Post-Quantum Cryptography (PQC) Readiness & Hybrid Deployments

Example Sub-standard: ISAU-DS-CEK-1060

- Defines capability profiles and compatibility matrices.
- Pilots hybrid KEM and signature schemes for interoperability.
- Specifies migration triggers and rollback criteria aligned to risk.
- Conducts annual PQC readiness reviews per NIST PQC milestones.

Data Encryption Patterns (At Rest, In Transit, In Use)

Example Sub-standard: ISAU-DS-CEK-1070

- Establishes envelope-encryption patterns by data classification.
- Segregates DEKs/KEKs by domain and trust tier.
- Requires independent KEKs for backup encryption.
- Evaluates deterministic and format-preserving encryption for limited cases with documented trade-offs.

Cryptographic Module Validation & Library Hygiene

Example Sub-standard: ISAU-DS-CEK-1080

- Requires FIPS 140-3 validation where mandated.
- Maintains approved cryptographic-library inventory and patch schedule.
- Enforces deprecation of legacy algorithms and modules.
- Validates self-tests and known-answer tests (KATs) during startup and runtime.

Table I-5. Example Sub-Standards:

Sub-Standard ID	Sub-Standard Name	Focus Area
ISAU-DS-CEK-1020	Enterprise PKI Architecture & Automated Certificate Lifecycle	PKI & Certificate Management
ISAU-DS-CEK-1040		Transport Security

Obsolete and withdrawn documents should not be used; please use replacements.

Sub-Standard ID	Sub-Standard Name	Focus Area
	TLS/mTLS Profiles for Services, APIs, and Admin Channels	
ISAU-DS-CEK-1030	Key Management Operations, Dual Control, and Key Ceremonies	Key Lifecycle Governance
ISAU-DS-CEK-1050	Secrets Management, Dynamic Credentials, and Telemetry	Secrets Governance
ISAU-DS-CEK-1060	PQC Readiness & Hybrid Deployment Strategies	Cryptographic Agility
ISAU-DS-CEK-1070	Data Encryption Patterns for Structured & Unstructured Data	Encryption Patterns
ISAU-DS-CEK-1080	Cryptographic Module Validation & Library Hygiene	Module Assurance

Note: Future CEK identifiers will continue the 1xxx series to maintain consistency with ISAUnted numbering.

Development and Approval Process

ISAUnted uses an open, peer-driven annual process to propose, review, and publish sub-standards:

- Open Season Submission: Contributors submit candidate sub-standards aligned with ISAU-DS-CEK-1000 objectives.
- Technical Peer Review: The Technical Fellow Society evaluates proposals for validity, accuracy, and applicability.
- Approval and Publication: Approved sub-standards receive formal versioning and publication as authoritative extensions of ISAU-DS-CEK-1000.
- Annual Review: All sub-standards undergo peer review each Open Season to incorporate advancements in NIST/ISO standards and cryptographic practice.

Sub-Standard Deliverables

Each CEK sub-standard includes the deliverables listed below, ensuring adoption remains measurable, testable, and traceable to the Parent Standard.

Obsolete and withdrawn documents should not be used; please use replacements.

- Inputs (Requirements): List the prerequisite conditions from Section 5 that the sub-standard depends on, including any readiness assumptions and boundary constraints.
- Outputs (Technical Specifications): Define concrete cryptographic behaviors and thresholds tied to Section 6, such as approved algorithms and parameters, certificate validity limits, rotation windows, revocation latency targets, entropy-health requirements, and signed-telemetry expectations.
- Verification and Validation: Define named tests and acceptance criteria tied to Section 12, including negative tests where applicable, such as downgrade rejection, revoked-certificate refusal, blocked key export at HSM boundaries, rotation success under load, and entropy-health failure handling.
- Evidence: Provide an artifact list and the Evidence Pack location for storage using the EP-09 structure. Implementation artifacts align to EP-09.2, foundational standards mappings align to EP-09.3, control mappings align to EP-09.4, and test evidence aligns to EP-09.5.
- Standards Mapping: Provide a traceability mapping from specification to NIST or ISO clause (Section 8) to control mapping (Section 9) to test identifier (Section 12) to Evidence Pack location.
- Interfaces and Boundaries: Define what the sub-standard enforces within CEK scope, including PKI services, HSM and KMS boundaries, certificate lifecycle automation, secrets platforms, and transport-security profiles. Separate delivery mechanics governed by Annex J from cryptographic engineering controls governed by this annex.

Section 12. Verification and Validation (Tests)

This section defines the structured evaluation methods that demonstrate Cryptography, Encryption, and Key Management (CEK) controls, architecture, and operations align with the intent of this Parent Standard. It mandates measurable, repeatable procedures so implementations are technically defensible and consistent with ISAUnited's engineering discipline.

Verification confirms capabilities were implemented in accordance with Section 5 Requirements (Inputs) and Section 6 Technical Specifications (Outputs).

Validation demonstrates that those capabilities perform under real-world conditions, withstand adversarial testing, and remain resilient as algorithms, environments, and threats evolve.

