

# Coprime-Factor Security Architecture

Technical Report v3.0

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

November 2025

**License:** CC-BY-4.0

## Abstract

Coprime-Factor Security is a quantitative defense-in-depth framework that engineers independence across security layers to minimize correlated failures. Rather than assuming isolation between cryptography, trust anchors, control planes, runtime, and supply chain, it **enforces demonstrable separation** using measurable independence criteria and architectural patterns.

We define two primary metrics:

- **Independence Score (IS)** — weighted pairwise disjointness across seven dimensions
- **Common-Mode Risk Index (CMRI)** — worst-case shared-dependency exposure

We introduce operational patterns — dual-primitive encryption, dual policy consensus, split-vendor key ceremony, divergent transport, and twin attestation — and provide a scoring method, drift detection loop, and fail-safe execution rules.

Coprime-Factor Security generalizes principles used in **ASIL-M multi-root attestation** to full security stacks, providing a repeatable architecture for reducing correlated compromise risk in high-assurance systems.

**Keywords:** correlated failure, independence metrics, defense-in-depth, cryptographic diversity, supply-chain trust, attestation diversity, AxoDen

## Contents

<b>1 Context &amp; Relationship to ASIL-M</b>	<b>3</b>
<b>2 Related Work</b>	<b>3</b>
<b>3 Executive Summary</b>	<b>3</b>
<b>4 Principles &amp; Independence Criteria</b>	<b>3</b>
<b>5 Formal Model</b>	<b>4</b>
5.1 Independence Score (IS) . . . . .	4
5.2 Common-Mode Risk Index (CMRI) . . . . .	4
5.3 Engineering Bound . . . . .	4
<b>6 Reference Architecture</b>	<b>4</b>
<b>7 Patterns</b>	<b>5</b>

<b>8 Example Config</b>	<b>5</b>
<b>9 Threat Model &amp; Controls</b>	<b>6</b>
9.1 Threats . . . . .	6
9.2 Kill Switches . . . . .	6
<b>10 Verification &amp; Testing</b>	<b>6</b>
10.1 Drills . . . . .	6
10.2 Mini-Case (data breach drill) . . . . .	6
<b>11 Metrics &amp; SLOs</b>	<b>6</b>
<b>12 Rollout Plan</b>	<b>7</b>
<b>13 Governance &amp; Audit</b>	<b>7</b>
<b>14 Limitations &amp; Tradeoffs</b>	<b>7</b>
<b>A Coprime Analogy</b>	<b>7</b>
<b>B Mini-Case Exercises</b>	<b>7</b>
B.1 Attestation loss on one root . . . . .	7
B.2 TLS library CVE . . . . .	7
B.3 Internal CA compromise . . . . .	7
<b>C Summary Tables</b>	<b>7</b>
C.1 Independence Dimensions . . . . .	7
C.2 Patterns . . . . .	7
C.3 Thresholds . . . . .	8
<b>D Architecture Sketch</b>	<b>8</b>

# 1 Context & Relationship to ASIL-M

**ASIL-M** demonstrates multi-root trust for AI inference.

**Coprime Security** is the **general parent architecture**:

Domain	ASIL-M	Coprime-Factor Security
Scope	AI inference trust	Full security stack
Guarantee	$\pi \rightarrow 0$ enforced at attestation layer	Independence engineered across entire stack
Thresholds	$IS \geq 0.85 + \text{semantic guard}$	$IS \geq 0.85 / CMRI \leq 0.10$
Artifacts	Canonical Attestation Record	Org-wide independence and failure audit trail

Table 1: Comparison of ASIL-M and Coprime-Factor Security

**Takeaway:** Coprime Security = ASIL-M principle applied system-wide.

## 2 Related Work

Prior research domains address independence partially:

Area	Relevance
Saltzer & Schroeder	Classical security design principles
NIST SP 800-160	System security engineering
Common Criteria / ISO 15408	Assurance & independence levels
IEC 61508 / DO-178C	Safety separation & independence requirements
Threshold cryptography / BFT	Formal independence assumptions in distributed protocols
SLSA / in-toto	Supply-chain provenance & build independence
Zero-trust runtime	Environment trust, not cross-stack independence

Table 2: Related work domains

**Novel contribution:** A unified model + metrics + implementation patterns for engineered **cross-domain independence**, not assumed independence.

**Note:** “Coprime” is used as a **design metaphor** for engineered pairwise independence, not a literal number-theoretic mapping.

## 3 Executive Summary

- **Goal:** Minimize correlated failure across layers.
- **Method:** Structure each security layer so exploiting one does not help exploit another.
- **Outcome:** Independently-verifiable, metric-driven defense-in-depth.

## 4 Principles & Independence Criteria

For layers  $L_i, L_j$ , ensure disjointness across:

- Crypto families & RNGs
- Trust roots (CA/KMS/HSM vendors, firmware lines)
- Codebase & build pipeline

- Runtime / kernel / hypervisor
- Ops credentials & reviewer paths
- Supply-chain & CI/CD provenance
- Control-plane policy engines

If a single catastrophic dependency overlaps, that pair = **not independent (score 0)**.

