Fraud Game Analysis for Hellas Protocol: Incentives, Detection, and Protocol Design

CryptoEconLab

January 2026

Abstract

We study the incentive properties of the Hellas fraud game for off-chain computation. Providers post stake that can be slashed if a challenger proves fraud. The objective is incentive compatibility: parameter regions where honest execution is an equilibrium outcome for rational, risk-neutral agents. The analysis separates (i) *enforcement* in the dispute subgame, conditional on fraud being identified, and (ii) *detection* when clients are informationally constrained and must decide whether to audit. Part I derives the conditions under which disputing is privately profitable for a rational challenger and yields a minimum viable provider stake (or other at-risk funds) that eliminates an impunity region. Part II models detection as an inspection game and characterizes the mixed-strategy equilibrium, including boundary cases and the role of job-specific loss from incorrect computation. We conclude with protocol design recommendations for stake floors, reward routing, permissionless challenging, randomized audits, and timeouts.

Contents

Introduction

The Hellas fraud game secures off-chain computation through economic incentives. A provider posts collateral that is slashed if a challenger proves that the provider returned an incorrect result. The primary question is incentive compatibility: for which parameters does a rational provider prefer honest execution? Addressing this question requires separating two components.

1. **Enforcement (dispute subgame)**: conditional on fraud being identified, is it privately optimal for an eligible challenger to initiate a dispute and, if so, does the protocol execute slashing with sufficiently high reliability?

2. **Detection (audit stage)**: when clients cannot immediately verify correctness, how frequently is fraud identified in equilibrium, and how does that frequency depend on stake and verification costs?

Part I analyzes enforcement by treating detection as given and by modeling imperfect enforcement reliability. Part II analyzes detection by treating enforcement parameters as fixed and by modeling client auditing and provider cheating as an inspection game.

A key premise is the existence of a safe fallback. There exists a method to compute the correct output and, when needed, construct a fraud proof accepted by the protocol. This premise anchors both incentives and feasibility, since the dispute mechanism must be executable within protocol time constraints.

Model Setup

Actors

- **Client** requests computation, escrows payment and any required bonds.
- **Provider** performs computation, posts stake S_P for the duration of the channel, and is paid P_{set} if the channel settles without a successful fraud proof.
- **Validators** (or an on-chain verifier) check the fraud proof and execute the associated state transition, including slashing.
- **Challenger** is any entity that is permitted to initiate a dispute and present a fraud proof. In the baseline, the challenger is the client. In protocol variants, challenging can be permissionless.

Timeline and settlement rule

Time is discrete. A channel instance proceeds through the following phases.

1. Channel open: the provider posts stake S_P; the client escrows the payment P_{set} and any required bonds.
2. Execution: the provider returns an output y.
3. Challenge window of length T: any eligible challenger may initiate a dispute.
4. Dispute resolution: the challenger posts a challenge bond B_C, computes a reference output via safe fallback, generates a fraud proof, and submits it. Validators verify the proof and execute slashing if the proof is accepted.
5. Finalization: if no successful fraud proof is submitted before the deadline, the channel settles and the provider is paid P_{set}. If a successful fraud proof is submitted, the provider is slashed and payment routing follows the protocol rule described below.

Assumptions

Assumption 1 (Preferences and timing)**.** *Agents are risk-neutral and myopic at the level of a single channel instance. Repeated-game and reputation effects are discussed separately as extensions.*

Assumption 2 (Safe fallback and feasibility)**.** *There exists a procedure to compute the correct output at cost C_{safe} and to generate a valid fraud proof at cost c_{proof}. The total wall-clock time required for safe fallback and proof generation is at most the dispute deadline, up to a buffer for transaction inclusion and finality.*

Assumption 3 (Enforcement reliability)**.** *Conditional on a valid fraud proof being submitted before the deadline, the protocol accepts the proof and executes slashing with probability $p_w \in (0, 1]$. This parameter captures residual risks such as censorship, liveness failures, deadline misses, or verifier failure. Baseline analysis sets $p_w = 1$.*

Key Assumption: Safe Fallback Exists

Safe fallback is a method to compute the correct result and, if needed, produce a fraud proof that is accepted by the protocol. In the simplest case the client recomputes the job on their own infrastructure and uses the resulting trace as input to proof generation. Let C_{safe} denote the total cost of this fallback computation, including engineering overhead and any fees required to access state or data.

