Coprime-Factor Security Architecture

Technical Report v0.2 (Working Draft)

Erkan Yalcinkaya
Luminesce Limited (United Kingdom)
AxoDen Labs Research Initiative

November 2025

Abstract

We disclose a security architecture in which defense layers are engineered to be mathematically independent in their failure characteristics, analogous to coprime factors. The design reduces correlated risk across cryptography, trust roots, codebases, runtime, supply chain, and control planes. We introduce the Independence Score (IS) and Common-Mode Risk Index (CMRI), with implementable patterns: dual-primitive encryption, split-vendor key ceremonies, dual policy consensus, multi-root attestation, and divergent transport protections. This work builds on defense-in-depth principles while introducing a quantitative model to minimize correlated failure surfaces across heterogeneous security layers.

Keywords: security architecture; defense-in-depth; independence metrics; cryptographic diversity; coprime security; correlated failures; provable security; AxoDen; Helix4

Contents

Related Work

Traditional defense-in-depth architectures emphasize layered controls, but typically lack quantitative guarantees of independence between those layers. Foundational work includes Saltzer & Schroeder's secure design principles, NIST SP 800-160, MITRE ATT&CK dependency mapping, and cloud security models from NIST and ENISA. These frameworks recognize correlated failure risk but do not formalize a mathematical model for independence. Coprime-Factor Security establishes quantitative independence metrics, applies a number-theoretic analogy for trust separation, outlines measurable audit criteria, and provides implementable patterns (DPE, DPC, SVK, DT, TA). To our knowledge, no existing framework offers a mathematically-grounded operational standard for provably independent security layers.

1 Executive Summary

- **Problem:** Conventional defense-in-depth often hides correlated risks (same crypto family, shared libraries, single control plane)
- **Idea:** Engineer layers as **coprime factors**—each using disjoint primitives, trust roots, codebases, and operational paths
- **Outcome:** Lower breach probability, provable separation of concerns, auditable independence metrics

2 Principles & Independence Criteria

Design layers so that for any pair (L_i, L_j):

- **Cryptographic independence:** different algorithm *families* (e.g., AES-GCM $\leftrightarrow$ ChaCha20-Poly1305)
- **Trust root independence:** separate CAs/KMS/HSMs (ideally different vendors + firmware trees)
- **Codebase independence:** different libraries, compilers, languages; no shared critical dependencies
- **Runtime independence:** different isolation tech (VM $\leftrightarrow$ container $\leftrightarrow$ enclave/microVM), different kernels
- **Ops path independence:** separate teams, approval flows, credentials, monitoring stacks
- **Supply-chain independence:** distinct artifact pipelines, signing roots, registries
- **Control-plane independence:** policy engines and orchestration stacks that cannot impersonate each other

> **Independence Rule:** If any two layers share a single catastrophic common dependency, they are *not* coprime.

3 Formal Model (Concise)

Let $(L_1, \ldots, L_k)$ be layers; (C_i) is event "Layer i compromised" with probability (p_i). We target minimal correlation.

Independence Score (IS): For each dimension $d \in \{$crypto, trust-root, codebase, runtime, ops, supply, co

$$IS = \frac{1}{\binom{k}{2}} \sum_{i<j} \sum_{d} w_d \cdot s_d^{(i,j)}$$

Target **IS** $\geq$ **0.85** before production.

4 Reference Architecture (High Level)

User $\leftrightarrow$ **Edge** $\leftrightarrow$ **App Plane** $\leftrightarrow$ **Data Plane** $\leftrightarrow$ **Control Plane** $\leftrightarrow$ **Audit Plane**

Layer A - Identity & Access (Coprime AuthN)

- FIDO2/WebAuthn (ECDSA P-256) on vendor HSM-A
- PLUS independent TOTP/HOTP (HMAC-SHA1/256) or passkey on HSM-B
- Optional biometric check locally; never elevates trust by itself

Layer B - Transport & Session

- mTLS (TLS 1.3, library L1) *and* secondary WireGuard tunnel (library L2) for admin paths
- Different cipher families (AES-GCM vs ChaCha20-Poly1305)

Layer C - Authorization (Dual Policy Consensus)

- OPA/Rego engine and AWS Cedar (or equivalent) run in separate control planes
- High-risk actions require **AND-consensus**; low-risk allow OR with rate-cap

Layer D - Data At Rest (Dual Encryption)

- Envelope encryption with AES-GCM (KMS-A)
- Nested layer with ChaCha20-Poly1305 (KMS-B)
- Keys held in HSMs from different vendors with distinct firmware

Layer E - Integrity & Build Trust

- Provenance via Sigstore (Fulcio/Rekor) + separate in-house CA
- Reproducible builds verified by two independent verifiers

Layer F - Observability & Audit

- Append-only Merkle log (e.g., Trillian) inside org
- Daily external anchoring (public transparency log or blockchain) with separate signer

Layer G - Runtime Isolation

- Service set S1 on microVM/Firecracker; S2 on containers with SELinux/AppArmor; disjoint kernels

5 Pattern Catalogue (Implementable)

P1. Dual-Primitive Encryption (DPE)

- Apply two independent AEAD schemes with unrelated keys from independent KMS/HSM
- Decrypt path requires both layers; failure leaves data safe

P2. Dual Policy Consensus (DPC)

- Two policy engines evaluate requests; decision = AND for privileged ops
- Engines have separate repos, CI, and deploy channels

P3. Split-Vendor Key Ceremony (SVK)

- Root keys generated in two vendor HSMs, different RNGs; quorum requires both

P4. Divergent Transport (DT)

- Administrative channels traverse a second, distinct VPN/overlay using disjoint crypto

P5. Twin Attestation (TA)

- Workload must present attestation from *two* independent roots (e.g., TPM quote + SEV-SNP/TEE report)

