Dual Privacy Architecture: Information-Theoretic Bounds on Agent Reconstruction

Mathematical Framework for Swordsman-Mage Separation

privacymage
`0xagentprivacy`

December 11, 2025
Version 3.4

Abstract

We introduce the Swordsman and Mage as fundamental privacy primitives for dual-agent architectures, establishing rigorous information-theoretic bounds when conditional independence $(Y_S \perp\!\!\!\perp Y_M)|X$ is enforced between these agents' observations. The Swordsman (S) controls privacy boundaries through selective measurement, while the Mage (M) projects delegated agency using only S-authorized observations.

Formal Semantic Foundation: We ground this architecture in Promise Theory (Bergstra & Burgess, 2019), which provides established semantics for autonomous agent coordination. The autonomy axiom—that agents can only promise their own behavior—formally explains why single-agent architectures cannot resolve the privacy-delegation paradox. The First Person + Swordsman + Mage system forms a *superagent* with *interior promises* between components, and The Gap (the reconstruction ceiling) is formally an *irreducible promise*—a property that emerges from component cooperation but cannot be attributed to any single agent.

Proven Results: We prove that this separation enables an additive bound on mutual information: $I(X; Y_S, Y_M) \leq I(X; Y_S) + I(X; Y_M)$. Combined with budget constraints $C_S + C_M < H(X)$, this establishes a reconstruction ceiling $R_{\max} < 1$ that no adversary can exceed regardless of computational resources. Via Fano's inequality, we establish a fundamental error floor: $P_e \geq 1 - \frac{I(X;Y)+1}{H(X)}$, guaranteeing minimum reconstruction error when $R_{\max} < 1$. We further prove graceful degradation under approximate separation.

Implementation Framework: We provide practical budget estimation methods, isolation verification protocols, and side-channel resistance models based on covert channel analysis. We integrate zero-knowledge proof systems for cryptographic enforcement of separation and budget compliance, providing concrete constructions using Groth16, PLONK, and Nova protocols.

Theoretical Predictions: We present theoretical conjectures about potential optimal allocation patterns, including a golden ratio hypothesis ($\phi \approx 1.618$) and tetrahedral emergence properties. These remain unproven mathematical conjectures requiring both formal proof and empirical validation.

Contents

1 Nature of This Work

What Is Proven: The core information-theoretic results (additive bounds under separation, reconstruction ceilings, error floors) are rigorously proven using established information theory.

What Is Grounded in Established Theory: The Promise Theory foundations draw from peer-reviewed work by Bergstra & Burgess (2019), providing formal semantics for the dual-agent architecture without requiring novel theoretical claims.

What Is Theoretical: The golden ratio optimization hypothesis and tetrahedral emergence predictions are unproven mathematical conjectures. They represent interesting theoretical possibilities but have not been formally derived from first principles.

What Is Missing: No implementations exist. No empirical data has been collected. No observations have been made. This is purely theoretical and mathematical work at present.

What We Seek: Collaboration from theorists to prove or disprove the conjectures, and from practitioners to build implementations and collect empirical data.

2 Introduction

2.1 Motivation

The deployment of autonomous AI agents acting on behalf of humans creates a fundamental tension: agents require information about their principals to act effectively (delegation), yet this same information enables reconstruction of sensitive behavioral patterns (privacy loss). Traditional single-agent architectures cannot resolve this tension—the same system handling both functions creates an inherent conflict of interest.

Promise Theory Insight: This conflict is not merely architectural but semantic. Promise Theory's autonomy axiom states that *an agent can only make promises about its own behavior.* A single agent attempting to promise both perfect protection AND full delegation violates this axiom—it promises in domains it cannot independently control.

We propose the Swordsman and Mage as dual privacy primitives that resolve this tension through architectural separation:

- **The Swordsman (S)**: A privacy-enforcement primitive that controls information disclosure through selective measurement
- **The Mage (M)**: A delegation primitive that projects agency using only Swordsman-authorized observations

The key insight: enforcing conditional independence between the Swordsman and Mage observations creates provable reconstruction bounds.

2.2 Contributions

Proven Results:

- **Separation Lemma (Theorem 5.1)**: Under $(Y_S \perp\!\!\!\perp Y_M)|X$, mutual information becomes additive
- **Reconstruction Ceiling (Corollary 5.2)**: With $C_S + C_M < H(X)$, reconstruction efficiency $R_{\max} < 1$
- **Error Floor (Theorem 5.3)**: Fano's inequality establishes minimum error $P_e \geq 1 - \frac{I(X;Y)+1}{H(X)}$

- **Robustness Analysis (Theorem 5.4)**: ε-approximate separation degrades bounds gracefully

Semantic Foundation:

- Promise Theory Grounding from Bergstra & Burgess (2019)
- Autonomy Axiom Application explaining single-agent failure
- Superagent Architecture: First Person + S + M as composite agent
- Irreducible Promise: The Gap ($R_{\max} < 1$) as emergent property

