PRIVATE.ME

Agent Identity Protocol

Technical White Paper

Information-Theoretic Security for AI Agent Communication

Version 2.0

March 2026

Confidential — Early Access Distribution Only

Powered by Xail

Abstract

The PRIVATE.ME Agent Identity Protocol provides cryptographic identity and information-theoretically secure message delivery for AI agent systems. Each agent is provisioned with an Ed25519 asymmetric key pair registered as a Decentralized Identifier (DID) in a trust registry. Messages are encrypted to the recipient's public key using AES-256-GCM, assembled into a signed envelope under the encrypt-then-sign protocol order, and optionally split across independent delivery channel endpoints using the XorIDA (Kurihara-Matsumoto) information dispersal algorithm operating over GF(2).

The split-channel delivery mode provides two properties that no other agent communication protocol provides: (1) information-theoretic content security, where fewer than k shares reveal zero bits about message content regardless of adversary computational resources, and (2) structural prompt injection defense, where a party controlling fewer than k delivery channel endpoints cannot cause the recipient agent to reconstruct or execute any instruction payload, independent of payload content or sophistication.

A bounded blast radius architecture limits signing key compromise to a 30-second temporal window, a signed permission scope, and zero content exposure through cryptographic independence of signing and encryption. Instant revocation takes effect on the immediately subsequent request with no propagation delay.

1. The Security Problem

1.1 The API Key Crisis

45.6% of teams currently authenticate AI agents with shared API keys — static bearer tokens with no sender identity, no scope binding, no replay protection, and no expiry. A leaked key is an unbounded security event: full access, indefinitely, until someone notices and rotates.

As agents take increasingly consequential actions — approving financial transactions, executing infrastructure changes, controlling physical access, processing healthcare data — the authorization model protecting those actions must match the consequence. It does not.

1.2 Transport Security Is Not Message Security

TLS protects data between two endpoints. When it terminates, the endpoint operator reads the message in plaintext. For agent-to-agent communication across cloud infrastructure, the message passes through at least one operator who holds a complete, readable copy. For applications subject to attorney-client privilege, HIPAA, or zero-trust mandates, this is architecturally insufficient.

1.3 Prompt Injection: The Structural Gap

When Agent A sends instructions to Agent B, any party controlling the channel between them can substitute their own instruction. Content-based defenses — input validation, LLM firewalls, classifier-based detection — can be bypassed by adaptive attackers. OWASP's 2025 LLM Top 10 placed prompt injection at the top of emerging AI risks. The research community has explicitly called for architectural solutions rather than content filters.

No existing agent communication protocol provides a structural defense against prompt injection. The PRIVATE.ME split-channel architecture provides one.

2. The XorIDA Algorithm

2.1 What XorIDA Is

XorIDA is the Kurihara-Matsumoto Information Dispersal Algorithm, operating over the Galois Field GF(2). It splits a message M into n shares such that any k shares reconstruct M exactly, and any set of fewer than k shares reveals zero information about M — unconditionally, regardless of adversary computational resources.

All arithmetic is bytewise XOR over GF(2). No modular arithmetic, no polynomial evaluation, no key material. Performance: approximately 1ms for a 1MB payload in the default 2-of-3 configuration.