Obsolete and withdrawn documents should not be used; please use replacements.

Core Verification Activities

- Confirm all Section 6 outputs are deployed and configured in the target environment with coverage across declared scopes (on-premises, cloud, SaaS, edge, OT, and industrial control systems).
- Review hardened cryptographic baselines for PKI, HSM, and KMS, secrets platforms, and transport profiles; compare configurations to NIST SP 800-52, 800-57, 800-90, and ISO/IEC 19790 expectations.
- Verify integration paths (application to KMS to HSM, service to PKI CA, pipeline to secrets vault) have no fail-open states and preserve integrity, identity, and timing.
- Conduct peer review of architecture diagrams, trust-boundary maps, key-lifecycle workflows, and SLO logic to preserve traceability from requirement to output to test to evidence.

Core Validation Activities

- Execute adversary-informed testing, including TLS downgrade attempts, certificate-forgery simulations, key-leak injection, entropy-pool exhaustion, and revocation-path failures.
- Validate exploit resistance and recovery for cryptographic components, including HSM quorum loss, CA failover, OCSP interruption, and secrets-vault outage.
- Exercise cryptographic lifecycle events (key rotation, revocation, renewal) under load to confirm SLO adherence and automated recovery reliability.
- Assess algorithm-agility workflows through readiness review and rollback exercises. If hybrid deployments are piloted, validate interoperability and rollback behavior using controlled test environments.
- Measure performance against defined metrics, including:
 - Rotation Success Rate ($\geq 99\%$)
 - Certificate Renewal Latency (≤ 5 min)
 - Secrets TTL Compliance (≤ 24 h)
 - OCSP and CRL Availability ($\geq 99.9\%$)
 - Entropy Health Pass Rate ($= 100\%$ at startup and hourly)

Required Deliverables

All Verification and Validation efforts Must produce documented outputs that include:

1. Test Plans and Procedures — Scope, cases, test data, tools and simulators, positive and negative criteria, and safety constraints for production or regulated systems.
2. Validation Reports — Results, pass or fail, residual risk ranking, and re-test schedule.

Obsolete and withdrawn documents should not be used; please use replacements.

3. Evidence Artifacts — Logs, HSM transactions, TLS handshakes, certificate chains, entropy metrics, rotation and renewal records, screenshots, and change tickets.
4. Corrective Action Plans (CAPs) — Remediation steps, owners, deadlines, exceptions, and follow-up test IDs.

Common Pitfalls to Avoid

- Closing without proof – Marking key rotation, certificate renewal, or revocation as complete without attaching the validation artifact set (for example, renewal logs, handshake evidence, revocation confirmation, and SLO results) to EP-09.5.
- Testing success only – Validating only happy paths and skipping negative tests that prove fail-closed behavior, such as TLS downgrade attempts, revoked certificate rejection, OCSP timeout behavior, key export denial at HSM boundaries, or secrets access denial on policy violation.
- Surface-only verification – Relying on a TLS scanner summary without validating service identity properties (hostname and SAN checks, mutual authentication enforcement, and certificate chain evaluation) across representative clients and runtimes.
- No regression after change – Updating cipher policies, certificate profiles, rotation schedules, entropy sources, or crypto libraries without re-running the affected Verification and Validation activities and recording updated evidence.
- Entropy treated as assumed – Declaring randomness compliant without health checks, entropy monitoring, and nonce reuse detection evidence. Entropy failures often manifest as silent weaknesses until compromise.
- Revocation not exercised – Treating revocation as a configuration item rather than a tested capability. If revocation propagation, client behavior, and availability of status services are not validated, the compromise response remains theoretical.
- Scope blind spots – Excluding management planes, east–west service-to-service paths, service mesh control planes, backup encryption paths, or constrained devices from V&V coverage. Staging environments that do not mirror production introduce false confidence.
- Integration not exercised end-to-end – Validating components in isolation but not validating critical paths such as application to key management service to hardware security module, issuance to renewal to revocation, or secrets issuance to rotation to access audit correlation.
- Evidence gaps – Producing test outputs that are not traceable to a Test ID and Evidence Pack record. Artifacts lacking timestamps, configuration versions, or chain-of-custody metadata weaken auditability and peer review.
- Separation of duties collapses during testing – Allowing the same identity or pipeline to change cryptographic policy, approve the change, and validate the result. This undermines trust in the test outcome and complicates incident reconstruction.

Obsolete and withdrawn documents should not be used; please use replacements.

- Time synchronization ignored – Running tests without confirming time synchronization controls and clock skew baselines. Certificate validity, log correlation, OCSP behavior, and forensic timelines depend on reliable time.
- Post-quantum readiness misframed – Treating readiness as a blanket deployment requirement or running pilots without defined success criteria and rollback validation. Readiness evidence should remain scoped to inventory, triggers, and tested rollback unless a pilot is explicitly executed.

