Proof-Carrying Trust

Zero-Knowledge Constraints for EQBSL

O. C. Hirst
Independent Researcher

December 2025

Abstract

Trust systems on the internet tend to behave like oracles: they emit scores, decline to explain themselves, and insist that users take it on faith that the internals are sane. The EQBSL framework replaces this with an evidence-flow calculus for trust on dynamic (hyper)graphs, but it still leaves a gap between specification and implementation. This paper closes that gap with a proof-carrying construction: we define circuits and zero-knowledge constraints that force each trust update to respect the EQBSL operator, without revealing underlying evidence. In short, if a system claims to be running the EQBSL calculus, it can now be compelled to *prove* it.

Keywords: EQBSL; subjective logic; evidence flow; zero-knowledge proofs; SNARKs; reputation systems.

1 Introduction

A trust score that cannot be interrogated is not a metric, it is a superstition. The EQBSL paper [4] attacks this by putting trust on an explicit footing: evidence tensors on edges, an operator F that rewrites the state step-by-step, and a mapping from evidence to Subjective Logic opinions grounded in EBSL [2, 3, 1]. The result is a calculus that makes its assumptions legible.

What EQBSL does not do on its own is prevent an implementation from quietly cheating. Nothing in the algebra stops a developer from publishing $\mathcal{E}_{t+1}$ that is only loosely related to $F(\mathcal{E}_t, \Delta_t)$, or embeddings $\boldsymbol{u}_i(t)$ that have a more mystical than mathematical relationship to the stated operator Γ.

This paper proposes a remedy: a proof-carrying instantiation in which every EQBSL state transition and, optionally, every published embedding, is accompanied by a zero-knowledge proof that the right algebra was followed. Evidence and contextual details can remain private; the obedience to F and Γ cannot.

2 Preliminaries: EQBSL state model (recap)

We adopt the notation and definitions of EQBSL as given in [4], summarised here only as much as is needed to define constraints.

2.1 Evidence and opinions

For a binary proposition, EBSL interprets:

- r as the amount of positive evidence,
- s as the amount of negative evidence,

- $K > 0$ as a fixed normalisation / prior-weight constant.

The mapping from evidence to a Subjective Logic opinion (b, d, u) is

$$b = \frac{r}{r+s+K}, \qquad d = \frac{s}{r+s+K}, \qquad u = \frac{K}{r+s+K}, \tag{1}$$

with $b + d + u = 1$. The constant K is a protocol parameter, shared across this paper and [4]; algebraically it behaves the same here as there.

2.2 Evidence tensors and aggregation

For each ordered pair of agents (i, j) and time t, EQBSL maintains an evidence vector

$$\boldsymbol{e}_{ij}(t) \in \mathbb{R}^m. \tag{2}$$

Scalar evidence parameters are recovered via nonnegative aggregation maps

$$r_{ij}(t) = \phi_+\big(\boldsymbol{e}_{ij}(t)\big), \qquad s_{ij}(t) = \phi_-\big(\boldsymbol{e}_{ij}(t)\big), \tag{3}$$

with $\phi_+, \phi_- : \mathbb{R}^m \longrightarrow \mathbb{R}_{\geq 0}$ fixed as part of the system design (often linear). The induced edge opinion is

$$\omega_{ij}(t) = \big(b_{ij}(t), d_{ij}(t), u_{ij}(t), a_{ij}\big), \tag{4}$$

with (b_{ij}, d_{ij}, u_{ij}) given by (1) applied to (r_{ij}, s_{ij}).

2.3 Global state and operator

Let the time-indexed (hyper)graph be $\mathcal{G}_t = (V, E_t)$, and let the evidence field be

$$\mathcal{E}_t = \{\boldsymbol{e}_{ij}(t)\}_{(i,j)\in V\times V}. \tag{5}$$

Let Δ_t denote the collection of new events between t and $t + 1$. EQBSL defines an update operator

$$F : \ (\mathcal{E}_t, \Delta_t) \longmapsto \mathcal{E}_{t+1}, \tag{6}$$

and a state

$$S_t := \big(\mathcal{G}_t, \mathcal{E}_t\big), \tag{7}$$

so that

$$S_{t+1} = \big(\mathcal{G}_{t+1}, \mathcal{E}_{t+1}\big), \qquad \mathcal{E}_{t+1} = F(\mathcal{E}_t, \Delta_t). \tag{8}$$

Hyperedge evidence and its allocation to pairwise tensors are defined in [4] and imported here unchanged.