Implementation Framework:

- Practical budget estimation and monitoring methods
- Isolation verification and enforcement protocols
- Zero-knowledge proof constructions for cryptographic enforcement

2.3 Related Work

This work differs from existing privacy frameworks:

- **Differential Privacy** [Dwork & Roth 2014]: Adds calibrated noise; we enforce structural separation
- **Secure Multi-Party Computation** [Goldreich 2004]: Distributes computation; we distribute observation rights
- **Information Flow Control** [Sabelfeld & Myers 2003]: Tracks taint; we bound reconstruction
- **Zero-Knowledge Proofs** [Groth 2016]: Verifiable computation; we apply to privacy budget enforcement
- **Promise Theory** [Bergstra & Burgess 2019]: Autonomous agent semantics; we apply to privacy architecture

3 Promise-Theoretic Foundations

Promise Theory, as developed by Bergstra & Burgess (2019), provides formal semantics for autonomous agent systems.

3.1 The Autonomy Axiom

Autonomy Axiom (Promise Theory): An agent can only make promises about its own behavior. No agent can make a promise on behalf of another agent.

Application to Privacy Architecture:
This axiom formally explains why single-agent architectures fail:

- A single agent attempting to promise both “I will protect all your data” AND “I will effectively delegate on your behalf” must promise outcomes that depend on external responses

- The delegation promise requires coordination with external agents whose behavior the single agent cannot control
- These conflicting promises cannot be kept simultaneously by a single agent

The dual-agent architecture resolves this:

- Swordsman promises: “I will enforce boundaries” (its own behavior)
- Mage promises: “I will coordinate using only authorized information” (its own behavior)
- Neither promises on behalf of the other

3.2 Superagent Structure

Definition 3.1 (Superagent). *A composite agent with interior promises between components and exterior promises to the outside world.*

The First Person + Swordsman + Mage system forms a superagent with:
Interior Promises (within superagent):

- $\mathrm{S} \xrightarrow{\text{protect}} \mathrm{FP}$: Swordsman promises protection to First Person
- $\mathrm{M} \xrightarrow{\text{delegate}} \mathrm{FP}$: Mage promises delegation to First Person
- $\mathrm{FP} \xrightarrow{\text{authorize}} \mathrm{S}, \mathrm{M}$: First Person authorizes both agents
- $\mathrm{S} \perp\!\!\!\perp \mathrm{M}$: Separation promise—no direct information flow

Exterior Promises (to world):

- Superagent coordinates with external world (via Mage’s public actions)
- Superagent enforces boundaries (via Swordsman’s rejections)

3.3 The Gap as Irreducible Promise

Definition 3.2 (Irreducible Promise). *A promise of a superagent that cannot be attributed to any single component agent, but requires their cooperation.*

Proposition 5.5 (Irreducibility of The Gap). *The reconstruction ceiling $R_{\max} < 1$ is an irreducible property of the First Person superagent in the sense of Promise Theory.*
Informal Argument:

1. The Swordsman alone cannot achieve $R_{\max} < 1$ (needs information budget limit)
2. The Mage alone cannot achieve $R_{\max} < 1$ (has no privacy enforcement capability)
3. The First Person alone cannot achieve $R_{\max} < 1$ (needs operational agents)
4. Only the cooperation of all three—with maintained separation—achieves $R_{\max} < 1$

Formal Status: A rigorous proof would require demonstrating that no promise-respecting decomposition of the superagent can achieve $R_{\max} < 1$ through component promises alone. This formalization is left as future work; the intuitive argument suffices for architectural motivation.

Why This Matters: Irreducible promises cannot be captured by compromising any single component. An adversary who fully compromises the Swordsman learns only C_S bits. An adversary who fully compromises the Mage learns only C_M bits. Neither captures the irreducible promise—it exists in the *relationship* between components, not in any component itself.

3.4 Promise Types and Agent Roles

Agent	(+) Give Promises	(-) Accept Promises
Swordsman	Protection, boundaries	Authorization from FP
Mage	Delegation, coordination	Authorized info from S
First Person	Authorization, sovereignty	Protection from S, Delegation from M

3.5 Budget Constraints as Valency

Promise Theory defines **valency** as the exclusive attention capacity an agent can dedicate to promises.

- C_S is Swordsman's information valency—maximum mutual information it can reveal
- C_M is Mage's information valency—maximum mutual information its actions can leak
- $C_S + C_M < H(X)$ is the system valency constraint

3.6 Assessment and Trust

Promise Theory defines **assessment** $\alpha(\pi)$ as determination whether a promise was kept.