Table I-6. Traceability Matrix — Requirements (Section 5) to Verification and Validation (Section 12) to Related Technical Specs (Section 6)

Req ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
5.1	Enterprise crypto policy and governance	Policy catalog published; CI/CD policy checks active; deprecation registry maintained	Sample systems conform to approved algorithms and parameters; violations are blocked in CI/CD tests	6.1; 6.9
5.2	HSM and KMS deployment	Keys generated in HSM and KMS; no plaintext exports; M-of-N control enforced	Live rotations succeed without exposure; blocked exports alert; ceremony artifacts present and signed	6.4; 6.5
5.3	PKI hierarchy and automation	Offline root and intermediate CAs; ACME issuance and renewal configured	Leaf certs auto-renew ≤ 5 min; OCSP and CRL uptime ≥ 99.9 %; revocations reflected within 5 min	6.3; 6.2
5.4	Secrets management platform	Pre-commit and CI scanners enabled; dynamic secrets issued; RBAC applied	Secrets-in-code findings $\downarrow \geq 95$ % in 90 days; TTL ≤ 24 h; rotation on compromise ≤ 15 min	6.6
5.5	Authenticated time sync	Auth NTP and PTP configured; drift monitors enabled	Signed-log timestamps consistent; clock skew ≤ 1 s (P95); drift alerts resolved within SLA	6.9; 6.2
5.6	Secure software supply chain	Approved crypto libraries listed; SBOM verified; no custom crypto	Build fails on unapproved library; startup KAT and self-test pass	6.8
5.7	Network and transport readiness	TLS 1.3 policy applied; mTLS for service and admin; PFS enabled	Scans show only approved suites; mTLS ≥ 98 % coverage; downgrade attempts alert and block	6.2

Obsolete and withdrawn documents should not be used; please use replacements.

Req ID	Requirement (summary)	Verification (build-correct)	Validation (works-right)	Related Technical Specs
5.8	Audit-ready logging	Signed logs for key, certificate, and secret events; SIEM integration is active	Forensic replay succeeds; anomaly detections fire; retention meets policy	6.9
5.9	Entropy and randomness	DRBG configured; entropy sources validated; health monitors active	DRBG self-tests pass startup and hourly; 0 nonce reuse per key; alerts on reuse attempt	6.1; 6.4
5.10	Post-quantum readiness assessment	Capability inventory completed; migration triggers and rollback plan documented	Readiness review confirms dependencies and compatibility matrix; rollback exercise completed. If a pilot is executed, results confirm performance and interoperability within defined thresholds	6.7

How to use the matrix

- **Plan:** For each Section 5 requirement, schedule ≥ 1 Verification and ≥ 1 Validation activity linked to a Section 6 output.
- **Execute:** Run activities and record the test ID and Evidence Pack location for each row.
- **Maintain:** When requirements or outputs change, update tests and evidence; re-run entropy and certificate validation checks on the next release cycle.

Evidence Pack

Evidence for Section 12 Verification and Validation activities Must be collected and maintained in EP-09.5 (Verification and Validation). Each requirement row in Table I-6 Must include a dated record of build-correct verification and works-right validation, including objective pass or fail criteria, test execution artifacts, and remediation linkage where applicable. Evidence Must remain version-controlled and retained according to organizational audit requirements.


Minimum evidence expectations for EP-09.5 include:

- Test Plans and Procedures: Scope statement, environment boundary, test cases, tools or simulators used, prerequisites, and explicit pass or fail criteria. Include safety constraints for production and regulated systems.

Obsolete and withdrawn documents should not be used; please use replacements.

- **Test Identification and Traceability Ledger:** A ledger that maps *Table I-6 row to Section 6 output to Test ID to EP-09.5 artifact path to pass or fail status to date and owner*.
- **Verification Artifacts (build-correct):** Configuration snapshots and enforcement proofs showing the output exists and is configured as intended, such as TLS scan reports, certificate profile settings, key policy definitions, and secrets scanning outputs.
- **Validation Artifacts (works-right):** Proof that controls perform under operational conditions, including rotation results, renewal latency evidence, revocation propagation confirmation, OCSP and CRL availability results, and workload identity enforcement evidence.
- **Negative and Failure-Mode Tests:** Evidence of fail-closed behavior, including downgrade attempt results, revoked certificate rejection evidence, blocked key export events, entropy health failure simulations, and rollback outcomes when invoked.
- **SLO and Metric Snapshots:** Dated metric captures demonstrating compliance with defined targets, including rotation success rate, certificate renewal latency, secrets TTL compliance, OCSP and CRL availability, and entropy health pass rate.
- **Corrective Action Plans and Re-Test Evidence:** For any failed criterion, include CAP records with owner, deadline, exception handling, and the closure test showing the issue is remediated.
- **Change and Release Linkage:** A reference to the configuration change or release record that triggered the test cycle, including commit identifiers or change tickets, so reviewers can reconstruct the test context.

EP-09.5 entries Must link backward to EP-09.1 (Requirements) to show prerequisite readiness and Must link forward to EP-09.2 (Technical Specifications) to reference the implementation artifacts being tested.