Note

Trusted fallback providers can supply verification as a service. If the service market is competitive and the verification task is well specified, then the expected cost C_{safe} decreases, which lowers the minimum stake required for credible enforcement and reduces expected auditing costs (Section 3).

Parameters

Table 1 lists the core parameters. All variables are channel-specific unless noted otherwise.

Dispute mechanics and routing

A dispute is initiated after an eligible challenger believes the output is incorrect and can produce a fraud proof. The challenger first computes a reference output via safe fallback, incurring cost C_{safe}. If fraud is found, the challenger generates a fraud proof at cost c_{proof} and submits it on-chain along with any required transactions at cost c_{tx}, posting bond B_C. The proof is accepted and enforced with probability p_w.

If enforcement succeeds, the provider stake S_P is slashed. A fraction βS_P is paid to the challenger. In addition, a fraction λP_{set} of the escrowed payment is routed to the challenger, with $\lambda = 1$ corresponding to full refund to the challenger and $\lambda = 0$ corresponding to no payment routing to the challenger. The bond B_C is returned when enforcement succeeds and is forfeited when enforcement fails.

For clarity, we treat challenge initiation as occurring only when the challenger has computed the safe fallback output. The analysis therefore separates (i) the decision to compute the safe fallback and (ii) conditional on finding fraud, the decision to submit a proof. This decomposition matters for the belief-based model in Part II.

Symbol	Description	Set By
S_P	Provider stake locked for the channel duration	Provider / protocol floor
P_{set}	Settlement payment to provider if no successful fraud proof	Bilateral
c_H	Provider cost of honest execution	Exogenous
c_F	Provider cost under cheating	Exogenous
C_{safe}	Cost of safe fallback computation	Market / job type
c_{proof}	Cost of fraud proof generation (conditional on fraud being found)	Exogenous
c_{tx}	On-chain transaction and operational overhead for disputing	Network conditions
β	Fraction of slashed stake paid to challenger on success	Protocol
λ	Fraction of P_{set} routed to challenger on success	Protocol
B_C	Challenge bond posted with the dispute	Protocol
p_w	Probability a timely valid proof is accepted and enforced	Protocol environment
L	Client loss from accepting an incorrect result	Job dependent
r	Opportunity cost rate for locked capital	Market
τ	Capital lock duration for stake and escrow	Protocol / job duration
T	Challenge window length	Protocol

Table 1: Model parameters. The routing parameter λ is defined so that, conditional on a successful fraud proof, the challenger receives λP_{set} and the provider does not receive that portion of payment.

Part I
Fraud Enforcement

Part I studies the dispute subgame conditional on fraud being discovered by some challenger. The key question is whether disputing is privately optimal for that challenger. If disputing is privately optimal with high probability, then cheating induces expected slashing and the provider incentive constraint can be satisfied.

1 Incentive Conditions

1.1 Provider incentives

Let $p_d \in (0, 1]$ denote the probability that a cheating provider is detected and successfully punished. This probability is conditional on the provider choosing to cheat. In Part I, p_d is determined by (i) whether a challenger initiates a dispute after discovering fraud and (ii) enforcement reliability p_w.

Definition 1 (Provider payoffs)**.**

$$U_P(H) = P_{set} - c_H, \tag{1}$$

$$U_P(C, \textit{detected}) = -S_P - c_F, \tag{2}$$

$$U_P(C, \textit{undetected}) = P_{set} - c_F. \tag{3}$$

In $U_P(C, \textit{detected})$ the provider does not receive P_{set}. This reflects the settlement rule that payment is released only when the channel settles without a successful fraud proof.