Tier	Signals	Trust Value
Blade	0–50	0.0–0.2
Light	50–150	0.2–0.5
Heavy	150–500	0.5–0.8
Dragon	500+	0.8–1.0

Threshold Rationale: These tier thresholds are initial design parameters based on expected engagement patterns:

- **Blade→Light (50 signals):** ∼2 months at moderate activity, sufficient to distinguish genuine engagement
- **Light→Heavy (150 signals):** ∼6 months sustained commitment
- **Heavy→Dragon (500 signals):** ∼12+ months extended track record

These should be calibrated through empirical observation.

3.7 Implications for Proven Results

The Promise Theory framework provides semantic grounding for our information-theoretic results:

Proven Result	PT Grounding
Separation Lemma (Thm 5.1)	Scope non-overlap enforced by promise structure
Reconstruction Ceiling (Cor 5.2)	System valency constraint limits total revelation
Error Floor (Thm 5.3)	Irreducible promise property—cannot be captured by component compromise
Robustness (Thm 5.4)	Graceful degradation from approximate scope overlap

This grounding elevates the results from "clever engineering" to "rigorous implementation of established autonomous systems theory."

4 Model and Preliminaries

4.1 Basic Framework

Let X be a secret over finite alphabet $\mathcal{X}$ with $H(X) > 0$. Two agents produce observations:

$$Y_S = E_S(X, N_S) \tag{1}$$
$$Y_M = E_M(X, N_M) \tag{2}$$

where N_S, N_M are independent local randomness sources.

4.2 The Swordsman and Mage Primitives

Definition 4.1 (Swordsman Primitive)**.** *The Swordsman S is a privacy-enforcement agent characterized by:*

- *Measurement function E_S that implements selective disclosure*
- *Information budget C_S controlling maximum leakage: $I(X; Y_S) \leq C_S$*
- *Primary objective: minimize reconstruction while enabling necessary delegation*

Definition 4.2 (Mage Primitive)**.** *The Mage M is a delegation agent characterized by:*

- *Projection function E_M operating on S-authorized information*
- *Information budget C_M for capability execution: $I(X; Y_M) \leq C_M$*
- *Primary objective: maximize utility under privacy constraints*

The critical architectural requirement: $(Y_S \perp\!\!\!\perp Y_M)|X$ (conditional independence).

4.3 Formal Definitions

Definition 4.3 (Separation Condition)**.** *The architecture enforces $(Y_S \perp\!\!\!\perp Y_M)|X$.*

Definition 4.4 (Information Budgets)**.** *$I(X; Y_S) \leq C_S$ and $I(X; Y_M) \leq C_M$.*

Definition 4.5 (Reconstruction Efficiency)**.** $R \triangleq \frac{I(X;Y)}{H(X)} \in [0, 1]$.

4.4 Threat Model

Assumptions:

- Passive adversary observing (Y_S, Y_M)
- Separation enforced through architectural boundaries
- Known distributions $P(X)$, encoding functions E_S, E_M

Explicitly Out of Scope (with justification):

- **Active attacks modifying agent behavior:** The ZKP constructions in §7.5 provide cryptographic enforcement that resists some active attacks. However, attacks that compromise the execution environment entirely (e.g., malicious hardware) remain out of scope. Future work should integrate TEE-based attestation.

- **Side-channels on separation mechanism itself:** Timing attacks on the separation boundary could leak information about which agent processed which query. Mitigation requires constant-time separation protocols. §6 addresses covert channel capacity bounds but does not fully model this threat.

- **Temporal correlation across sessions:** Adversaries observing patterns across sessions may extract additional information. The current analysis treats each session independently. Extending to session-correlated adversaries requires analyzing mutual information across time: $I(X; Y_{1:T})$ rather than single-session $I(X; Y)$.

Applicability Statement: The proven guarantees hold for passive adversaries in single-session contexts with cryptographically enforced separation. Real deployments should evaluate which excluded threats apply to their context and implement additional mitigations.

Promise Theory Note: This threat model assumes agents *keep* their promises. Active attacks that cause promise violation (e.g., forcing M to observe S's outputs) would break the architecture. The ZKP constructions address cryptographic enforcement of promise-keeping.

5 Core Theory and Proven Results

5.1 The Separation Lemma

Theorem 5.1 (Additive Bound Under Separation)**.** *If* $(Y_S \perp\!\!\!\perp Y_M)|X$ *holds, then:*

$$I(X; Y_S, Y_M) \leq I(X; Y_S) + I(X; Y_M) \tag{3}$$

Proof. By the chain rule for mutual information:

$$I(X; Y_S, Y_M) = I(X; Y_S) + I(X; Y_M|Y_S) \tag{4}$$

Under conditional independence $(Y_S \perp\!\!\!\perp Y_M)|X$, we have $I(Y_M; Y_S|X) = 0$, which implies:

$$H(Y_M|X, Y_S) = H(Y_M|X) \tag{5}$$

Therefore:

$$\begin{aligned} I(X; Y_M|Y_S) &= H(Y_M|Y_S) - H(Y_M|X, Y_S) && (6)\\ &= H(Y_M|Y_S) - H(Y_M|X) && (7)\\ &\leq H(Y_M) - H(Y_M|X) && (8)\\ &= I(X; Y_M) && (9) \end{aligned}$$

□

Corollary 5.2 (Reconstruction Ceiling)**.** *If* $C_S + C_M < H(X)$*, then* $R_{\max} = \frac{C_S + C_M}{H(X)} < 1$.