	<p>Practitioner Guidance:</p> <ul style="list-style-type: none"> • Bind every control to proof. A CEK control is not complete until Verification and Validation results show compliance with SLOs, and the Evidence Pack record is attached. • Test the failure, not only the success. Perform negative tests, including entropy depletion, certificate revocation failure, and key rotation abort, to confirm fail-closed behavior and resilience. • Measure what matters. Track weekly rotation success rate, renewal latency, OCSP and CRL uptime, entropy health, and secrets TTL compliance; escalate breaches to remediation tickets within SLA.
---	--

Obsolete and withdrawn documents should not be used; please use replacements.

- Keep tests with the code. Store test plans, suites, and results in policy-as-code and control-as-code repositories; include a Test ID and an Evidence Pack entry for every change.



Quick Win Playbook:

Title: Certificate Renewal Reliability Baseline

Objective: Eliminate certificate-expiration outages by implementing automated certificate issuance and renewal, continuous status monitoring, and daily trust-chain verification, with measurable proof recorded in EP-09.5.

Target: Close certificate renewal failures and expiration outages across hybrid environments (§ 6.3).

Component/System: PKI issuance service (ACME), monitoring dashboards, automation agents.

Protects: Availability of secure transport channels and service identity.

Stops/Detects: Expired certificates, manual-renewal errors, and unmonitored trust-chain breaks.

Action: Deploy ACME issuance with ≤ 90 -day validity; enable auto-renew agents; alert on OCSP stale responses; verify trust chains daily.

Proof: ACME configuration diff + renewal log + OCSP availability report to EP-09.5 (implementation artifacts cross-linked to EP-09.2).

Metric: 100 % certs auto-renew within ≤ 5 min; OCSP/CRL uptime ≥ 99.9 %; 0 P1 outages from expiry.

Rollback: Revert auto-renew agent version if instability occurs; document exception and re-validation date.

Section 13. Implementation Guidelines

This section does not prescribe vendor-specific tooling or products. Parent Standards are durable, long-lived architectural foundations. Here, we describe how Sub-Standards and delivery teams translate the Parent's intent (ISAU-DS-CEK-1000) into testable,

Obsolete and withdrawn documents should not be used; please use replacements.

automatable, and auditable operational behaviors for Cryptography, Encryption, and Key Management (CEK).

Delivery mechanics for CICD integration, artifact signing, attestation, promotion, and rollback are governed by Annex J.

Purpose of This Section in Sub-Standards

Sub-Standards should use Implementation Guidelines to:

- Translate Parent expectations into enforceable CEK behaviors, such as key rotation service levels, certificate renewal latency targets, entropy health objectives, and dual control enforcement for sensitive key operations.
- Provide platform-agnostic practices that improve adoption, reduce integration risk, and align with ISAUnited's defensible-by-design philosophy.
- Surface common cryptographic failure modes and reduce their likelihood through measurable gates and automated tests.
- Provide repeatable as-code patterns that support lifecycle discipline, cryptographic assurance, and engineering rigor across hardware security modules, key management services, public key infrastructure, transport security profiles, secrets platforms, and post-quantum readiness activities.

Open Season Guidance for Contributors

Contributors developing CEK Sub-Standards should:

- Align guidance with the Parent's strategic posture and Section 6 outputs, including key and certificate lifecycle targets and observability expectations.
- Avoid vendor or product names and express controls as requirements, tests, and evidence linked to an Evidence Pack location.
- Include lessons learned, including what failed, why it failed, and how the test demonstrates correction.
- Favor reproducible engineering patterns expressed as policy-as-code or control-as-code.
- Provide a minimal standards mapping from specification or control to NIST or ISO clause (Section 8) to the Evidence Pack location. Control framework mappings remain in Section 9.

Technical Guidance

A. Organizing Principles

1. Everything as Code- Key policies, certificate lifecycles, HSM and KMS configurations, entropy monitors, and secrets issuance logic should be version-controlled, peer-reviewed, and released from protected branches.

Obsolete and withdrawn documents should not be used; please use replacements.

2. Non-bypassable Security Gates - Each merge or release should satisfy gates aligned to Sections 6 and 12. Example gates include:
 - DEK rotation success rate $\geq 99\%$.
 - Certificate renewal latency ≤ 5 minutes.
 - Secrets TTL ≤ 24 hours and rotation on compromise ≤ 15 minutes.
 - DRBG health pass rate = 100 % at startup and hourly.
 - Evidence Pack location recorded for each configuration change.
3. Immutable and Reproducible Deployments - Manual key or certificate changes after build should be avoided. Artifacts such as PKI bundles and policy definitions should be signed and pinned, with integrity verified before activation.
4. Least Privilege and Separation of Duties - Distinct identities should separate key administration, policy automation, and validation workflows. Secrets should be vaulted and rotated. Identity overlap and role misuse should trigger alerts and reviews.
5. Environment Parity - Staging should mirror production for key hierarchies, certificate profiles, transport profiles, and readiness exercises. Drift should be detected and reconciled before promotion.