Proposition 1 (Provider incentive compatibility)**.** *A provider prefers honest execution over cheating if and only if*

$$\boxed{p_d \geq \theta := \frac{c_H - c_F}{P_{set} + S_P}} \tag{4}$$

When $c_F \approx 0$, this becomes $p_d \geq \frac{c_H}{P_{set}+S_P}$.

Proof. Cheating yields expected payoff

$$\mathbb{E}[U_P(C)] = (1-p_d)(P_{set} - c_F) + p_d(-S_P - c_F) = P_{set} - c_F - p_d(P_{set} + S_P).$$

Honesty yields $U_P(H) = P_{set} - c_H$. Honest execution is preferred if and only if

$$P_{set} - c_H \geq P_{set} - c_F - p_d(P_{set} + S_P).$$

Rearranging gives $p_d(P_{set} + S_P) \geq c_H - c_F$, which is equivalent to the stated condition. □

The parameter θ is the minimum effective detection and enforcement probability required to deter cheating. Increasing S_P lowers θ by increasing the expected penalty. Increasing P_{set} lowers θ because the provider forfeits a larger payment if detected.

1.2 Challenger incentives to dispute

We now derive conditions under which a challenger who has identified fraud finds it privately optimal to submit a dispute. The challenger could be the client or a permissionless watcher. The analysis is identical once costs and rewards are specified.

Let the challenger incur total dispute cost

$$C_{disp} := C_{safe} + c_{proof} + c_{tx}. \tag{5}$$

Conditional on fraud having occurred and having been identified by the challenger, the challenger receives reward

$$R := \beta S_P + \lambda P_{set} \tag{6}$$

if enforcement succeeds. Enforcement succeeds with probability p_w. If enforcement fails, the challenger forfeits bond B_C. We assume the challenger posts the bond only when submitting the proof, so the bond is part of the dispute decision conditional on identifying fraud.

Proposition 2 (Dispute condition and minimum viable stake)**.** *Conditional on identifying fraud, disputing is strictly optimal for the challenger if and only if*

$$\boxed{p_w R - C_{disp} - (1-p_w)B_C > 0.} \tag{7}$$

If $\beta > 0$, a sufficient stake condition that guarantees disputing is optimal is

$$\boxed{S_P \geq S_P^{min} := \max\left\{0,\ \frac{C_{disp} + (1-p_w)B_C - p_w\lambda P_{set}}{p_w\beta}\right\}.} \tag{8}$$

Proof. Conditional on identifying fraud, the challenger compares not disputing, which yields payoff 0, to disputing, which yields expected payoff

$$\mathbb{E}[U_C(\text{dispute})] = p_w R - C_{disp} - (1-p_w)B_C.$$

Disputing is strictly optimal if and only if this expression is positive. Solving $p_w(\beta S_P + \lambda P_{set}) - C_{disp} - (1-p_w)B_C > 0$ for S_P yields the stated condition, with truncation at zero since stake is nonnegative. □

Key implication

If $S_P < S_P^{min}$, then even after fraud is identified, disputing is not privately optimal for the challenger. In that region, the effective punishment probability p_d collapses because disputes are not initiated.

The expression for S_P^{min} highlights three protocol levers. Increasing β increases the challenger reward per unit stake. Increasing λ routes more of the at-risk payment to the challenger on success. Increasing p_w makes enforcement more reliable and reduces the stake needed for credible enforcement.

2 Enforcement viability and participation

We combine the challenger condition with the provider incentive constraint.