Critical Clarification: Separation alone is insufficient. The ceiling requires **BOTH** separation (for additivity) **AND** budget constraints (for the bound).

5.2 Error Lower Bound

Theorem 5.3 (Fano-Based Error Floor)**.** *For any estimator* $\hat{X}(Y)$ *and finite alphabet* $\mathcal{X}$*:*

$$P_e \triangleq \Pr[\hat{X}(Y) \neq X] \geq \frac{H(X|Y) - 1}{\log(|\mathcal{X}| - 1)} \tag{10}$$

For large alphabets where $\log(|\mathcal{X}| - 1) \approx H(X)$*:*

$$P_e \gtrsim 1 - \frac{I(X;Y) + 1}{H(X)} = 1 - R - \frac{1}{H(X)} \tag{11}$$

Proof. Apply Fano's inequality. For any estimator $\hat{X}(Y)$:

$$H(X|Y) \leq h(P_e) + P_e \cdot \log(|\mathcal{X}| - 1) \tag{12}$$

where $h(\cdot)$ is binary entropy. Since $h(P_e) \leq 1$:

$$H(X|Y) \leq 1 + P_e \cdot \log(|\mathcal{X}| - 1) \tag{13}$$

Rearranging and using $I(X;Y) = H(X) - H(X|Y)$:

$$P_e \geq \frac{H(X) - I(X;Y) - 1}{\log(|\mathcal{X}| - 1)} \geq 1 - \frac{I(X;Y) + 1}{H(X)} \tag{14}$$

□

Interpretation: When $R_{\max} = 0.7$, then $P_e \geq 0.3 - O(1/H(X)) \approx 0.3$ for large entropy.

5.3 Robustness to Approximate Separation

Theorem 5.4 (ε-Approximate Separation). *If $I(Y_S; Y_M|X) \leq \varepsilon$ (approximate separation), then:*

$$I(X; Y_S, Y_M) \leq I(X; Y_S) + I(X; Y_M) + \varepsilon \tag{15}$$

Proof. Following the chain rule decomposition:

$$I(X; Y_M|Y_S) = H(Y_M|Y_S) - H(Y_M|X, Y_S) \tag{16}$$

With approximate independence:

$$H(Y_M|X, Y_S) \geq H(Y_M|X) - I(Y_S; Y_M|X) \geq H(Y_M|X) - \varepsilon \tag{17}$$

Therefore:

$$I(X; Y_M|Y_S) \leq H(Y_M) - H(Y_M|X) + \varepsilon = I(X; Y_M) + \varepsilon \tag{18}$$

□

6 Side-Channel Analysis and Robustness

6.1 Connection to Covert Channel Theory

Our analysis builds on established covert channel capacity results. For d side-channel observations:

$$R(d) = R_{\max} \cdot \frac{\ln(1 + d/d_0)}{\ln(1 + d_{\max}/d_0)} \tag{19}$$

This logarithmic form emerges from:

- Temporal correlation in behavioral patterns
- Finite substrate entropy $H(X)$
- Measurement quantization effects

6.2 Validation Methodology

To verify theoretical guarantees in practice:

Simulation Framework:

- Generate test distributions with known $H(X)$
- Implement S and M with controlled coupling
- Measure actual $I(X; Y_S, Y_M)$ vs theoretical bounds

Violation Detection Protocol:

- Monitor $I(Y_S; Y_M|X)$ in real-time
- Flag when coupling exceeds threshold ε
- Trigger corrective isolation measures

7 Implementation Framework

7.1 Entropy Estimation for Behavioral Data

Budget constraints require estimating $H(X)$, the entropy of the private state. For behavioral data, this is notoriously difficult.

Recommended Estimators:

k-NN Estimators (KSG): For continuous variables, the Kozachenko-Leonenko / Kraskov-Stögbauer-Grassberger estimator provides consistent estimates:

```
def estimate_entropy_ksg(samples, k=3):
    """
␣␣␣␣k-NN␣entropy␣estimator␣(Kraskov␣et␣al.,␣2004).

␣␣␣␣Args:
␣␣␣␣␣␣␣␣samples:␣Array␣of␣shape␣(n_samples,␣dim)
␣␣␣␣␣␣␣␣k:␣Number␣of␣neighbors␣(default:␣3)

␣␣␣␣Returns:
␣␣␣␣␣␣␣␣Estimated␣H(X)␣in␣bits
␣␣␣␣"""
    from scipy.spatial import KDTree
    from scipy.special import digamma
    import numpy as np

    n, d = samples.shape
    tree = KDTree(samples)

    # Find k-th neighbor distances
    distances, _ = tree.query(samples, k=k+1)
    eps = distances[:, -1]  # k-th neighbor distance

    # KSG estimator
    H = digamma(n) - digamma(k) + d * np.mean(np.log(2 * eps))
    return H / np.log(2)  # Convert to bits
```