B. Guardrails by Pipeline Stage

1. **Pre-Commit and Local**
 - Signed commits and secret scanning should run by default.
 - Cryptographic policy linting should reject unapproved algorithms or key lengths.
 - Test stubs for rotations and renewals should be generated for changed policies.
2. **Pull Request and Code Review**
 - CODEOWNERS review should be used for cryptographic policy and lifecycle changes.
 - Coverage gates should validate affected keys or certificates in staging or sandbox tests.
 - Pull requests should include Test IDs and the intended Evidence Pack location.
3. **Build and Package**
 - Deterministic policy bundles should be produced and signed.
 - Validation suites should be packaged alongside changes, such as TLS scanner configurations and entropy checks.
4. **Pre-Deploy and Release**
 - Drift checks should compare deployments to approved registries and key usage policies.
 - Canary rollouts should be paired with health monitors and rollback procedures.
 - Positive and negative tests should include renewal latency, revocation propagation, entropy variance, and fail-closed behaviors.
5. **Deploy and Runtime**

Obsolete and withdrawn documents should not be used; please use replacements.

- Runtime monitoring should track rotation compliance, renewal health, entropy signals, and secrets lifecycle telemetry.
- Unverified key material or certificates lacking evidence linkage should trigger incident handling and remediation workflows.

6. **Post-Deploy Validation and Operations**

- Continuous validation should execute key rotation tests, renewal checks, and downgrade simulations aligned to Section 12.
- Evidence artifacts should be captured per release, including policy diffs, validation results, and rollback records.

C. Identity, Access, and Secrets (normative alignment to §6)

- Dedicated service identities should be used for PKI, KMS and HSM, and secrets APIs, with mutual TLS and signed tokens for service calls.
- Secrets should be stored in an approved vault with audit logging, rotation, and access controls.
- Telemetry should include key identifiers, certificate identifiers, policy version, and timestamps to support forensic traceability.

D. CEK Supply Chain Integrity

- Only signed policy bundles that passed Section 12 tests should be promoted.
- Unverified modules or libraries should be quarantined until validated.
- Build and deploy identities should remain separated, and production writes from build jobs should be treated as high-risk events.


E. Measurement and Acceptance (aligned to §6 and §12)


- Key Lifecycle Integrity
Metric or Gate: DEK rotation $\geq 99\%$; KEK rotation within defined cryptoperiod; cryptographic erasure confirmed.
Evidence: Rotation logs; HSM and KMS audit records.
- Certificate Reliability
Metric or Gate: Auto-renew ≤ 5 minutes; OCSP and CRL uptime $\geq 99.9\%$.
Evidence: Renewal logs; revocation dashboards.
- Secrets Governance
Metric or Gate: TTL ≤ 24 hours; rotation on compromise ≤ 15 minutes.
Evidence: Vault audit trails; pipeline test results.
- Entropy Health
Metric or Gate: DRBG self-test pass = 100 %; nonce reuse events = 0.
Evidence: Entropy monitors; test reports.
- Evidence Completeness
Metric or Gate: Each change links Section 5 to Section 6, then to Section 12, via evidence linkage.
Evidence: Evidence ledger; review diffs.

Obsolete and withdrawn documents should not be used; please use replacements.

Common Pitfalls and the Engineered Countermeasure

1. Long-lived keys or certificates - rotation gates and renewal monitors with promotion holds for out-of-policy validity.
2. Entropy degradation or DRBG failure - continuous entropy health monitoring with quarantine and review when failures occur.
3. Manual renewal or manual rotation - automation-first workflows supported by validation evidence before promotion.
4. Secrets sprawl - secret-scanning gates with immediate remediation workflow for detected plaintext credentials.
5. Incomplete audit trail - evidence linkage required for changes, tests, and operational acceptance decisions.
6. Separation of duties collapse - pipeline identity separation supported by alerting and periodic review.

	<p>Practitioner Guidance:</p> <ul style="list-style-type: none"> • Integrate CEK configuration, validation, and evidence collection into CICD pipelines so assurance is continuous rather than periodic. • Maintain traceability using the Controls to Outputs to Tests to Evidence mapping approach described in Section 12. • Execute quarterly rotation and renewal drills, entropy health checks, and certificate revocation exercises to confirm operational readiness. • Capture lessons learned and feed them into the Open Season peer review to strengthen future revisions.
---	--

	<p>Quick Win Playbook:</p> <p>Title: Key and Certificate Rotation Ownership Dashboard</p> <p>Objective: Provide immediate visibility into key rotation and certificate renewal compliance by tying every active key and certificate to an owner, a lifecycle target, and evidence linkage aligned to the EP-09 structure.</p> <p>Target: Deploy a certificate and key rotation monitoring dashboard that ties every PKI certificate and HSM and KMS key to an owner, lifecycle target, and evidence linkage (Section 6.3, Section 6.4, Section 12).</p> <p>Component and System: PKI management service, HSM and KMS audit interfaces, and an internal dashboard or reporting tool.</p>
---	--

Obsolete and withdrawn documents should not be used; please use replacements.