## 5 Formal Model

### 5.1 Independence Score (IS)

Dimensions: {crypto, trust, code, runtime, ops, supply, control}

Let  $s_d^{(i,j)} \in [0, 1]$  be independence per dimension.

Let weights  $w_d \geq 0$  with  $\sum w_d = 1$ .

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

**Hard-fail rule:** If any mandatory dimension is not independent  $\rightarrow$  score = 0.

### 5.2 Common-Mode Risk Index (CMRI)

Binary overlap indicator  $O_d^{(i,j)} \in \{0, 1\}$ .

Weights  $\alpha_d \geq 0$ ,  $\sum \alpha_d = 1$ .

$$C(i, j) = \sum_d \alpha_d \cdot O_d^{(i,j)} \quad (2)$$

$$\text{CMRI} = \max_{i < j} C(i, j) \quad (3)$$

### 5.3 Engineering Bound

If each layer has compromise probability  $\pi$  and  $\text{IS} \geq \tau$ , then worst-case correlated breach bound:

$$P(\text{breach}) \leq k\pi + \binom{k}{2} \cdot (1 - \tau) \cdot \pi \quad (4)$$

**Interpretation:** Increasing IS directly shrinks correlated compromise term.

**Targets:**

- Launch:  $\text{IS} \geq 0.85$ ,  $\text{CMRI} \leq 0.10$
- Tier-1 critical:  $\text{IS} \geq 0.90$ ,  $\text{CMRI} \leq 0.05$

## 6 Reference Architecture

Planes:

User  $\rightarrow$  Edge  $\rightarrow$  App  $\rightarrow$  Data  $\rightarrow$  Control  $\rightarrow$  Audit

The architecture implements independence across six key layers (see Figure 1 for complete diagram):

- **AuthN:** FIDO2/HSM-A AND TOTP/HSM-B
- **Transport:** TLS vs WireGuard dual overlay
- **Policy:** OPA AND Cedar
- **Storage:** AES/KMS-A + ChaCha/KMS-B
- **Build trust:** Sigstore + independent internal CA
- **Runtime:** microVM vs container kernel separation
- **Audit:** Merkle internal + external anchor

## 7 Patterns

ID	Pattern	Purpose
P1	Dual-Primitive Encryption	Crypto & key independence
P2	Dual Policy Consensus	Control-plane independence
P3	Split-Vendor Key Ceremony	Hardware RNG/vendor separation
P4	Divergent Transport	Separate channels & cipher families
P5	Twin Attestation	Multi-root trust enforcement

Table 3: Operational patterns for independence

## 8 Example Config

Listing 1: Example configuration demonstrating independence patterns

```

1 authn:
2   require: AND
3   factors:
4     - type: fido2
5       hsm: vendorA
6     - type: totp
7       hsm: vendorB
8
9 transport:
10   tls:
11     lib: rustls
12   admin_overlay:
13     type: wireguard
14
15 policy:
16   engines:
17     - opa
18     - cedar
19   decision: AND
20
21 data:
22   layer1:
23     aead: aes-gcm
24     kms: KMS - A
25   layer2:
26     aead: chacha20
27     kms: KMS - B

```

```

28
29  attestation:
30    require:
31      - sigstore-fulcio
32      - org-ca

```

## 9 Threat Model & Controls

### 9.1 Threats

- Software monoculture exploits
- Supply-chain compromises
- Insider abuse of control-plane paths
- Cryptographic break/downgrade
- Hypervisor / cloud control-plane breach

### 9.2 Kill Switches

- Hard fail → deny privileged ops
- AND→OR override only via board-supervised M-of-N ceremony

## 10 Verification & Testing

### 10.1 Drills

- Disable 1 HSM → expect degraded mode only
- Inject signed but invalid binary → attestation fail
- Force TLS lib CVE scenario → overlay must sustain protection

### 10.2 Mini-Case (data breach drill)

- Compromise KMS-A keys
- Verify ChaCha/KMS-B layer protects data
- Score recomputed; audit logs reflect degraded independence

## 11 Metrics & SLOs

Metric	Target
IS	$\geq 0.85$ ( $\geq 0.90$ Tier-1)
CMRI	$\leq 0.10$ ( $\leq 0.05$ Tier-1)
Dual coverage	$\geq 99\%$ privileged ops dual-gated
APR	$\geq 70\%$ reduction in single-path exploitability

Table 4: Metrics and Service Level Objectives

## 12 Rollout Plan

- 0:** Model dependencies, baseline IS/CMRI
- 1:** Deploy P1-P3 on critical service
- 2:** Add P4-P5, automate scoring
- 3:** Extend to Tier-1 systems; enforce IS gates

## 13 Governance & Audit

- Quarterly independence review
- Dual third-party supply-chain proofs
- External timestamp anchoring (e.g., RFC-3161)
- Exemption process w/ compensating controls

## 14 Limitations & Tradeoffs

- Cost & operational complexity
- Harder for single-cloud tenants to reach runtime independence
- Cultural shift: security monoculture → **security dual-culture**

## A Coprime Analogy

Coprime = metaphor: pairwise-engineered independence.