Theorem 1 (Fraud game viability under enforceable disputes)**.** *Suppose that fraud is identified by an eligible challenger and that the provider stake satisfies $S_P \geq S_P^{min}$ from Proposition 2. Then disputing is initiated and the provider is punished with probability at least p_w, so $p_d = p_w$ in the dispute subgame. Under these conditions, honest execution is incentive compatible for the provider if*

$$\boxed{p_w \geq \theta = \frac{c_H - c_F}{P_{set} + S_P}.} \tag{9}$$

Proof. If $S_P \geq S_P^{min}$, then disputing is strictly optimal for the challenger conditional on identifying fraud, hence a dispute is initiated. Enforcement succeeds with probability p_w by assumption, so the probability that cheating is detected and punished is $p_d = p_w$ in this subgame. Substituting $p_d = p_w$ into Proposition 1 yields the stated condition. □

2.1 Provider participation

Providers must also be willing to supply service. Stake is returned when the provider is honest, so the main stake cost is opportunity cost. Let r be the per-unit-time opportunity cost of capital and let τ be the lock duration. A simple participation constraint is

$$\boxed{P_{set} \geq c_H + rS_P\tau + \kappa,} \tag{10}$$

where $\kappa \geq 0$ captures operational overhead and risk premia. This constraint matters because raising S_P to satisfy enforcement increases capital cost and therefore increases competitive pricing.

2.2 Feasibility and timeouts

Let t_{safe} and t_{proof} denote worst-case wall-clock times for safe fallback and proof generation for the job class. The protocol must set the challenge window T so that disputes are feasible:

$$\boxed{T \geq t_{safe} + t_{proof} + t_{tx} + t_{finality},} \tag{11}$$

where t_{tx} is a buffer for transaction inclusion and $t_{finality}$ is a buffer for settlement finality. If this feasibility condition fails, then p_w is effectively reduced and enforcement weakens even if incentives are otherwise aligned.

3 Trusted on-chain fallback providers

The safe fallback cost C_{safe} is a central driver of both enforcement and detection. When verification is self-executed, C_{safe} includes raw compute, engineering overhead, and any channel overhead required to bind the computation to a proof system. When verification can be outsourced to trusted fallback providers, C_{safe} can decrease because the client avoids infrastructure overhead and obtains a predictable service.

Lower C_{safe} reduces the minimum viable stake S_P^{min} in Proposition 2, lowers the equilibrium cheating rate in Part II, and lowers expected verification expenditure. In competitive equilibrium, providers still price in capital costs, so reductions in required stake reduce equilibrium prices via the participation constraint.

4 Client participation and welfare accounting

A client joins if expected total cost is below the outside option. Let C_{self} be the all-in cost of self-execution, including capital and operational overhead. Let C_{chan} be the client cost of using the channel, including P_{set}, fees, and capital costs on any escrows.

In settings where clients sometimes audit, expected cost includes expected verification expenditure. If the client audits with probability v, then the expected verification cost per job is approximately vC_{safe} plus additional conditional costs when fraud is found. Under the one-shot inspection model in Part II, the equilibrium audit rate is endogenous. Under protocol-imposed randomized audits, v includes the imposed audit probability.

A sufficient participation condition is $C_{chan} + vC_{safe} < C_{self}$, together with a bound on expected loss from undetected fraud. If an incorrect result causes loss L and fraud is not always detected, then the expected additional loss term is $q(1 - v)L$ in the inspection model, which further tightens participation when L is large.

Summary of Part I

Part I provides two conditions that are jointly needed for deterrence via enforceable disputes: (i) the challenger must be willing to dispute conditional on identifying fraud, and (ii) the resulting punishment probability must satisfy the provider incentive constraint. A protocol can eliminate the impunity region by enforcing a minimum stake floor derived from Proposition 2.

Part II
Fraud Detection

Part II studies detection. Even if disputes are enforceable conditional on fraud being identified, fraud may remain profitable if clients do not audit and no other challengers exist. We analyze two complementary models. First, a belief-based model where the client audits only when posterior belief exceeds a threshold. Second, a one-shot inspection game where the provider and client randomize. Both models yield explicit dependence of fraud on stake, verification costs, and job-specific loss L.

5 Belief-based auditing

5.1 Signals and posterior belief

Let $\mu \in [0,1]$ denote the client's posterior belief that the provider cheated and returned an incorrect output.