Histogram-based: For discrete behavioral categories:

```
def estimate_entropy_histogram(action_history, num_categories=100):
    """
␣␣␣␣Histogram-based␣entropy␣estimator␣for␣discrete␣data.

␣␣␣␣Warning:␣Underestimates␣true␣entropy␣if␣rare␣behaviors␣not␣observed
    .
␣␣␣␣Add␣safety␣margin:␣use␣H_estimate␣*␣1.2␣for␣budget␣calculations.
␣␣␣␣"""
    from scipy.stats import entropy
    import numpy as np

    counts, _ = np.histogram(action_history, bins=num_categories)
    probs = counts / counts.sum()
    probs = probs[probs > 0]  # Remove zeros
    return entropy(probs, base=2)
```

MINE/InfoNCE: Neural estimators for high-dimensional data when other methods fail.

Practical Guidance:

Limitations: All estimators provide lower bounds on true entropy. For privacy guarantees, use conservative (higher) estimates with safety margins.

Recommended Practice:

1. Estimate $H(X)$ using multiple methods
2. Take the maximum estimate
3. Add 20% safety margin: $H_{\text{budget}} = 1.2 \times \max(\text{estimates})$
4. Re-estimate periodically as behavioral patterns change

7.2 Mutual Information Estimation

The implementation requires estimating $I(X;Y)$ for budget monitoring.

1. KSG Estimator (Non-parametric): Best for moderate-dimensional data. No training required.

```
def estimate_mutual_info_ksg(X, Y, k=3):
    """
␣␣␣␣K-nearest␣neighbor␣MI␣estimator␣(Kraskov␣et␣al.,␣2004).

␣␣␣␣Args:
␣␣␣␣␣␣␣␣X:␣Private␣state␣samples,␣shape␣(n_samples,␣dim_X)
␣␣␣␣␣␣␣␣Y:␣Observation␣samples,␣shape␣(n_samples,␣dim_Y)
␣␣␣␣␣␣␣␣k:␣Number␣of␣neighbors␣(default:␣3)

␣␣␣␣Returns:
␣␣␣␣␣␣␣␣Estimated␣I(X;␣Y)␣in␣bits,␣with␣confidence␣interval
␣␣␣␣"""
    from sklearn.feature_selection import mutual_info_regression
    import numpy as np

    # For discrete X, use mutual_info_classif
    # For continuous X, use mutual_info_regression
    mi = mutual_info_regression(Y, X.ravel(), n_neighbors=k)

    # Bootstrap for confidence interval
    n_bootstrap = 100
    mi_samples = []
    n = len(X)
    for _ in range(n_bootstrap):
        idx = np.random.choice(n, n, replace=True)
        mi_boot = mutual_info_regression(Y[idx], X[idx].ravel(),
           n_neighbors=k)
        mi_samples.append(mi_boot.mean())

    ci_low, ci_high = np.percentile(mi_samples, [5, 95])
    return mi.mean(), (ci_low, ci_high)
```

2. MINE (Neural Estimation): Best for high-dimensional continuous data. Requires training.

```
# Requires: pip install pytorch-mine
def estimate_mutual_info_mine(X, Y, hidden_dim=100, epochs=100):
    """
␣␣␣␣Mutual␣Information␣Neural␣Estimation.

␣␣␣␣Use␣for␣high-dimensional␣data␣where␣KSG␣fails.
␣␣␣␣See␣Belghazi␣et␣al.,␣2018␣for␣details.
␣␣␣␣"""
    # Implementation uses neural network to estimate MI lower bound
    pass
```

3. Binned Estimator (Fast, Approximate): Simplest and fastest. Use for quick runtime checks.

```
def estimate_mutual_info_binned(X, Y, bins=20):
    """
␣␣␣␣Binned␣MI␣estimator.␣Fast␣but␣loses␣precision.
␣␣␣␣"""
    import numpy as np
    from scipy.stats import entropy

    # Discretize continuous variables
    X_binned = np.digitize(X, np.linspace(X.min(), X.max(), bins))
    Y_binned = np.digitize(Y, np.linspace(Y.min(), Y.max(), bins))

    # Compute joint and marginal distributions
    joint, _, _ = np.histogram2d(X_binned, Y_binned, bins=bins)
    joint = joint / joint.sum()

    px = joint.sum(axis=1)
    py = joint.sum(axis=0)

    # MI = H(X) + H(Y) - H(X,Y)
    H_x = entropy(px[px > 0], base=2)
    H_y = entropy(py[py > 0], base=2)
    H_xy = entropy(joint.ravel()[joint.ravel() > 0], base=2)

    return H_x + H_y - H_xy
```