Finally, the embedding operator

$$\boldsymbol{u}_i(t) = \Gamma\big(i, \mathcal{E}_t, \mathcal{G}_t\big) \in \mathbb{R}^{d_u} \tag{9}$$

produces node-level trust embeddings. In this paper we treat F, $\phi_\pm$, Γ, and K as fixed public parameters: they determine the constraints we are going to prove.

3 Commitments and statement of soundness

The core idea is conceptually simple: commit to state, update according to F, and prove that this update occurred as advertised.

3.1 State commitments

At time t the EQBSL state is $S_t = (\mathcal{G}_t, \mathcal{E}_t)$. A verifier does not need to see S_t; they only need a binding handle for it. Let

$$C_t = \mathsf{Com}(S_t; r_t) \tag{10}$$

be a commitment under some binding, hiding vector commitment scheme Com with randomness r_t. The scheme itself is not the point; the fact that C_t cryptographically pins down S_t is.

3.2 Transition statement

Given the EQBSL operator F, the honest update from t to $t+1$ is

$$S_{t+1} = \big(\mathcal{G}_{t+1}, F(\mathcal{E}_t, \Delta_t)\big). \tag{11}$$

The public statement we want to prove is:

Given public inputs (C_t, C_{t+1}, Δ_t), there exist states S_t, S_{t+1} and randomness (r_t, r_{t+1}) such that

$$\begin{aligned} C_t &= \mathsf{Com}(S_t; r_t), \\ C_{t+1} &= \mathsf{Com}(S_{t+1}; r_{t+1}), \\ S_{t+1} &= \big(\mathcal{G}_{t+1}, F(\mathcal{E}_t, \Delta_t)\big). \end{aligned}$$

In other words, the prover claims that the two commitments encode consecutive EQBSL states linked by the same operator F that appears in the algebra.

4 Circuits for F and Ψ

To make the statement above machine-checkable, we compile the EQBSL update into arithmetic circuits or equivalent constraint systems.

4.1 Circuit C_F for the evidence operator

Define an arithmetic circuit

$$C_F : (S_t, \Delta_t) \longmapsto S_{t+1} \tag{12}$$

whose gates implement exactly the EQBSL update rules:

- per-edge and per-hyperedge evidence updates,
- temporal decay (if any) on components of $\boldsymbol{e}_{ij}(t)$,
- aggregation of hyperedge evidence into pairwise tensors via the same α_{ijh} and Π_{ij} as in [4],
- any domain-specific clamping or normalisation.

The constraint system for C_F contains, in arithmetised form, the same algebra that defines F; there is no room for a second, "convenient" operator.

4.2 Circuit C_Ψ for evidence-to-opinion mapping

While the mapping Ψ from $\boldsymbol{e}_{ij}(t)$ to $\omega_{ij}(t)$ is not strictly necessary for state soundness, it often matters for interface soundness: if a system publishes opinions or uses them for access control, it should not lie about them either.

We therefore define a per-edge circuit

$$C_\Psi : \ \boldsymbol{e}_{ij}(t) \longmapsto \omega_{ij}(t) \tag{13}$$

whose constraints implement:

$$r_{ij}(t) = \phi_+\big(\boldsymbol{e}_{ij}(t)\big), \qquad s_{ij}(t) = \phi_-\big(\boldsymbol{e}_{ij}(t)\big), \tag{14}$$

$$b_{ij}(t) = \frac{r_{ij}(t)}{r_{ij}(t)+s_{ij}(t)+K}, \qquad d_{ij}(t) = \frac{s_{ij}(t)}{r_{ij}(t)+s_{ij}(t)+K}, \tag{15}$$

$$u_{ij}(t) = \frac{K}{r_{ij}(t)+s_{ij}(t)+K}, \qquad b_{ij}(t)+d_{ij}(t)+u_{ij}(t) = 1. \tag{16}$$

In practice, denominators are handled via fixed-point arithmetic and multiplicative constraints rather than literal division, but the algebra is identical. The same K and the same $\phi_\pm$ appear here as in (1)–(3).