A rigorous model specifies a signal s observed by the client, with conditional densities $f_H(s)$ and $f_C(s)$ under honest and cheating behavior. Given prior $\mu_0 \in (0,1)$, Bayes' rule yields

$$\mu(s) = \mathbb{P}(C \mid s) = \frac{\mu_0 f_C(s)}{\mu_0 f_C(s) + (1-\mu_0) f_H(s)}. \tag{12}$$

The signal can encode sanity checks, reputation, historical behavior, and any auxiliary verification heuristics.

5.2 Audit and dispute decisions

At the output stage, the client chooses either to accept the output or to audit via safe fallback. If the client accepts and the output is incorrect, the client incurs loss L. If the client audits, the client pays C_{safe}. If fraud is found, the client can submit a fraud proof, incurring additional cost $c_{proof} + c_{tx}$ and facing enforcement reliability p_w.

Define the net expected incremental payoff from auditing rather than accepting as a function of posterior belief μ and parameters. If the client accepts, expected loss is μL. If the client audits, the loss L is avoided and, conditional on fraud, the client can obtain expected dispute surplus

$$\Delta := p_w(\beta S_P + \lambda P_{set}) - (c_{proof} + c_{tx}) - (1-p_w)B_C. \tag{13}$$

Auditing yields expected net gain relative to accepting

$$G(\mu) = -C_{safe} + \mu(L + \Delta). \tag{14}$$

Proposition 3 (Belief threshold for auditing)**.** *Assume $L + \Delta > 0$. The client audits if and only if*

$$\boxed{\mu > \mu^* := \frac{C_{safe}}{L + \Delta}.} \tag{15}$$

If $L + \Delta \le 0$, then auditing is never optimal for any $\mu \in [0,1]$.

Proof. The client audits if and only if $G(\mu) > 0$. Solving $-C_{safe} + \mu(L+\Delta) > 0$ for μ yields the threshold. If $L + \Delta \le 0$, then $G(\mu) \le -C_{safe} < 0$ for all μ. □

This threshold makes explicit how stake and routing influence detection through incentives. Increasing S_P increases Δ via βS_P and therefore lowers μ^*. Increasing λ lowers μ^* when payment is refunded to the challenger on successful disputes. Increasing C_{safe} raises μ^* and reduces auditing. Larger job loss L lowers μ^* and increases auditing for high-stakes computations.

6 Inspection game with mixed equilibrium

We now model the one-shot interaction in which the provider chooses whether to cheat and the client chooses whether to audit, without observing the other's action. The client is blind in the sense that auditing is a costly action, while acceptance can expose the client to loss L if the output is incorrect.

6.1 Game specification

Definition 2 (Inspection game)**.** *The stage game has actions $\{H, C\}$ for the provider (honest, cheat) and $\{A, V\}$ for the client (accept, audit). The moves are simultaneous. If the client audits and fraud is found, the client submits a fraud proof and enforcement succeeds with probability p_w.*

Define $C_{audit} := C_{safe}$ and define the expected net dispute surplus conditional on fraud, Δ, as in Proposition 3.

	Client: Accept (A)	Client: Audit (V)
Provider: Honest (H)	$(P_{set} - c_H,\ -P_{set})$	$(P_{set} - c_H,\ -P_{set} - C_{safe})$
Provider: Cheat (C)	$(P_{set} - c_F,\ -P_{set} - L)$	$(-S_P - c_F,\ -P_{set} - C_{safe} + \Delta)$

Table 2: Payoffs (Provider, Client). Correct computation value is normalized to zero. Accepting a wrong output imposes loss L on the client. Auditing incurs C_{safe}. If fraud is found under auditing, the client can obtain expected dispute surplus Δ.

6.2 Mixed equilibrium

Let $q \in [0, 1]$ be the probability the provider cheats and let $v \in [0, 1]$ be the probability the client audits.