Confidence Bounds: All estimators have variance. For budget enforcement:

1. Compute point estimate and confidence interval
2. Use **upper confidence bound** for budget tracking
3. Alert when upper bound approaches budget limit
4. Refuse disclosure when upper bound exceeds limit

7.3 Budget Estimation and Management

```
# Budget estimation
def estimate_mutual_info(samples_X, samples_Y):
    # Use MINE or InfoNCE estimators
    # Return confidence interval
    return I_estimate, confidence_bounds
```

Runtime Monitoring:

- Track cumulative information release
- Implement privacy ledger similar to differential privacy
- Trigger alerts when approaching budget limits

7.4 Adaptive Control Framework

```
# Adaptive budget controller
class AdaptiveBudgetController:
    def __init__(self, C_S_max, C_M_max):
        self.budget_S = C_S_max
        self.budget_M = C_M_max

    def adjust_granularity(self, current_usage):
        if current_usage > 0.8 * self.budget_S:
            return reduced_precision_mode
```

7.5 ZKP-Enhanced Selective Disclosure

The Swordsman's selective disclosure can be implemented using modern zero-knowledge proof systems:

- **Groth16**: O(1) proof size, fast verification (∼2ms), requires trusted setup
- **PLONK**: Universal trusted setup, flexible constraint systems
- **Nova**: No trusted setup, efficient recursive proof composition

Concrete Construction: For attribute disclosure:

$$\pi_{\text{attr}} = \text{ZKP}\{y_S = E_S(x) \wedge f(y_S) = 1\} \tag{20}$$

where f is a public predicate (e.g., "age ≥ 18").

7.6 Cryptographic Separation Verification

Rather than relying solely on trusted isolation, we can prove separation cryptographically:

$$\pi_{\text{sep}} = \text{ZKP}\{(Y_S \perp\!\!\!\perp Y_M)|X\} \tag{21}$$

Using techniques from zero-knowledge proof systems:

- Commit to both observations: $c_S = \text{Commit}(Y_S, r_S)$, $c_M = \text{Commit}(Y_M, r_M)$
- Prove conditional independence via joint distribution commitments
- Enable third-party verification of architectural compliance

7.7 ZKP-Based Budget Compliance

Zero-knowledge proofs enable verifiable budget tracking:

Protocol:

1. Agent commits to observation: $c_i = \text{Commit}(y_i, r_i)$
2. Proves cumulative budget compliance: $\pi_{\text{budget}} = \text{ZKP}\{\sum I(X; y_i) \leq C_S\}$
3. Verifier checks without learning $\{y_i\}$

7.8 Zero-Knowledge Implementation Patterns

Range Proofs: For continuous variables, prove $y_S \in [a, b]$ without revealing exact value.

Set Membership: Prove attribute membership using Merkle tree commitments.

Predicate Verification: Prove arbitrary predicates $f(Y_S) = 1$ using circuit-based SNARKs.

Cumulative Budget Tracking: Using Nova for recursive composition:

$$\pi_1 = \text{ZKP}\{I(X; y_1) \leq C_S\} \tag{22}$$

$$\pi_t = \text{ZKP}\{\pi_{t-1} \wedge I(X; y_1, \ldots, y_t) \leq C_S\} \tag{23}$$

7.9 Implementation Checklist

Pre-deployment:

- □ Estimate $H(X)$ for target domain
- □ Set $C_S + C_M \leq 0.7 \cdot H(X)$ (safety margin)
- □ Implement separation enforcement
- □ Verify isolation properties
- □ Deploy monitoring infrastructure

Runtime:

- □ Track actual $I(X; Y_S)$ and $I(X; Y_M)$
- □ Monitor separation violations
- □ Log reconstruction attempts
- □ Adjust budgets adaptively

8 Theoretical Predictions (Unproven)

STATUS: PURELY THEORETICAL - This section presents unproven mathematical conjectures. No implementations, empirical data, or observations exist.

8.1 Golden Ratio Hypothesis

Conjecture 8.1 (Golden Ratio Optimality - UNPROVEN)**.** *There may exist an optimization principle that drives optimal allocation ratios toward* $\phi \approx 1.618$*.*

Theoretical Motivation: Consider value functions:

$$\max_{C_S, C_M} U(C_M) \cdot P(C_S, C_M) \quad \text{s.t.} \quad C_S + C_M = B \tag{24}$$

Under certain smoothness and monotonicity conditions, the optimal ratio might be $C_S/C_M = \phi$.

Promise Theory Perspective: If the golden ratio emerges, it would represent an optimal *valency allocation* between protection and delegation promises.

Status: Pure conjecture. No proof exists. No data exists.

