Split-Flow

Split-Flow is an orchestrator protocol that implements a dual-strategy to preserve confidentiality and ensure computation verifiability in Nesa's decentralized inference network. It constitutes an intelligent task-directed allocation system that leverages both hardware and cryptographic composites, whose employment is determined by an automated assessment mechanism based on input sensitivity, computational burden, model-oriented criteria, and consensus conformity parameters.

At its core, the Split-Flow Protocol operates by dissecting the workflow of any given computation into two principal streams: the confidentiality stream and the verifiability stream.

These streams are aligned with security controls that respond dynamically to variable requisites of the data model and inference objectives, yielding an efficient, secure, and verifiable computation cycle.

The Confidentiality Stream

The confidentiality stream utilizes a dual-modality operation to tackle the problem that traditional static encryption poses to computation. Upon initial assessment, the model size and computational complexity dictate whether a TEE-hosted enhanced node or cryptographic constructs take precedence, and the extent of composite interplay.

The Verifiability Stream

To buttress the protocol’s trust in computation outcomes, Split-Flow utilizes ZKPs to establish the correctness of computations without any requirement for data disclosure. It does so in a combinative nature, in concert with the confidentiality measures of both composites to ensure a system that maintains user privacy while concurrently producing attestable, accurate results.

Protocol Operation Workflow

Upon receiving a query request, the Split-Flow Protocol stratifies and directs the inference task through its security controls:

1. **Automated Evaluation:** The protocol first appraises task characteristics – sensitivity, model size, computational complexity, and desired consensus.
2. **Confidentiality Routing:** Based on the assessment, an optimal privacy approach is selected, engaging either hardware or cryptographic composite or a hybrid blend as warranted by the task.
3. **Verifiability Assurance:** Concurrently, ZKPs are orchestrated to align with the chosen confidentiality mode, ensuring that computations are demonstrably accurate without revealing sensitive data.
4. **Result Synthesis:** The multi-party computations are conducted within the confines of the confidentiality construct, producing aggregate inference data.
5. **Validation:** Results are aggregated and validated to ensure correctness and avert any adversarial influence, at which time they are rolled up for settlement

Previous
Challenges in Achieving Confidentiality and Verifiability

Next
Composite 1: Hardware

Last updated 1 month ago