# Sealed Trade Protocol: A Peer-to-Peer Bilateral Trade System with Information-Sealed Negotiation

**Shota Nagafuchi**

sealed-trade@proton.me

## Abstract

*We propose a protocol for bilateral trade that reduces the leakage of private information during negotiation. In conventional private markets, the act of negotiating reveals private valuations which counterparties exploit to extract surplus. A buyer's willingness to pay, once signaled through an offer, can never be unsignaled. A seller's urgency, once revealed through a concession, permanently weakens their position. Sealed Trade addresses this by confining negotiation to AI agents operating within hardware-isolated enclaves. Each agent is bound by cryptographically signed parameters from its principal. Upon completion, the enclave is terminated and negotiation state is cleared. Only the final outcome — agreement or no deal — crosses the enclave boundary. Settlement occurs on-chain via bonded smart contracts using stablecoin as the payment rail.*

## 1. Introduction

Private bilateral markets — IP licensing, M&A, real estate — require negotiation to reach agreement. But negotiation itself is a destructive act: every offer, counteroffer, and hesitation reveals private information that the counterparty can exploit. The information required to reach a deal is the same information used to worsen its terms.

We call this the **double-use of private information**. When two parties negotiate, each party's private information — their reservation price, urgency, alternatives — is necessarily communicated during the process. Once communicated, this information can be used by the counterparty to extract surplus beyond what a fair exchange would produce.

Consider a patent licensing negotiation. The licensee knows the maximum they would pay; the licensor knows the minimum they would accept. In an ideal negotiation, they would agree on a price within this zone of possible agreement (ZOPA) without either party learning the other's boundary. In practice, every offer reveals information. A first offer of $100K signals willingness to pay at least $100K. A counteroffer of $500K signals willingness to accept at most $500K. Through iterated rounds, each party's private valuation is progressively disclosed and exploited.

Myerson and Satterthwaite [1] proved that no incentive-compatible mechanism can achieve ex-post efficiency in bilateral trade with private valuations. We do not claim to circumvent this impossibility. The strategic tension in parameter-setting remains: a principal who signs a maximum price of $80K when their true maxi-

mum is $100K is shading their parameters, just as a negotiator might shade their initial offer. The impossibility applies at the parameter-setting stage with the same force as at the negotiation stage.

What the protocol does achieve is a reduction in information leakage during negotiation itself. The counterparty cannot observe the negotiation dynamics — how quickly the agent conceded, how many rounds it took, what intermediate offers were made. This intermediate information, which experienced negotiators exploit systematically, is sealed and destroyed.

### 1.1 Related Work

Hardware enclaves for confidential computation have been explored in blockchain contexts, including Oasis Network and Secret Network, which use TEEs to enable private smart contract execution. These systems protect on-chain computation but do not address multi-round bilateral negotiation between autonomous agents.

Multi-party computation (MPC) [2] enables joint computation over private inputs without revealing them. However, MPC requires the computation to be expressed as a circuit, which precludes the open-ended natural-language negotiation central to bilateral trade.

Yao's garbled circuits [3] and subsequent work on privacy-preserving negotiation [4] address two-party computation with private inputs, but are limited to predefined protocols rather than free-form agent interaction.

Fully homomorphic encryption [7] would in principle allow computation over encrypted negotiation state, but current implementations impose overhead several orders of magnitude beyond what is feasible for LLM inference.

Dark pools [5][6] solve information leakage for fungible token swaps. Bilateral trade involves non-fungible, complex assets that require multi-dimensional negotiation, not simple price matching.

### 1.2 Scope and Limitations of This Claim

The protocol addresses **Layer 2** of the information double-use problem: leakage during negotiation. It does not fully solve Layer 1 (discovery leakage — expressing interest reveals demand) or the inference possible from outcomes (settlement prices and no-deal results carry information). These residual leakage channels are discussed in Section 6.

## 2. The Information Double-Use Problem

### 2.1 Definition

In bilateral trade, private information serves two conflicting purposes:

1. **Reaching agreement**: Both parties must communicate preferences to find mutually acceptable terms.