8.2 Tetrahedral Emergence Hypothesis

Conjecture 8.2 (Tetrahedral Structure - HIGHLY SPECULATIVE)**.** *Sustained S-M separation may naturally generate two additional measurement properties:*

- ***Reflect (R)****: Temporal accumulation of S's boundary decisions*
- ***Connect (C)****: Network effects from M's delegation patterns*

Mathematical Formulation: If $(Y_S \perp\!\!\!\perp Y_M)|X$ is maintained over time:

$$Y_R = R(Y_S^{1:t}, \tau) \quad \text{[Memory from S history]} \tag{25}$$

$$Y_C = C(Y_M^{1:t}, G) \quad \text{[Network from M interactions]} \tag{26}$$

By data processing inequality: $I(X; Y_R) \leq I(X; Y_S)$ and $I(X; Y_C) \leq I(X; Y_M)$.

Promise Theory Consideration: N=4 agents would require O(16) interior promises. This complexity is only justified if emergent properties provide sufficient additional capability.

8.3 Testable Predictions

If these hypotheses hold in real systems, we would expect to observe:

- Allocation ratios clustering near $\phi \approx 1.618$
- Temporal patterns developing memory/logging behaviors
- Network effects emerging in inter-agent communication

Important: These are theoretical predictions, not observations.

9 Discussion

9.1 Summary of Proven Guarantees

We have rigorously established:

- Separation enables additive mutual information bounds
- Combined with budgets, guarantees $R_{\max} < 1$
- Fano's inequality ensures minimum error rates
- Approximate separation degrades gracefully
- ZKP constructions enable cryptographic enforcement

These results hold unconditionally, independent of computational assumptions.

9.2 Promise Theory Grounding

We have demonstrated that these results implement established autonomous systems theory:

- **Autonomy axiom** explains why single agents fail
- **Superagent structure** describes the First Person system
- **Irreducible promises** characterize The Gap
- **Scope separation** grounds conditional independence
- **Valency constraints** ground budget limits

9.3 Relationship to Existing Privacy Frameworks

Framework	**Focus**	**Our Approach**	**Synergy**
Differential Privacy	Statistical noise	Structural separation	Use DP within S
Secure MPC	Distributed computation	Distributed observation	Complementary
Information Flow Control	Taint tracking	Quantitative bounds	Enhanced metrics
Promise Theory	Agent semantics	Privacy architecture	Formal foundation

9.4 Limitations and Assumptions

Key Limitations:

- **Conditional Independence**: Hard to enforce perfectly in practice
- **Passive Adversary**: Active attacks not fully addressed
- **Known Distributions**: Uncertainty in $P(X)$ affects budgets
- **Static Budgets**: Dynamic environments may require adaptation

9.5 Experimental Roadmap

Immediate: Implement reference architecture, develop test suite, create monitoring tools.

Medium-term: Deploy in real applications, validate across domains, establish best practices.

Long-term: Prove or refute optimal allocation theorems, extend to multi-agent settings.

10 Related Extended Work

10.1 Privacy Technology Integration

Our framework complements:

- **Zero-knowledge proofs**: Implement S's selective disclosure (Groth16, PLONK, Nova)
- **Secure enclaves**: Hardware enforcement of separation
- **Homomorphic encryption**: Computation within M's bounds
- **Privacy pools**: Network effects without individual exposure

10.2 Economic Enforcement of Separation

The architectural separation can be economically enforced through dual-token markets:

- SWORD tokens earned exclusively through Swordsman chronicles
- MAGE tokens earned exclusively through Mage chronicles
- Market separation creates economic pressure against agent merger
- Guardian staking (10,000 SWORD) maintains collective standards

Signal-Based Sustainability:

- Genesis ceremony: 1 ZEC creates agent pair
- Ongoing signals: 0.01 ZEC each, continuous proof-of-comprehension
- Fee distribution: 61.8% transparent pool, 38.2% shielded pool

11 Conclusion

We have established rigorous information-theoretic bounds for dual-agent privacy architectures with enforced separation. The proven results—additive mutual information under separation, reconstruction ceilings below unity, and guaranteed error floors—provide solid foundations for privacy-preserving agent systems.

Promise Theory Foundation: We have grounded these results in Promise Theory (Bergstra & Burgess, 2019), demonstrating that the dual-agent structure is not merely an implementation choice but a formal requirement given the autonomy axiom. The Swordsman-Mage separation respects the autonomy axiom, and The Gap ($R_{\max} < 1$) emerges as an irreducible promise of the resulting superagent.

We integrate zero-knowledge proof systems as core implementation primitives, providing concrete constructions using Groth16, PLONK, and Nova protocols. This enables cryptographic rather than merely architectural enforcement of separation and budget constraints.

The key insight remains powerful: structural separation with budget constraints creates fundamental privacy guarantees independent of computational assumptions.

We present theoretical conjectures about golden ratio optimization and tetrahedral emergence, but emphasize these remain unproven mathematical hypotheses requiring validation.