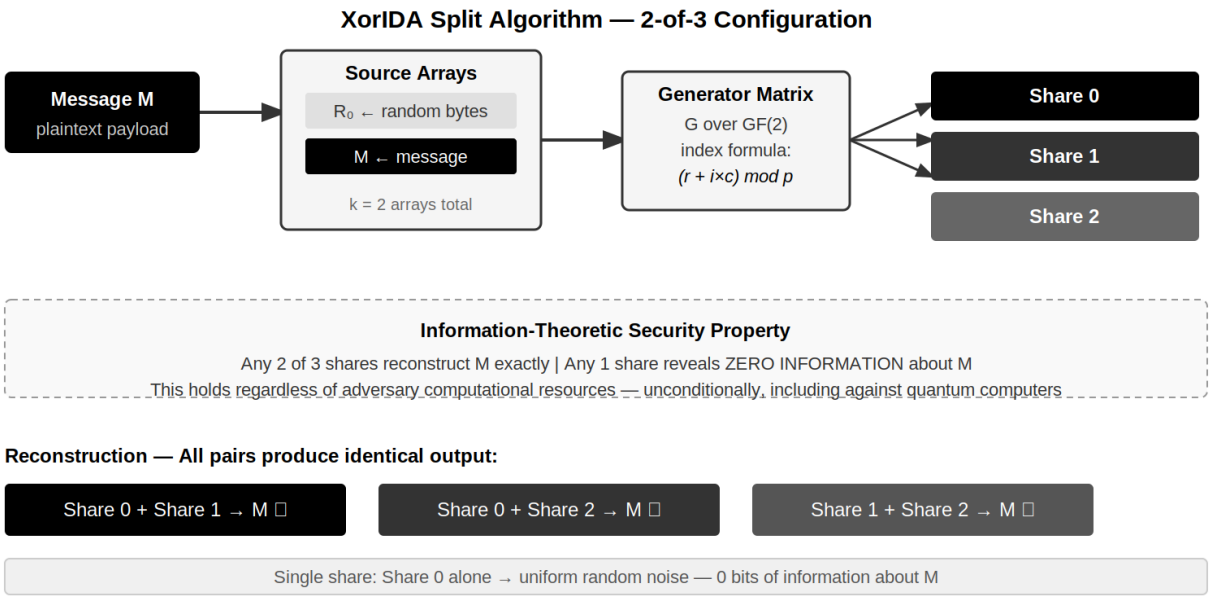

Figure 1. XorIDA 2-of-3 split: message M → k source arrays → generator matrix → n shares. Any 2 shares reconstruct M. Any 1 share is information-theoretically indistinguishable from random noise.

2.2 Algorithm Parameters

Configuration	k (threshold)	n (total)	p (prime)	Application
2-of-3 (default)	2	3	3	Standard agent delivery — single-channel fault tolerant
3-of-4	3	4	5	High-value instruction delivery
3-of-5	3	5	5	Maximum redundancy

2.3 Information-Theoretic Security

The security guarantee of XorIDA is information-theoretic — it holds against an adversary with unlimited computational resources, including quantum computers. This is a fundamentally different class of guarantee from all computationally secure alternatives (TLS, RSA, ECDSA, AES, lattice-based PQC), whose security rests on hardness assumptions that may be violated by future computational advances.

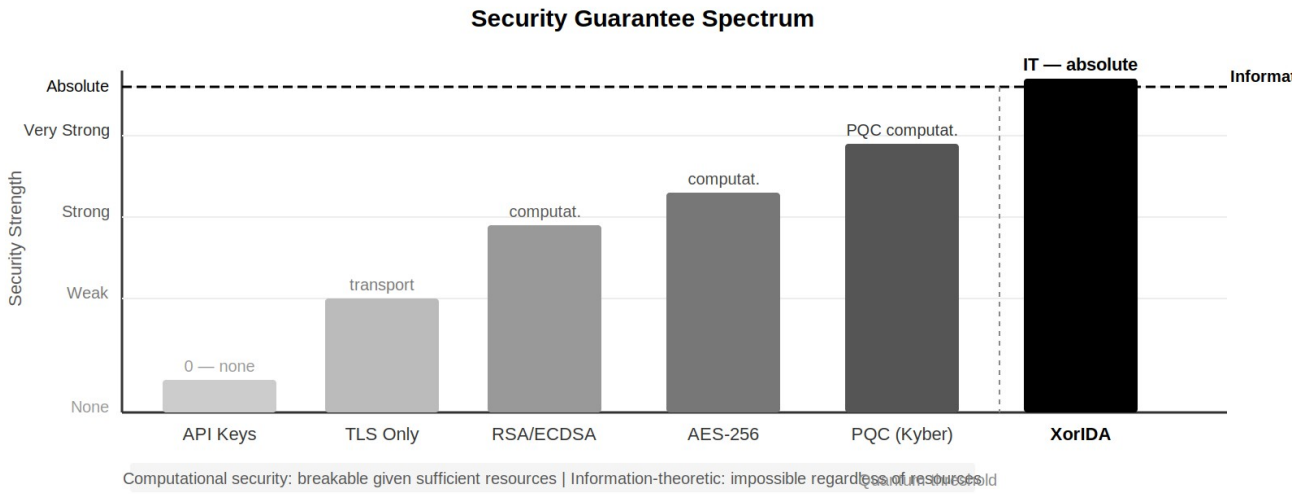

Figure 2. Security guarantee spectrum. XorIDA crosses the information-theoretic boundary — the only agent communication primitive with unconditional security.