	<p>Protects: Against expiration-related outages, stale keys, and rotation drift.</p> <p>Stops and Detects: Expired certificates, missed rotations, and dual-control violations.</p> <p>Action:</p> <ol style="list-style-type: none"> 1. Aggregate certificate and key metadata from PKI and HSM and KMS systems. 2. Add fields: owner, creation date, next rotation date, evidence linkage, and compliance status. 3. Visualize status (green = current; yellow = review due; red = expired or non-rotated). 4. Send a weekly report to the security engineering lead and auto-generate tickets for red items. <p>Proof: Dashboard screenshots, export file, rotation logs, and remediation ticket records stored in EP-09.2, with validation results stored in EP-09.5.</p> <p>Metric:</p> <ul style="list-style-type: none"> • 100 % of keys and certificates have owners and evidence linkage. • ≥ 99 % rotation compliance within defined lifecycle targets. • 0 expired certificates or keys beyond the rotation window. <p>Rollback: Restore the previous dashboard snapshot as read-only and retain artifacts as superseded evidence.</p>
--	--

Obsolete and withdrawn documents should not be used; please use replacements.

Appendices

Appendix A: Engineering Traceability Matrix (ETM)

This Engineering Traceability Matrix (ETM) links the CEK Parent Standard requirements to measurable technical specifications, core principles, control mappings, and Verification and Validation activities. It is designed for practitioners who need a single view of what must exist, what must be built, how it is tested, and where evidence is recorded.

Evidence Pack alignment: Evidence for this matrix is recorded using the five EP-09 locations. For each row, the primary acceptance evidence is captured in EP-09.5 (tests and results), with supporting artifacts referenced from EP-09.1 (readiness), EP-09.2 (implementation), EP-09.3 (foundational standards mapping), and EP-09.4 (control mappings).

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
5.1	Enterprise cryptographic policy and governance	6.1 Algorithm and Parameter Baselines; 6.9 Observability, KPIs, and SLOs	RP-05 Secure by Design; RP-06 Minimize Attack Surface; RP-15 Evidence Production	CSA CCM CEK-01; OWASP ASVS V6.2	Policy catalog published; parameter registry under change control; CICD gates present for disallowed algorithms	Sample services conform to approved baselines; violations blocked in CI CD; deprecation timelines enforced in review cadence	EP-09.5
5.2	HSM and KMS are operational for key protection	6.4 Key Management Operations; 6.9 Observability, KPIs, and SLOs	RP-01 Least Privilege; RP-03 Complete Mediation; RP-19 Protect Integrity	CSA CCM CEK-10; CSA CCM CEK-12	Keys generated inside HSM and KMS boundary; export denial configured; M-of-N approval path defined for KEKs	Rotation succeeds without plaintext exposure; blocked export events alert; ceremony artifacts verifiable;	EP-09.5

Obsolete and withdrawn documents should not be used; please use replacements.

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
						compromise drill completes within the target	
5.3	PKI hierarchy and certificate automation	6.3 PKI and Certificate Lifecycle; 6.2 Transport Security Profiles	RP-04 Defense in Depth; RP-14 Resilience and Recovery; RP-20 Protect Availability	CSA CCM CEK-03; CIS v8 3.10	Offline root and constrained intermediates established; ACME automation configured; OCSP and CRL endpoints operational	Renewal latency ≤ 5 minutes; OCSP and CRL availability $\geq 99.9\%$; revocation reflected within ≤ 5 minutes across representative clients	EP-09.5
5.4	Secrets management platform and lifecycle	6.6 Secrets Management; 6.9 Observability, KPIs, and SLOs	RP-01 Least Privilege; RP-15 Evidence Production; RP-16 Make Compromise Detection Easier	OWASP ASVS V6.4; CIS v8 8.9	Pre-commit and CI scanning enabled; dynamic issuance path established; access controls and audit logging configured	Secrets-in-code findings trend downward; production secret TTL ≤ 24 hours; compromise rotation ≤ 15 minutes; access misuse alerts observable	EP-09.5
5.5	Authenticated time synchronization	6.9 Observability, KPIs, and SLOs	RP-15 Evidence Production; RP-20 Protect Availability	CIS v8 8.9	Authenticated time sync configured; drift monitors active; time source documented	Clock skew ≤ 1 second (P95); certificate validity checks consistent; log correlation supports	EP-09.5

Obsolete and withdrawn documents should not be used; please use replacements.

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
						forensic replay without timeline gaps	
5.6	Secure software supply chain for crypto modules and libraries	6.8 Module Validation and Library Hygiene; 6.1 Algorithm and Parameter Baselines	RP-06 Minimize Attack Surface; RP-19 Protect Integrity; RP-15 Evidence Production	OWASP ASVS V6.2	Approved library allowlist defined; builds block unapproved crypto components; module validation tracked where required	Startup KAT and self-tests pass; failure triggers quarantine workflow; patching cadence evidenced; no deprecated library remains in protected Branches	EP-09.5
5.7	Network and transport readiness	6.2 Transport Security Profiles; 6.9 Observability, KPIs, and SLOs	RP-02 Zero Trust; RP-04 Defense in Depth; RP-18 Protect Confidentiality	CIS v8 3.10; CSA CCM CEK-03	TLS 1.3 profile configured; TLS 1.2 exception policy documented; mTLS configured for service and admin paths	Scans show only approved protocol and cipher-suite profile; mTLS coverage ≥ 98 % east-west and admin; downgrade attempts blocked and alerted	EP-09.5
5.8	Audit-ready logging and retention	6.9 Observability, KPIs, and SLOs	RP-15 Evidence Production; RP-16 Make Compromise Detection Easier	CIS v8 8.9	Signed and tamper-evident logs configured for key, cert, and secret events; SIEM ingestion verified	Forensic replay succeeds; anomaly detections fire on misuse; retention meets policy; evidence supports incident reconstruction	EP-09.5

Obsolete and withdrawn documents should not be used; please use replacements.