2. **Maximizing individual surplus**: Each party wants to conceal preferences to prevent the counterparty from extracting value.

The fundamental tension is that the information required for (1) is the same information exploited in (2). We call this **information double-use**: private information that is used once to facilitate a deal is reused to worsen its terms.

### 2.2 Three Layers

The problem manifests at three distinct points in the trade lifecycle:

**Layer 1: Discovery leakage.** Expressing interest in an asset reveals demand. A buyer who contacts a patent holder has revealed that the patent has strategic value to them.

**Layer 2: Negotiation extraction.** Each offer and counteroffer is a signal. Anchoring effects, response timing, concession patterns, and even the choice to continue negotiating all leak information about private valuations.

**Layer 3: Post-settlement persistence.** After a deal closes, the intermediary retains complete knowledge of both parties' reservation prices and negotiation behavior.

## 3. Protocol Design

### 3.1 Trade Lifecycle

A trade progresses through six states: Listed → Matched → Negotiating → Agreed → Settled, with cancellation possible from any pre-agreement state.

**Listed.** A seller publishes a hashed asset description and a maximum deal value. They post a Discovery bond.

**Matched.** A buyer expresses interest and posts a matching bond.

**Negotiating.** Both parties escalate their bonds. Each party signs a parameter set defining acceptable terms for their AI agent. The signed parameters are loaded into a hardware-isolated enclave. The agents negotiate via a structured message protocol. Neither agent can communicate outside the enclave.

**Agreed.** Both parties escalate to the highest bond tier. The agreement hash and enclave attestation are recorded on the settlement layer.

**Settled.** The deal value transfers from buyer to seller, net of the protocol fee. Bonds are returned.

**Cancelled.** Before negotiation, bonds are returned without penalty. During or after negotiation, the cancelling party's bond is slashed: half to the counterparty, half to the insurance pool.

### 3.2 Information Sealing via Hardware Enclaves

Agents run inside Trusted Execution Environments (Intel TDX, AMD SEV-SNP). The enclave provides memory isolation and remote attestation. Upon negotiation completion, the enclave executes a memory-zeroing procedure. Standardized attestation of destruction is an area of active development; implementations must demonstrate this property through platform-specific mechanisms.

### 3.3 Trust Assumption

The enclave requires trusting the hardware vendor. This differs from mathematical hardness assumptions: ECDSA's security rests on a mathematical conjecture; enclave security rests on hardware manufacturing and firmware, subject to supply chain risks and implementation bugs. We accept this tradeoff because the alternative — no confidentiality guarantee at all — is strictly worse. An insurance pool bounds the economic consequence of failure.

### 3.4 Why Crypto as Payment Rail

Agent-to-agent settlement requires a programmable, permissionless payment rail. Traditional payment systems (bank transfers, credit cards) require human approval, take hours to days, and impose counterparty risk. Stablecoin on an L2 chain provides sub-second finality, full programmability, and negligible transaction cost — the only payment rail compatible with autonomous agent operation.

Smart contracts provide atomic settlement (deal value and bond release in one transaction), programmatic bond escalation, and an immutable audit trail — capabilities that would otherwise require a trusted third-party escrow agent.

### 3.5 Agent Design

Each agent is a language model running inside the enclave, bound by the principal's signed parameters as an irrevocable mandate. Agents exchange structured messages in alternating turns until agreement is reached, a parameter boundary is hit, or a round limit is exceeded. This constrains the agent's action space but does not eliminate non-determinism — negotiation quality depends on the underlying model's capabilities. The agent runtime is not yet implemented; the current release covers the settlement layer only.

## 4. Economic Model

### 4.1 Fee Structure

A 0.3% fee is collected at settlement. The buyer transfers the full deal value; the seller receives the deal value minus the fee. The fee is routed to the protocol's insurance pool and operational treasury.