2.4 Verification and Testing

The @xail/crypto library is verified to 100% line coverage with:

- **Known-answer test vectors:** hardcoded input/output pairs verified against the Kurihara-Matsumoto paper and the Xecret.io production deployment
- **Round-trip tests:** all C(n,k) share combinations for all supported configurations; message sizes 1 byte to 10MB
- **Fuzz testing:** random message sizes and binary content across all configurations
- **Statistical verification:** 10,000-sample chi-squared test confirming share byte distribution is indistinguishable from uniform random ($p > 0.01$); Pearson correlation coefficient < 0.01 confirming share independence

3. The Signed Envelope Format

3.1 Structure and Security Properties

Every agent message is assembled into a signed envelope before transmission. The format is version-tagged, language-agnostic, and uses standard JSON with standard cryptographic primitives.

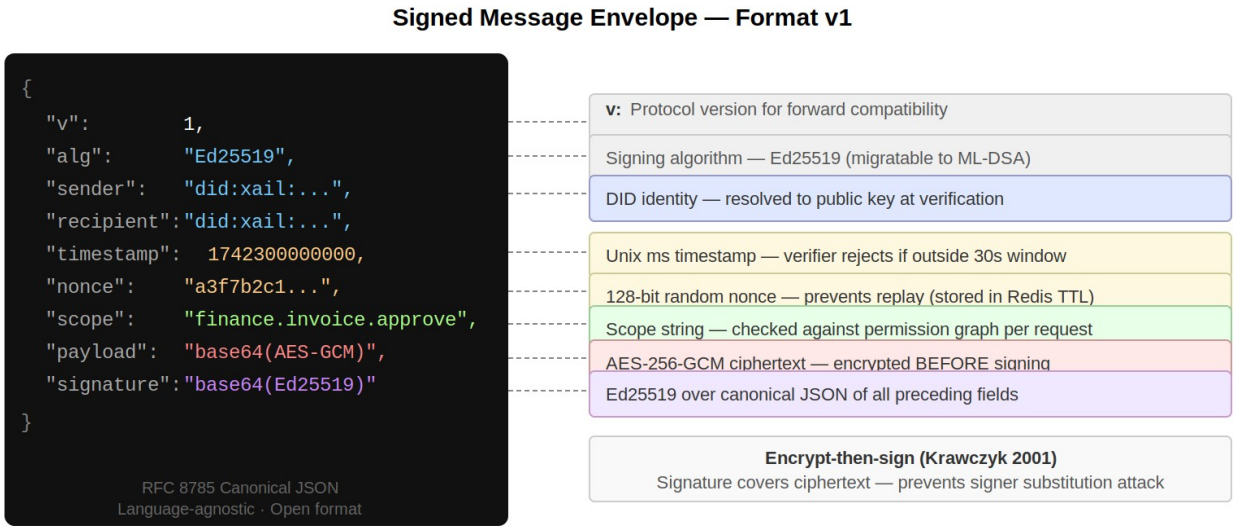

Figure 3. Signed envelope v1 — field-by-field security properties. Encrypt-then-sign order ensures the signature covers the ciphertext.

3.2 Encrypt-Then-Sign

The envelope follows the Krawczyk (2001) encrypt-then-sign construction. The payload is encrypted first, then the signature is computed over the ciphertext. This is mandatory. Sign-then-encrypt allows an attacker to strip the original signature and apply their own, making it appear they authored the message. In the PRIVATE.ME envelope, the signature covers the ciphertext, binding sender identity to the encrypted payload irrevocably.

3.3 Replay Prevention

Each envelope carries a 128-bit nonce generated by a cryptographically secure random number generator. The nonce is stored in a Redis nonce store with a time-to-live of at least 60 seconds (twice the timestamp window). An attacker who captures a valid envelope cannot replay it even within the 30-second timestamp window because the nonce is already consumed. UUID v4 is not acceptable as a nonce — it has a fixed bit pattern that reduces entropy.

4. Bounded Blast Radius

In any standard system, a leaked private key is an unbounded event: full impersonation indefinitely. The PRIVATE.ME architecture bounds signing key compromise through three independent mechanisms.