Req ID	Requirement (Inputs) (§5)	Technical Specifications (Outputs) (§6)	Core Principles (§7)	Control Mappings (§9)	Verification – Build Correct (§12)	Validation – Works Right (§12)	Evidence Pack ID
5.9	Entropy sources and randomness assurance	6.1 Algorithm and Parameter Baselines; 6.9 Observability, KPIs, and SLOs	RP-19 Protect Integrity; RP-16 Make Compromise Detection Easier	OWASP ASVS V6.3	DRBG configured; entropy sources validated; health monitors enabled; nonce and IV controls defined	DRBG health checks pass at startup and hourly; nonce reuse events = 0; reuse attempts alert and are investigated and closed	EP-09.5
5.10	Post-quantum readiness assessment	6.7 Cryptographic Agility and Post-Quantum Readiness	RP-17 Cryptographic Agility; RP-05 Secure by Design; RP-14 Resilience and Recovery	CSA CCM CEK-01; OWASP ASVS V6.2	Capability inventory completed; dependencies and exposure cataloged; migration triggers and rollback plan documented	Readiness review confirms compatibility matrix and rollback exercise results; pilot evidence is recorded only when a pilot is executed and evaluated	EP-09.5

Obsolete and withdrawn documents should not be used; please use replacements.

Appendix B: Evidence Pack Matrix

This summary matrix provides practitioners with a single, readable view of how the CEK Evidence Pack repository is organized for Parent Standard adoption. Each Evidence Pack location corresponds to a core section of the annex standard, enabling consistent evidence collection and review without prematurely creating substandard evidence structures.

Evidence Pack alignment: EP-09 is the Evidence Pack repository for D09. Evidence is organized into five section-aligned locations: EP-09.1 captures readiness artifacts for Section 5, EP-09.2 captures implementation artifacts for Section 6, EP-09.3 preserves clause-level foundational standards mappings for Section 8, EP-09.4 maintains external control mappings for Section 9, and EP-09.5 contains Verification and Validation test evidence for Section 12. Together, these five locations provide end-to-end traceability from prerequisites to implementation to proof.

Layer	EP Identifier	Purpose	Evidence Categories Included
EP Repository	EP-09	Evidence Pack repository for D09. Serves as the single entry point for CEK adoption evidence and traceability across Sections 5, 6, 8, 9, and 12.	<ul style="list-style-type: none"> • Index and file structure overview for EP-09.1 through EP-09.5 • Evidence Pack ledger showing Section reference, artifact name, date, owner, and review status • Traceability snapshot linking Inputs to Outputs to Tests and Evidence Pack locations • Change log capturing updates to evidence sets and review outcomes
Requirements	EP-09.1	Captures readiness and prerequisite evidence for Section 5 (Inputs). Demonstrates that baseline capability exists before implementation work begins.	<ul style="list-style-type: none"> • Cryptographic policy catalog approval record and governance ownership • Baseline inventories for keys, certificates, secrets, and trust stores • HSM and KMS boundary documentation and role separation notes • PKI topology overview and issuance and revocation service readiness

Obsolete and withdrawn documents should not be used; please use replacements.

Layer	EP Identifier	Purpose	Evidence Categories Included
			<ul style="list-style-type: none"> • Authenticated time synchronization configuration and drift monitoring baseline • Approved crypto library and module allowlist and provenance requirements • Post-quantum readiness assessment as a planning artifact, including dependency inventory and transition triggers
Technical Specifications	EP-09.2	Captures implementation evidence for Section 6 (Outputs). Demonstrates controls are built, configured, and enforced as engineered behaviors.	<ul style="list-style-type: none"> • Algorithm and parameter registry exports and deprecation timelines • Transport profile definitions and conformance outputs for protocol versions and cipher suites • PKI automation configuration, renewal configuration, and issuance and revocation logs • Key management policies, rotation job definitions, blocked export events, and ceremony artifacts, where applicable • Secrets management configurations, scanner outputs, dynamic issuance rules, and access audit trails • Observability artifacts such as signed and tamper-evident audit telemetry samples and dashboard definitions • Readiness artifacts for cryptographic agility, including compatibility matrices and rollback procedures when maintained
Foundational Standards	EP-09.3	Captures Section 8 alignment to the adopted NIST and ISO/IEC baselines. Provides clause-level mapping for design, implementation, and validation reviews.	<ul style="list-style-type: none"> • Clause-level mapping sheet linking CEK outputs to NIST and ISO/IEC references • Citation snapshots and revision identifiers for referenced publications • Standards selection rationale tied to CEK scope areas such as

Obsolete and withdrawn documents should not be used; please use replacements.