5 Circuits for embeddings Γ

When node-level embeddings are published or used as features, we may wish to prove that they are honest functions of the committed EQBSL state.

5.1 Circuit C_Γ

Let $U_t = \{\boldsymbol{u}_i(t)\}_{i\in V}$ be the embedding set at time t. Define a circuit

$$C_\Gamma : \ S_t \longmapsto U_t \tag{17}$$

whose constraints implement $\boldsymbol{u}_i(t) = \Gamma(i, \mathcal{E}_t, \mathcal{G}_t)$ exactly. The details of Γ are domain-specific: it may be a graph neural network, a hand-designed feature map, or a simpler summary. What matters is that the same Γ that appears in specifications is the one whose constraints end up in C_Γ.

The public statement then becomes:

> "Given C_t and a candidate embedding set U_t, there exists S_t, r_t such that $C_t = \mathsf{Com}(S_t; r_t)$ and $U_t = C_\Gamma(S_t)$."

If the system wants to enjoy the authority of "EQBSL embeddings", it must accept the cost of proving it.

6 Zero-knowledge layer

All of the above can, in principle, be done with transparent commitments and non-zero-knowledge proofs. In practice, it is often socially and legally necessary to keep raw evidence private, while still providing verifiable guarantees about trust computation.

6.1 Prover and verifier views

The *prover* knows:

- the full states S_t, S_{t+1},
- the randomness r_t, r_{t+1} for commitments,
- the complete event set Δ_t (or at least its private portions).

The *verifier* sees:

- public inputs $(C_t, C_{t+1}, \Delta_t^{\text{pub}})$,
- optional published embeddings U_t, U_{t+1} and/or opinions for selected edges,
- a zero-knowledge proof π_t attesting that the constraints of C_F (and optionally C_Ψ, C_Γ) are satisfied.

A modern proof system (SNARK, STARK, etc.) provides succinct π_t and efficient verification. The choice of scheme is a question of engineering, not of algebra.

6.2 Revelation policies

One of the advantages of working in evidence space is granularity. A system can commit to the entire $\mathcal{E}_t$ but only reveal:

- aggregate opinions for a subset of edges,
- embeddings for a subset of nodes,
- coarse-grained statistics for compliance or audit.

The proof does not care; it only cares that the revealed pieces are consistent with some hidden whole that obeys the EQBSL operator. The rest of the ledger stays under the floorboards.

7 Discussion

The point of this paper is not to sell yet another proof system. It is to show that once trust is written as evidence flow (EQBSL), the usual excuses for unverifiable behaviour look thin. If you can write down F and Γ, you can arithmetise them. If you can arithmetise them, you can prove you followed them.

There are, of course, costs: circuit size, prover time, and the usual headaches of efficient arithmetisation. But these are honest costs. They make explicit the computational price of epistemic hygiene in a domain that has grown accustomed to free-floating scores. In exchange, one gains a property that “trust oracles” by design never had: the obligation to show their work, or at least a zero-knowledge proof that the work exists.

Acknowledgements

The EQBSL framework that underpins this paper is due to O. C. Hirst [4]. Subjective Logic is credited to Audun Jøsang [1], and Evidence-Based Subjective Logic and evidence-flow discounting to Škorić and collaborators [2, 3]. Any remaining heresies in how these ideas are wired into zero-knowledge constraints are the author’s own.

References

[1] Audun Jøsang. *Subjective Logic: A Formalism for Reasoning Under Uncertainty*. Springer, 2016. DOI: 10.1007/978-3-319-42337-1.

[2] Boris Škorić, Sebastiaan J. A. de Hoogh, and Nicola Zannone. Flow-based reputation with uncertainty: evidence-based subjective logic. *International Journal of Information Security*, 15:381–402, 2016. (Published online 20 Aug 2015.) DOI: 10.1007/s10207-015-0298-5.

[3] Boris Škorić. Flow-based reputation with uncertainty: Evidence-Based Subjective Logic. arXiv:1402.3319, 2014 (revised 2015).

[4] Oliver C. Hirst. EQBSL: Against Trust Scores – Evidence, Uncertainty, and Trust Flow in Dynamic Networks. Manuscript, 2025.