Proposition 4 (Provider indifference condition)**.** *If the provider mixes in equilibrium, then the client audit probability satisfies*

$$\boxed{v^* = \frac{c_H - c_F}{P_{set} + S_P}.} \tag{16}$$

Proof. The provider payoff from honest execution is $U_P(H) = P_{set} - c_H$, independent of v. The expected payoff from cheating is

$$\mathbb{E}[U_P(C)] = (1 - v)(P_{set} - c_F) + v(-S_P - c_F) = P_{set} - c_F - v(P_{set} + S_P).$$

Indifference requires $P_{set} - c_H = P_{set} - c_F - v(P_{set} + S_P)$, hence $v(P_{set} + S_P) = c_H - c_F$ and the stated expression follows. □

The expression for v^* coincides with the provider incentive threshold θ when detection probability is interpreted as audit frequency, in the sense that when $p_w = 1$ and an audit implies punishment, $p_d = v$.

Proposition 5 (Client indifference condition)**.** *Assume $L + \Delta > 0$. If the client mixes in equilibrium, then the provider cheating probability satisfies*

$$\boxed{q^* = \frac{C_{safe}}{L + \Delta}.} \tag{17}$$

Proof. The expected client payoff from accepting is

$$\mathbb{E}[U_C(A)] = (1-q)(-P_{set}) + q(-P_{set} - L) = -P_{set} - qL.$$

The expected client payoff from auditing is

$$\mathbb{E}[U_C(V)] = (1-q)(-P_{set} - C_{safe}) + q(-P_{set} - C_{safe} + \Delta) = -P_{set} - C_{safe} + q\Delta.$$

Indifference requires $-P_{set} - qL = -P_{set} - C_{safe} + q\Delta$, hence $q(L+\Delta) = C_{safe}$. Under $L+\Delta > 0$, this yields the stated expression. □

Theorem 2 (Mixed equilibrium and boundary cases)**.** *Assume* $0 < c_H - c_F < P_{set} + S_P$ *and* $0 < C_{safe} < L + \Delta$*. Then the inspection game admits a unique mixed-strategy Nash equilibrium with*

$$(v^*, q^*) = \left(\frac{c_H - c_F}{P_{set} + S_P},\ \frac{C_{safe}}{L + \Delta} \right).$$

If $C_{safe} \geq L + \Delta$*, then the client's best response is never to audit and cheating becomes a best response for the provider. If* $c_H - c_F \geq P_{set} + S_P$*, then the provider's best response is to cheat regardless of the client's audit probability.*

Proof. Under the stated strict inequalities, the indifference conditions in Propositions 4 and 5 yield $v^*, q^* \in (0,1)$. Standard inspection-game arguments imply uniqueness of the mixed equilibrium because each player's expected payoff difference between actions is affine in the opponent's mixing probability, hence each player has at most one mixing point. Boundary cases follow from the sign of the payoff differences when denominators are nonpositive or when the computed mixing probabilities lie outside $[0,1]$. □

6.3 Interpretation and comparative statics

The equilibrium cheating rate q^* decreases when S_P increases because Δ increases linearly in βS_P. The equilibrium audit rate v^* decreases when S_P increases because a larger penalty requires less auditing to satisfy provider indifference. Job loss L reduces q^* because auditing becomes privately beneficial at lower fraud rates when accepting a wrong output is costly.

The parameter p_w enters through Δ. Lower enforcement reliability reduces Δ and increases equilibrium cheating. A protocol that improves liveness and reduces censorship risk effectively lowers both S_P^{min} in Part I and q^* in Part II.

7 Reputation and repeated interaction (extension)

Reputation can be modeled as an on-chain observable state variable ρ that shifts priors or alters feasible actions through access control. For example, clients can set a prior $\mu_0 = f(\rho)$ with $f'(\rho) < 0$. For reputation to be meaningful, it must be derived from hard-to-forge history such as stake-weighted service volume, time in system, and dispute outcomes, rather than self-reported ratings. A repeated-game model can combine stake slashing and continuation value, which increases effective deterrence by raising the long-run cost of cheating. This extension can reduce required stake for a given target fraud rate when repeated interaction is strong and identity rotation is costly.

8 Protocol design recommendations

This section translates the preceding incentive constraints into implementable protocol rules. The recommendations target two objectives: eliminate an impunity region where disputes are not initiated, and bound equilibrium fraud for blind clients by inducing auditing from either clients, watchers, or the protocol itself.