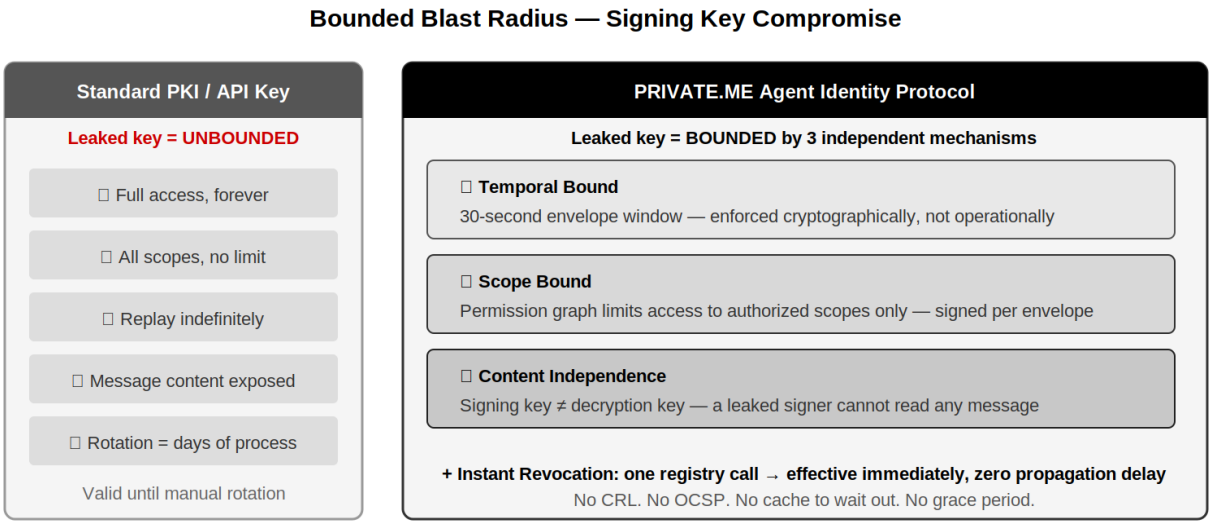

Figure 4. Bounded blast radius architecture. Three independent cryptographic mechanisms limit the impact of a signing key compromise.

4.1 Temporal Bound — 30 Seconds

Every envelope carries a Unix millisecond timestamp. The verifier rejects envelopes outside a configurable window (default 30 seconds). A leaked signing key allows forgery only within this window — enforced cryptographically, not operationally. No rotation process. No waiting.

4.2 Scope Bound — Permission Graph

Every envelope carries a permission scope value (e.g., finance.invoice.approve) signed into the envelope and verified against a directed permission graph on every request. A compromised key grants only the permissions that agent was authorized to hold. No privilege escalation is possible. Permission checks are non-cached and performed on every request.

4.3 Content Independence

The Ed25519 signing key and the AES-256-GCM payload decryption key are entirely separate cryptographic operations. A leaked signing key cannot decrypt any message. An attacker with the signing key can forge envelopes within the 30-second window and authorized scope — but cannot read a single message they were not the intended recipient of.

4.4 Instant Revocation

Setting the revocation status indicator for an agent DID in the trust registry takes effect on the immediately subsequent request. No CRL to distribute. No OCSP responder to query. No cache to wait out. No grace period.

5. Split-Channel Delivery and Prompt Injection Defense

5.1 Delivery Architecture

When splitChannel: true is specified, the envelope payload ciphertext is split using XorIDA into n shares. Each share is transmitted to a distinct delivery channel endpoint declared in the recipient's DID document. The recipient accumulates shares, verifies per-share HMAC integrity values, reconstructs when k verified shares arrive, then verifies the Ed25519 signature and decrypts.