### 4.2 Bonds

Bonds prevent spam and provide economic recourse. The bond amount at each stage is:

$$B(d, stage) = clamp(d \cdot r_{stage}, B_{min}, B_{max})$$

| Stage | Rate | Minimum | Maximum |
|---|---|---|---|
| Discovery | 1% | $1 | $1,000 |
| Negotiation | 3% | $5 | $5,000 |
| Execution | 10% | $10 | $50,000 |

Escalation is additive. Bonds are returned on successful settlement. On dispute, the faulty party's bond is split equally between the counterparty and the insurance pool.

Bonds are locked capital, not spent capital. The non-recoverable cost of a settled trade is the 0.3% fee only.

### 4.3 Insurance Pool

The insurance pool is funded by protocol fees and 50% of slashed bonds. It provides economic recourse for enclave breach claims. Pool balance and claim history are publicly verifiable on-chain.

## 5. Security Analysis

### 5.1 Enclave Breach

The protocol's confidentiality depends on hardware enclave integrity. Breach probability is non-zero. The insurance pool bounds economic consequence but does not guarantee recovery. Historical TEE vulnerabilities (Foreshadow, ÆPIC Leak, CacheOut) demonstrate that enclave security is not absolute.

### 5.2 Economic Attacks

**Bond griefing.** Cost: attacker's own bond (3%+ of deal value). Benefit: counterparty's wasted time. Minimum bond ensures non-trivial cost.

**Sybil trading.** The protocol offers no reward for self-trading. The 0.3% fee is a pure cost with no offsetting incentive.

### 5.3 Settlement Integrity

The settlement layer enforces a strict state machine. Each transition requires bond escalation, cryptographic signatures, and enclave attestation. No administrative overrides exist.

## 6. Limitations

**Hardware vendor trust.** Confidentiality depends on enclave vendor integrity. Unlike mathematical hardness assumptions, this involves supply chains and manufacturing processes.

**Discovery leakage.** The protocol does not seal the discovery phase. Expressing interest and posting bonds are observable actions that reveal demand.

**Outcome leakage.** Settlement prices partially reveal private valuations. No-deal outcomes provide bounds on the other party's reservation price.

**Bond side channel.** Bond amounts are proportional to deal value and publicly observable, revealing deal scale to third parties.

**On-chain attestation.** The current implementation stores attestation hashes but does not cryptographically verify them. Verification requires integration with platform-specific TEE attestation services.

**Protocol asymmetry.** The seller sets the maximum deal value; the buyer has no on-chain equivalent. The protocol is structurally seller-initiated.

**Agent non-determinism.** Language models are stochastic. Identical parameters may produce different negotiation outcomes.

**Legal enforceability.** The protocol produces a cryptographic agreement hash, but legal contracts — not on-chain state — are the enforceable instrument in most jurisdictions.

## 7. Conclusion

We have presented a protocol that reduces the leakage of private information during bilateral trade negotiation. The mechanism confines negotiation to hardware-isolated AI agents and, conditional on enclave integrity, removes negotiation dynamics — offers, counteroffers, timing, concession patterns — from observability by either party after the trade concludes.

The protocol does not solve the information double-use problem completely. Discovery leakage, outcome inference, and parameter-setting incentives remain. What it seals is the richest source of exploitable information: the negotiation process itself.

A reference implementation is available as open-source software.

## References

[1] R. Myerson and M. Satterthwaite, "Efficient Mechanisms for Bilateral Trading," *Journal of Economic Theory*, vol. 29, pp. 265-281, 1983.

[2] O. Goldreich, S. Micali, and A. Wigderson, "How to Play Any Mental Game," *STOC*, 1987.

[3] A. Yao, "Protocols for Secure Computations," *FOCS*, 1982.

[4] R. Moraffah and B. Sankar, "Privacy-Preserving Multi-Round Negotiations," *IEEE Transactions on Information Forensics and Security*, 2021.

[5] Renegade, "On-Chain Dark Pool," 2023.

[6] Penumbra Labs, "A Private DEX and Shielded Pool," 2023.

[7] C. Gentry, "Fully Homomorphic Encryption Using Ideal Lattices," *STOC*, 2009.