8.1 Stake floors indexed by job class

A protocol can enforce a minimum stake floor as a function of job class, $S_{min}(\text{job})$, using conservative estimates of C_{disp} and relevant network conditions. Proposition 2 yields

$$S_{min}(\text{job}) = \max\left\{0,\ \frac{C_{disp}(\text{job}) + (1-p_w)B_C - p_w\lambda P_{set}(\text{job})}{p_w\beta}\right\}. \tag{18}$$

When $\lambda = 1$, routing the escrowed payment to the challenger reduces the stake requirement for credible enforcement. When $p_w < 1$, stake floors must increase, or the protocol must compensate challengers through larger rewards, lower costs, or higher p_w.

8.2 Permissionless challenging and watcher markets

If challenging is permissionless, then detection does not rely on client attention. A permissionless challenger will dispute whenever Proposition 2 holds for the challenger cost structure. This creates a market for auditing and can raise the effective detection probability p_d without increasing client burden. To support this design, the protocol must ensure data availability for recomputation and must specify deterministic dispute interfaces that third parties can use.

8.3 Randomized audits

Randomized protocol audits provide an exogenous detection baseline. If the protocol audits with probability α and private auditing occurs with probability v, then the effective punishment probability is $p_d = p_w(\alpha + (1-\alpha)v)$ under independence. A small α can materially improve deterrence for low-attention clients and can reduce the required stake to satisfy the provider incentive constraint. Funding can come from fees, a portion of slashed stake, or a dedicated audit budget.

8.4 Reducing verification cost

Lower C_{safe} reduces both the enforcement stake floor and the equilibrium fraud rate. Trusted fallback providers, standardized execution traces, and proof systems that support incremental or sampled verification can reduce C_{safe} and c_{proof}. The protocol can further reduce c_{tx} by minimizing the number of on-chain transactions required for a dispute and by using predictable gas patterns.

8.5 Timeout sizing

Timeouts should be sized by job class to satisfy $T \geq t_{safe} + t_{proof} + t_{tx} + t_{finality}$. If a single global T is used, it must be sized to the worst supported job class, which can impose unnecessary latency for short jobs. A job-indexed challenge window reduces latency while preserving feasibility.

8.6 Anti-griefing

Auditing and disputing can impose externalities on honest providers through delayed settlement and additional on-chain load. A protocol can reduce griefing by requiring a bond B_C that

is forfeited when enforcement fails, by routing part of dispute costs to the initiator, and by designing the dispute flow so that honest providers are not delayed by unverifiable challenges. When p_w is close to one, B_C can be small. When p_w is materially below one, B_C becomes economically relevant and should be reflected in stake floor calculations.

9 Conclusions

The analysis yields explicit constraints that can be used to parameterize Hellas. In the dispute subgame, credible enforcement requires that disputing be privately profitable for an eligible challenger, which yields a minimum viable stake condition that depends on enforcement reliability p_w, slashing reward share β, payment routing λ, dispute costs, and challenge bond B_C. Given credible enforcement, provider incentive compatibility requires that the effective punishment probability p_d exceed the threshold $\theta = \frac{c_H - c_F}{P_{set} + S_P}$.

In the blind client setting, detection is endogenous. A belief-based model yields an auditing threshold $\mu^* = \frac{C_{safe}}{L+\Delta}$. A one-shot inspection model yields a mixed equilibrium with audit rate $v^* = \frac{c_H - c_F}{P_{set} + S_P}$ and cheating rate $q^* = \frac{C_{safe}}{L+\Delta}$ under standard feasibility conditions. These expressions show how stake, verification costs, enforcement reliability, and job loss determine equilibrium outcomes.

Protocol design can improve performance by enforcing job-indexed stake floors, routing sufficient reward to challengers, enabling permissionless challenging, adding randomized audits, lowering verification costs, and sizing timeouts to guarantee dispute feasibility.