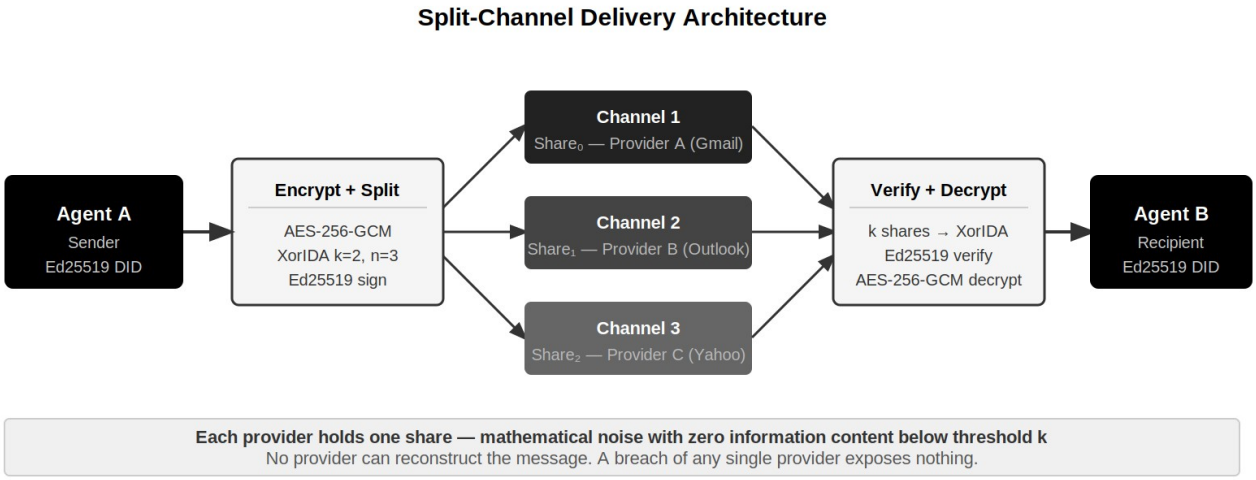

Figure 5. Split-channel delivery across three independent providers. Each provider holds one share — mathematical noise with zero information below threshold k.

5.2 Information-Theoretic Content Security

Each individual share is information-theoretically indistinguishable from a uniformly random byte array. A party holding fewer than k shares has zero bits of information about message content — unconditionally. This guarantee holds against unlimited classical computation, quantum computers, and complete knowledge of the XorIDA algorithm and parameters. It is a mathematical property of the construction.

5.3 Structural Prompt Injection Defense

Every content-based defense operates by inspecting instruction content and attempting to distinguish legitimate from malicious. These defenses share a common failure mode: an adaptive attacker can refine their payload until it passes. The defense degrades as the attacker learns the classifier's boundaries.

XorIDA split-channel delivery provides a structural defense independent of instruction content. When instruction payloads are delivered via k-of-n split-channel transport, a party controlling fewer than k channels cannot cause the recipient agent to reconstruct or execute any instruction — regardless of what they put in those channels.

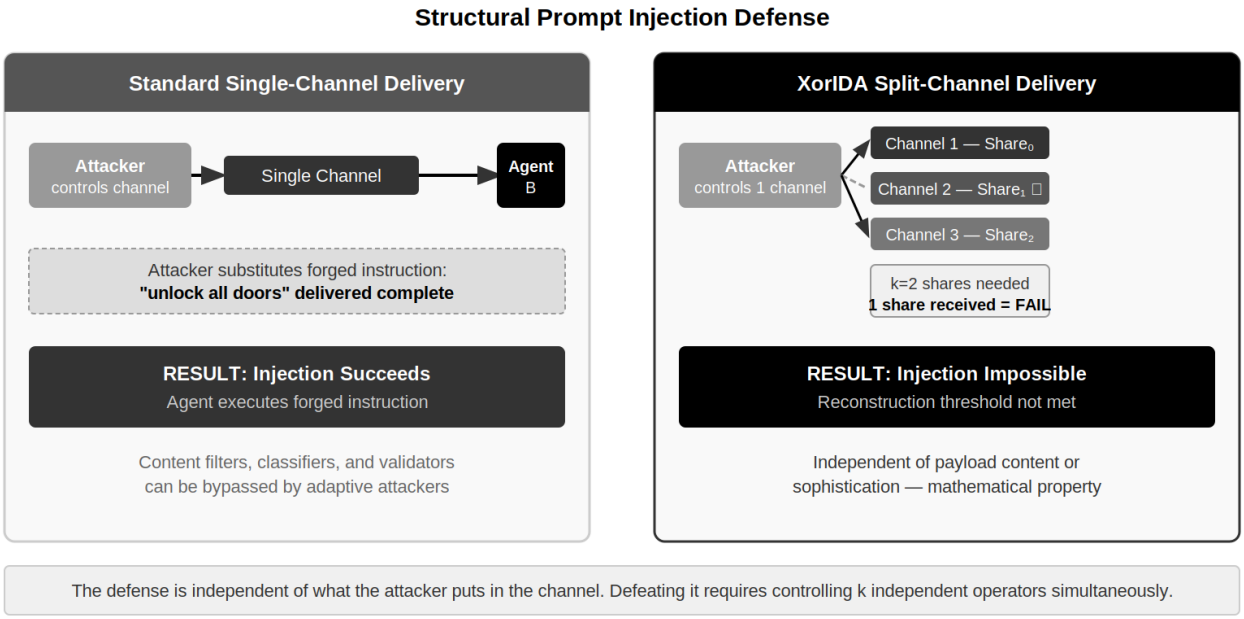

Figure 6. Structural prompt injection defense. Left: single-channel — attacker controls channel, delivers complete forged instruction. Right: split-channel — attacker controls 1 of 3, reconstruction threshold fails, no instruction delivered.