Layer	EP Identifier	Purpose	Evidence Categories Included
			key lifecycle, randomness, and module assurance <ul style="list-style-type: none"> • Documented divergence notes and compensating control statements when applicable • Mapping change history with dates and the responsible owner
Control Mappings	EP-09.4	Captures Section 9 mappings to external control frameworks. Shows how CEK outputs relate to widely used assurance catalogs without treating them as foundational baselines.	<ul style="list-style-type: none"> • Control mapping sheet linking each external control to related Section 6 outputs and Section 7 principles • Control selection rationale describing the CEK risk addressed by each mapping • Equivalence notes to prevent duplicate mappings across frameworks • Framework version tracking and update history • Exceptions and compensating measures when a control mapping is not applicable in a declared scope
Verification and Validation	EP-09.5	Captures Section 12 test evidence and acceptance records. Demonstrates build-correct verification and works-right validation with pass or fail outcomes and remediation linkage.	<ul style="list-style-type: none"> • Test plans and procedures with scope, prerequisites, and pass or fail criteria • Traceability ledger mapping Table I-6 rows to Test IDs and artifact paths • Verification artifacts such as configuration snapshots and enforcement proofs • Validation artifacts such as renewal latency, rotation success, revocation propagation, and OCSP and CRL availability results • Negative test artifacts demonstrating fail-closed behavior and recovery paths • SLO snapshots supporting acceptance decisions and corrective action plans with re-test results

Obsolete and withdrawn documents should not be used; please use replacements.

Layer	EP Identifier	Purpose	Evidence Categories Included
			<ul style="list-style-type: none">• Change references linking tests to the configuration or policy change that triggered validation

Obsolete and withdrawn documents should not be used; please use replacements.

Adoption References

NOTE: ISAUnited Charter Adoption of External Organizations.

ISAUnited formally adopts the work of the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) and the National Institute of Standards and Technology (NIST) as foundational standards bodies, and the Center for Internet Security (CIS), the Cloud Security Alliance (CSA), and the Open Worldwide Application Security Project (OWASP) as security control–framework organizations. This adoption aligns with each organization’s public mission and encourages use by practitioners and institutions. ISAUnited incorporates these organizations into its charter so that every Parent Standard and Sub-Standard is grounded in a common, defensible foundation.

a) **Foundational Standards (Parent level).**

ISAUnited adopts *ISO/IEC* and *NIST* as foundational standards organizations. Parent Standards align with these bodies for architectural grounding and auditability, and extend that foundation through ISAUnited’s normative, testable specifications. This alignment does not supersede *ISO/IEC* or *NIST*.

b) **Security Control Frameworks (Control level).**

ISAUnited adopts *CIS*, *CSA*, and *OWASP* as control framework organizations. Control mappings translate architectural intent into enforceable technical controls within Parent Standards and Sub-Standards. These frameworks provide alignment at the implementation level rather than at the foundational level.

c) **Precedence and scope.**

Foundational alignment (*ISO/IEC*, *NIST*) establishes the architectural baseline. Control frameworks (*CIS*, *CSA*, *OWASP*) provide enforceable mappings. ISAUnited’s security invariants and normative requirements govern implementation details while remaining consistent with the adopted organizations.

d) **Mapping.**

Each cited control mapping is tied to a defined output, an associated verification and validation activity, and an Evidence Pack ID to maintain end-to-end traceability from requirement to control, test, and evidence.

e) **Attribution.**

ISAUnited cites organizations by name, respects attribution requirements, and conducts periodic alignment reviews. Updates are recorded in the Change Log with corresponding evidence.

f) **Flow-downs.**

Obsolete and withdrawn documents should not be used; please use replacements.

(Parent to Sub-Standard). Parent alignment to the International *ISO/IEC* and *NIST* flows down as architectural invariants and minimum requirements that Sub-Standards must uphold or tighten. Parent-level mappings to *CIS*, *CSA*, and *OWASP* flow down as implementation control intents that Sub-Standards must operationalize as controls-as-code, tests, and evidence. Each flow-down **MUST** reference the Parent clause, the adopted organization name, the Sub-Standard clause that implements it, the associated verification/validation test, and an Evidence Pack ID for traceability. Any variance requires a written rationale, compensating controls, and a time-bounded expiry recorded with an Evidence Pack ID.

Obsolete and withdrawn documents should not be used; please use replacements.

Change Log and Revision History

Review Date	Changes	Committee	Action	Status
January 2026	Standards Revision	Standards Committee	Publication	Draft v1 published
November 2025	Standards Submitted	Technical Fellow Society	Peer review	Pending
October 2025	Standards Revision	Task Group ISAU-TG39-2024	Draft submitted	Complete
December 2024	Standards Development (Parent D01)	Task Group ISAU-TG39-2024	Draft complete	Complete

End of Document
IO.

Obsolete and withdrawn documents should not be used; please use replacements.