Goal: compromise **must** require independent attack paths.

## B Mini-Case Exercises

### B.1 Attestation loss on one root

- Expected: degrade, higher logging, no privileged paths

### B.2 TLS library CVE

- Secondary channel remains secure; CMRI re-evaluated

### B.3 Internal CA compromise

- Sigstore lane continues; alert + emergency ceremony

## C Summary Tables

### C.1 Independence Dimensions

Crypto | Trust Root | Code | Runtime | Ops | Supply | Control

### C.2 Patterns

DPE | DPC | SVK | DT | TA

### C.3 Thresholds

$IS \geq 0.85 / CMRI \leq 0.10$   
 Tier-1:  $IS \geq 0.90 / CMRI \leq 0.05$

## D Architecture Sketch

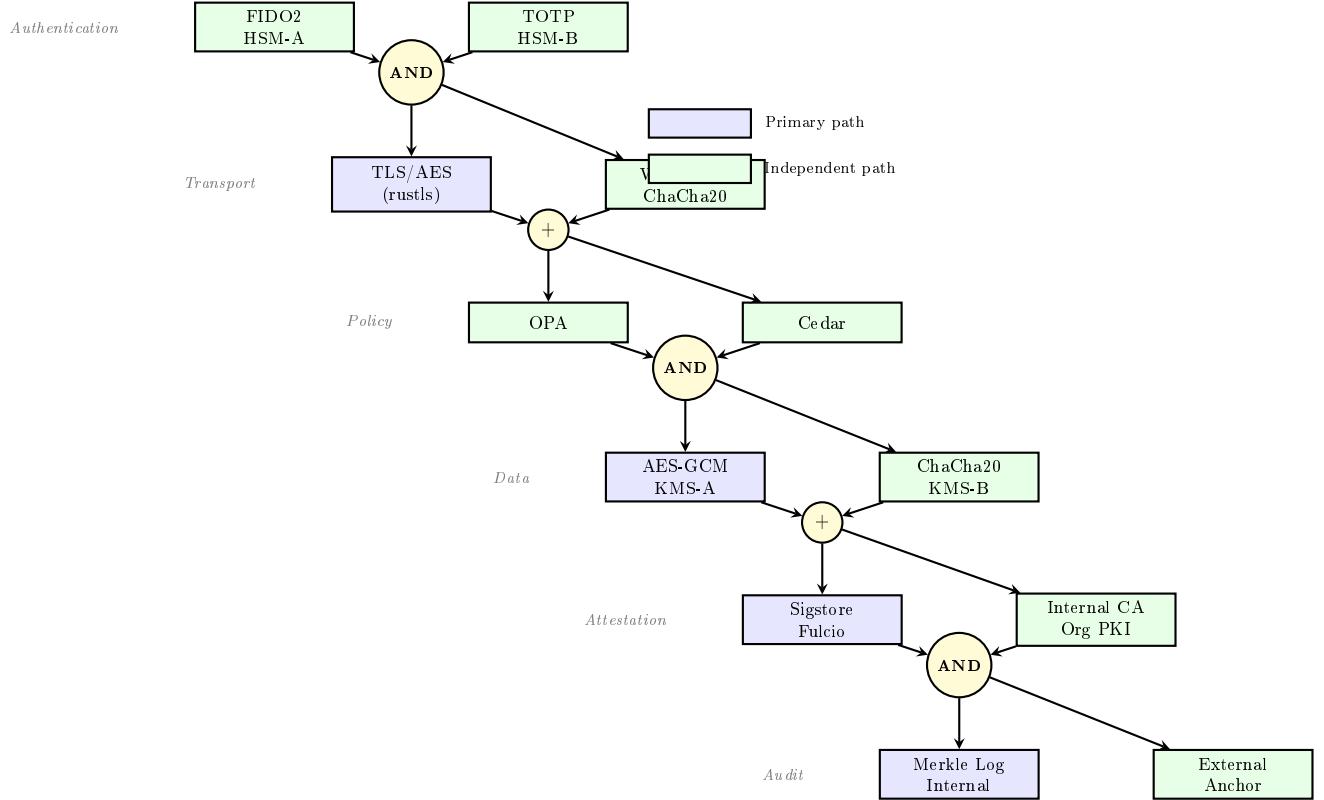


Figure 1: Coprime-Factor Security Architecture showing dual-path independence across layers. Each layer employs distinct cryptographic primitives, trust roots, and vendors to minimize correlated failure risk. **AND** operators enforce that both paths must succeed; **+** operators indicate additive protection.

**Outcome:** Correlated compromise becomes provably harder and auditable. Attackers must independently breach both paths at each layer to succeed.

---

**Project:** AxoDen Labs Research Initiative

**License:** CC-BY-4.0