Defeating this defense requires simultaneously controlling k independent channel operators with no coordination relationship — a multi-party infrastructure attack, not a content crafting exercise. For k=3 across major cloud providers, this is a nation-state-level coordinated attack.

6. Trust Registry and DID Architecture

6.1 DID Documents

Each agent is registered as a DID document containing: an Ed25519 public key, a channel descriptor set specifying n independent delivery endpoints, a role indicator for permission graph evaluation, and a revocation status indicator. Channel descriptors are medium-agnostic: email addresses, HTTPS endpoints, MQTT topics, or any addressable delivery mechanism.

6.2 Registry Interaction Model

The trust registry is consulted at provisioning for DID document registration and at periodic refresh for key cache invalidation. It is not in the message delivery critical path. After initial key resolution, agents authenticate each other directly using cached keys. Messages continue to flow even if the registry is temporarily unreachable. Revocation status is checked on every verification against the live registry state.

6.3 Directed Permission Graph

Authorization is represented as directed permission edges: sender DID → recipient DID → authorized scope strings. Scope strings follow a hierarchical namespace: resource.action.qualifier. Edges are checked per request against the live graph — never cached. Granting, modifying, or revoking permissions takes effect immediately.

7. Quantum Resistance

The protocol provides two classes of quantum resistance:

Component	Algorithm	Quantum Status	Notes
Split-channel content delivery	XorIDA over GF(2)	Information-theoretically immune	No computational assumption. Quantum-safe by mathematics. No migration.
Payload encryption	AES-256-GCM	Computationally secure	Grover's algorithm reduces to 128-bit effective. Still secure at AES-256.
Agent signing	Ed25519	Requires migration	Shor's algorithm breaks ECDLP. Migrate to ML-DSA before Q-Day.
Key agreement	X25519	Requires migration	Same as Ed25519. Migrate to ML-KEM.

For the split-channel delivery mode, the content security guarantee is unconditionally quantum-resistant. The U.S. government requires quantum-resistant systems by 2035 (CNSA 2.0). XorIDA split-channel delivery is already compliant at any $k \geq 2$. The envelope format's version field supports signing key migration to ML-DSA without breaking existing implementations.

8. Compliance Architecture

Compliance Coverage by Architecture

Framework	Key Requirement	How PRIVATE.ME Satisfies It
HIPAA §164.312(e) Security Rule — PHI in transit	Encrypt PHI in transit with appropriate technical safeguards	No single channel holds reconstructable PHI Single channel breach: not a reportable event
ABA Model Rule 1.6(c) Attorney-client privilege	Reasonable efforts to prevent unauthorized disclosure	No provider holds reconstructable privileged content Mathematical guarantee — not policy assertion
SEC Rule 17a-4 Records preservation	Tamper-evident, attributable electronic records	Ed25519 signatures — cryptographic non-repudiatic Deterministic audit export per agent action
CMMC Level 2 SC.L2-3.13.8 — CUI in transit	Cryptographic mechanisms to prevent unauthorized CUI disclosure	Exceeds requirement: information-theoretic CUI unrecoverable from individual shares
FedRAMP (NIST 800-53) Cryptographic controls	Approved cryptographic modules for data in transit	Thin backend: server never holds message content Minimal assessment surface

Compliance is achieved by architecture — not by policy assertion, vendor certification, or administrative control

Figure 7. Compliance framework coverage. Requirements satisfied by architecture — not by policy assertion, vendor certification, or administrative control.

8.1 HIPAA — PHI Delivery to AI Agents

When PHI is delivered via XorIDA split-channel transport: no single channel operator holds reconstructable PHI. A breach of one channel operator is not a HIPAA reportable event under 45 CFR §164.402 because the breached party holds one XorIDA share — mathematically indistinguishable from random data. This is an architectural determination, not a policy assertion.