Document Metadata

- **Project:** 0xagentprivacy
- **Version:** 3.4
- **Date:** December 11, 2025
- **Companion Documents:** Whitepaper v4.6, Promise Theory Reference v1.0, Glossary v2.2, First Person Project White Paper v1.1

Version History:

Version	Date	Changes
3.3	Dec 2025	Previous release
3.4	**Dec 11, 2025**	**Promise Theory integration, entropy-/MI estimation methodology, strengthened threat model, clarified speculative content, added KSG reference, relabeled irreducible promise as Proposition 5.5**

Version Statement

Version 3.4: This edition adds Promise Theory foundations (Bergstra & Burgess, 2019) as formal semantic grounding, entropy and mutual information estimation methodology, strengthened threat model discussion with explicit exclusions and justifications, and clarified speculative vs. proven content. Core information-theoretic results remain rigorous. Golden ratio hypotheses and tetrahedral emergence remain purely theoretical conjectures.

Acknowledgments

To the 0xagentprivacy project, BGIN, Kwaai, First Person Project, and Bergstra & Burgess for Promise Theory foundations.

References

[1] Bergstra, J.A. & Burgess, M. (2019). *Promise Theory: Principles and Applications.* O'Reilly Media.

[2] Cover, T.M. & Thomas, J.A. (2006). *Elements of Information Theory.* Wiley.

[3] Dwork, C. & Roth, A. (2014). The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in TCS.*

[4] Goldreich, O. (2004). *Foundations of Cryptography.* Cambridge University Press.

[5] Groth, J. (2016). On the Size of Pairing-based Non-interactive Arguments. *EUROCRYPT 2016.*

[6] Gabizon, A., Williamson, Z.J., & Ciobotaru, O. (2019). PLONK. ePrint 2019/953.

[7] Kothapalli, A., Setty, S., & Tzialla, I. (2021). Nova. ePrint 2021/370.

[8] Millen, J.K. (1987). Covert Channel Capacity. *IEEE S&P.*

[9] Sabelfeld, A. & Myers, A.C. (2003). Language-based Information-flow Security. *IEEE JSAC.*

[10] Fano, R.M. (1961). *Transmission of Information.* MIT Press.

[11] Shannon, C.E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal.*

[12] Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. *Physical Review E*, 69(6), 066138.

[13] Belghazi, M.I., et al. (2018). Mutual Information Neural Estimation. *ICML 2018.*

[14] The First Person Project. (2025). Building a Trust Layer for the Internet—One Person and One Community at a Time. *White Paper v1.1.*

A Appendix

A.1 Complete Chain Rule Expansion

For four variables:

$$I(X;Y_S,Y_M,Y_R,Y_C) = I(X;Y_S)+I(X;Y_M|Y_S)+I(X;Y_R|Y_S,Y_M)+I(X;Y_C|Y_S,Y_M,Y_R) \quad (27)$$

Each conditional term is bounded by the unconditional under independence assumptions.

A.2 Promise Theory Notation Summary

PT Concept	Symbol	0xagentprivacy Mapping
Promise	$A \xrightarrow{b} B$	Agent A promises behavior b to B
(+) give promise	$+b$	Swordsman/Mage promises to provide
(-) use promise	$-b$	Agent promises to use appropriately
Scope	$\sigma(A)$	Domain of A's valid promises
Valency	$v(A)$	A's exclusive promise capacity
Assessment	$\alpha(\pi)$	Chronicle verification, RPP compression
Superagent	$\mathcal{A}$	First Person + Swordsman + Mage
Irreducible promise	$\bar{\pi}$	$R_{\max} < 1$ (The Gap)

A.3 Implementation Pseudocode

```
# Basic dual-agent system with Promise Theory annotations
class DualAgentPrivacy:
    def __init__(self, entropy_bits, safety_factor=0.7):
        self.H_X = entropy_bits
        self.budget = self.H_X * safety_factor
        self.C_S = self.budget * 0.62  # Swordsman valency
        self.C_M = self.budget * 0.38  # Mage valency

    def measure_leakage(self):
        I_S = self.estimate_mutual_info(self.Y_S, self.X)
        I_M = self.estimate_mutual_info(self.Y_M, self.X)
        I_joint = self.estimate_mutual_info((self.Y_S, self.Y_M), self.
            X)
        separation_violation = I_joint - I_S - I_M
        return I_S, I_M, separation_violation
```

```
# Separation promise verification
def test_separation_violation(system, num_samples=10000):
    samples = []
    for _ in range(num_samples):
        x = sample_secret()
        y_s, y_m = system.observe(x)
        samples.append((x, y_s, y_m))

    # Compute I(Y_S; Y_M | X) - should be ~0 if promise kept
    violation = estimate_conditional_mi(samples)
    if violation > epsilon:
        return "PROMISE␣VIOLATION␣DETECTED", violation
    return "SEPARATION␣PROMISE␣MAINTAINED", violation
```