6 Example Config (Pseudo-YAML)

```
authn:
  factors:
    - type: fido2
      curve: p256
      hsm: vendorA
    - type: totp
      algo: hmac-sha256
      hsm: vendorB
  require: AND

transport:
  primary_tls:
    lib: rustls
    aead: aes-256-gcm
  admin_overlay:
    type: wireguard
    aead: chacha20-poly1305

authorization:
  engines:
    - name: opa
      source_repo: git://corp/opa-policies
    - name: cedar
      source_repo: git://corp/cedar-policies
  decision: AND

data_at_rest:
  layer1:
    aead: aes-256-gcm
    kms: KMS-A
  layer2:
    aead: chacha20-poly1305
    kms: KMS-B

attestation:
  require:
    - sigstore-fulcio
    - org-ca
```

7 Threat Model & Common-Mode Kill Switches

Adversaries

APT with supply-chain access; insider with privileged creds; cryptographic downgrade attacks

Common-Mode Risks & Mitigations

- *Shared library vuln (e.g., OpenSSL)* → Use different TLS stacks (BoringSSL vs rustls), different crypto primitives
- *Single CA/KMS breach* → Split vendors/roots; rotate quorum
- *Cloud provider control-plane bug* → Cross-account + cross-region + shadow control plane
- *Policy engine bug* → Dual consensus with canary deny on anomaly

Kill Switches

- Per-layer hard failure defaults to *deny* for privileged ops
- Emergency policy: force AND→OR only via out-of-band M-of-N key ceremony with board-level approval

8 Verification & Testing

- **Tabletop & Red Team:** Simulate single-layer failures; verify others hold
- **Fault Injection:** Disable one HSM or policy engine; ensure system continues with reduced capability
- **Synthetic Compromise Runs:** Introduce signed but invalid artifact to test twin attestation
- **Metrics:** see Section **??**

9 Metrics & SLOs

- **Independence Score (IS):** ≥ 0.85 pre-prod; ≥ 0.9 for Tier-1 systems
- **Common-Mode Risk Index (CMRI):** measured via shared-dep graph; target ≤ 0.1
- **Attack Path Reduction (APR):** $\geq 70\%$ fewer valid single-path exploits vs baseline
- **Dual Decision Coverage:** $\geq 99\%$ of privileged actions gated by DPC
- **Key Diversity Index:** ≥ 2 vendors, ≥ 2 RNG classes, ≥ 2 crypto families in use

10 Rollout Plan

- **Phase 0 (2-3 weeks):** Dependency graphing, IS/CMRI baseline, choose dual primitives, pick vendors
- **Phase 1 (4-6 weeks):** Implement P1-P3 for a critical service; wire metrics; run failure drills
- **Phase 2 (6-10 weeks):** Extend to transport (P4) and twin attestation (P5); introduce automated checks
- **Phase 3 (Quarterly):** Expand to all Tier-1 services; enforce org policy: any new high-risk component must meet IS ≥ 0.85

11 Governance & Audit

- Quarterly **Independence Review** with sign-off on IS/CMRI
- SBOM and provenance checks from *two* independent verifiers
- External anchoring of logs; third-party attest review

12 Limitations & Tradeoffs

- Complexity and latency overhead; dual encryption and consensus add cost
- Operational burden: two stacks to maintain; require clear runbooks
- Some domains can't easily achieve runtime independence (e.g., single cloud kernel)

Mitigate via scoping (Tiered application), automation, and clear exemption process with compensating controls.

13 Visuals (Placeholders)

- **Layered Stack Diagram:** user→edge→app→data with dual paths highlighted
- **Independence Matrix:** rows=layers, cols=dimensions (crypto/trust/code/runtime/ops/-supply/control)

14 Appendix: Coprime Analogy

In number theory, if factors are **coprime**, divisibility properties compose cleanly: to defeat a product, you must defeat *each* prime factor independently.

In security, we emulate this by engineering **pairwise-independent failure modes** so that a single exploit vector cannot cascade across layers.

15 Summary Tables (V0.2 Consolidation)

Core Independence Dimensions

Dimension	Requirement
Crypto	Distinct algorithm families, RNGs, KMS/HSM vendors
Trust Root	Separate CAs/HSM vendors & firmware lines
Codebase	Independent libs/languages/compilers
Runtime	microVM + container isolation, kernel separation
Ops Path	Distinct credentials & approval flows
Supply Chain	Independent CI/CD + signing systems
Control Plane	Dual policy engines; no impersonation capability

Engineering Patterns

Metrics

These summary tables provide a standards-ready reference. Full elaboration in prior sections.

Pattern	Meaning
DPE	Dual-Primitive Encryption
DPC	Dual Policy Consensus
SVK	Split-Vendor Key Ceremony
DT	Divergent Transport
TA	Twin Attestation

Metric	Target
Independence Score (IS)	≥ 0.85
Common-Mode Risk Index (CMRI)	≤ 0.10

16 Visual Architecture Reference (Placeholder)

```
User
  |
  v
[AuthN Layer A - FIDO2/P256/HSM-A] ---AND--- [AuthN Layer B - TOTP/HSM-B]
  |
  v
[Transport TLS / AES] + [Admin Overlay WireGuard / ChaCha]
  |
  v
[OPA Policy] ---AND--- [Cedar Policy]
  |
  v
[Data Encryption AES] + [Data Encryption ChaCha]
  |
  v
[Sigstore + Merkle Anchoring]
```

> **Blueprint Outcome:** A repeatable method to design, measure, and audit independence across security layers, reducing correlated failure risk.

Author: Erkan Yalcinkaya
Affiliation: Luminesce Limited (United Kingdom)
Project: AxoDen Labs Research Initiative