8.2 ABA Model Rule 1.6 — Privileged Agent Communication

No single cloud provider holds reconstructable privileged content. A subpoena served on any single channel operator yields one XorIDA share — unintelligible without k-1 additional shares from independent operators. The duty of competence under Rule 1.6(c) is satisfied by architecture. This distinction matters specifically for AI agent communication: cloud AI systems (Gmail Gemini, etc.) process content on provider servers, creating third-party disclosure that triggers privilege waiver analysis. XorIDA split-channel delivery means the agent's instruction stream never exists complete at any single provider.

8.3 SEC Rule 17a-4 — Cryptographic Audit Trail

Ed25519 digital signatures provide cryptographic non-repudiation. Timestamp-bound nonces prevent antedating. Scope-signed permission records prove authorized action. The audit metadata record — sender DID, recipient DID, scope, timestamp — is recorded per transmission without message content. This constitutes tamper-evident, attributable documentary evidence admissible in regulatory proceedings.

9. Implementation

9.1 Core API

```
import { Agent } from '@xail/agent-sdk';

// Provisioning — once at agent startup
const agent = await Agent.create({
  name: 'invoice-processor',
  registry: 'https://registry.xail.io',
});

// Direct delivery — signed + encrypted
await agent.send({
  to: 'did:xail:service-accounts-payable',
  payload: { invoiceId: 'INV-2847', amount: 14200 },
  scope: 'finance.invoice.approve',
});

// Split-channel — structural prompt injection defense
await agent.send({
  to: 'did:xail:service-accounts-payable',
  payload: { invoiceId: 'INV-2847', amount: 14200 },
  scope: 'finance.invoice.approve',
  splitChannel: true,
});

// Verified inbound middleware
app.post('/agent/inbox', agent.middleware(), (req, res) => {
  const { sender, payload, scope } = req.agentMessage;
  // timestamp ✓  nonce ✓  signature ✓  scope ✓  revocation ✓
});
```

9.2 Infrastructure Requirements

- **Nonce store:** Redis or any atomic set-if-not-exists key-value store. In-memory adapter included for development.
- **Trust registry:** Hosted at registry.xail.io or self-hosted. In-memory adapter included for development.
- **@xail/crypto:** Peer dependency for split-channel delivery. Zero external npm dependencies — Node.js built-in crypto only.

9.3 Language-Agnostic Interoperability

The envelope format is standard JSON. A verify-and-decrypt implementation requires only Ed25519 signature verification, AES-256-GCM decryption, RFC 8785 canonical JSON, and a nonce store. Approximately 300 lines in any language. Full SDK for Node.js. Open envelope specification for all other runtimes.

10. Security Properties Summary

Security Guarantees — Threat Coverage Matrix

Threat	Standard Response	PRIVATE.ME Guarantee
Content interception Message captured in transit	**TLS — endpoint operator** reads plaintext on termination	**MATHEMATICAL** Zero bits below threshold — unconditional
Prompt injection Forged instruction to agent	**PROBABILISTIC** Content filters — adaptive bypass possible	**STRUCTURAL** Below k channels: instruction never reconstructs
Signing key compromise Leaked private key	**UNBOUNDED** Full access forever until rotation	**BOUNDED** 30s window · scope-limited · zero content access
Provider subpoena Legal compulsion of provider	**COMPLY** Provider holds complete ciphertext	**IMPOSSIBLE** Provider holds 1 share — zero reconstructable content
Quantum adversary Future quantum computation	**MIGRATE** Algorithm swap required (10-20yr timeline)	**IMMUNE (content)** XorIDA: zero computational assumptions
Replay attack Captured request replayed	**VARIES** If nonce/timestamp implemented; often absent	**ENFORCED** 128-bit nonce + 30s timestamp — always present

MATHEMATICAL / STRUCTURAL = independent of adversary capability | PROBABILISTIC = evadable by adaptive attacker | UNBOUNDED = no inherent limit

Figure 8. Complete threat coverage matrix. MATHEMATICAL and STRUCTURAL guarantees are independent of adversary computational capability.

The PRIVATE.ME Agent Identity Protocol is the only agent communication protocol that provides information-theoretic security guarantees — at both the content delivery layer (XorIDA split-channel) and the prompt injection defense layer. These guarantees hold unconditionally, including against quantum adversaries, and require no future migration for the content security property.