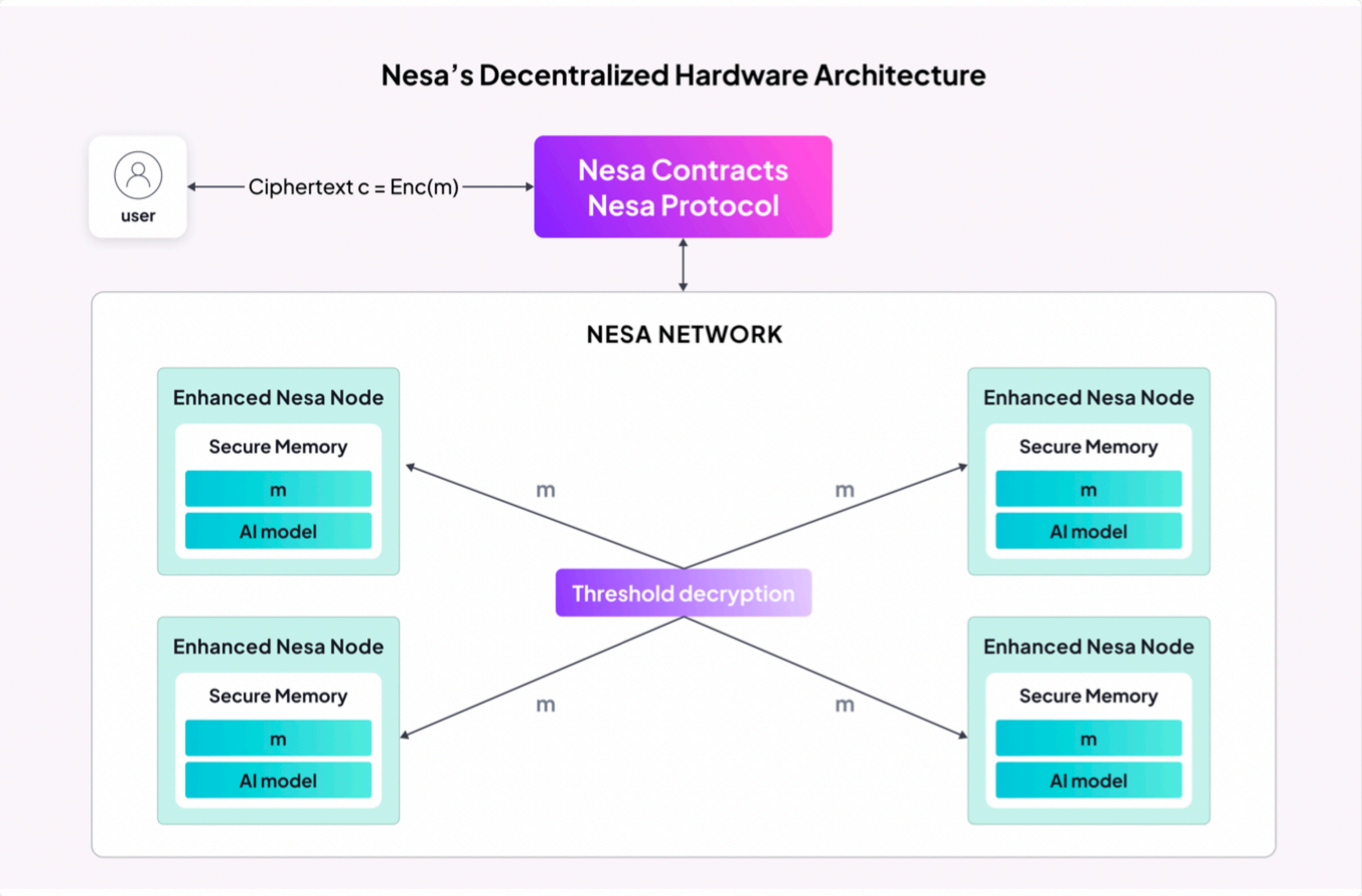
Composite 1: Hardware

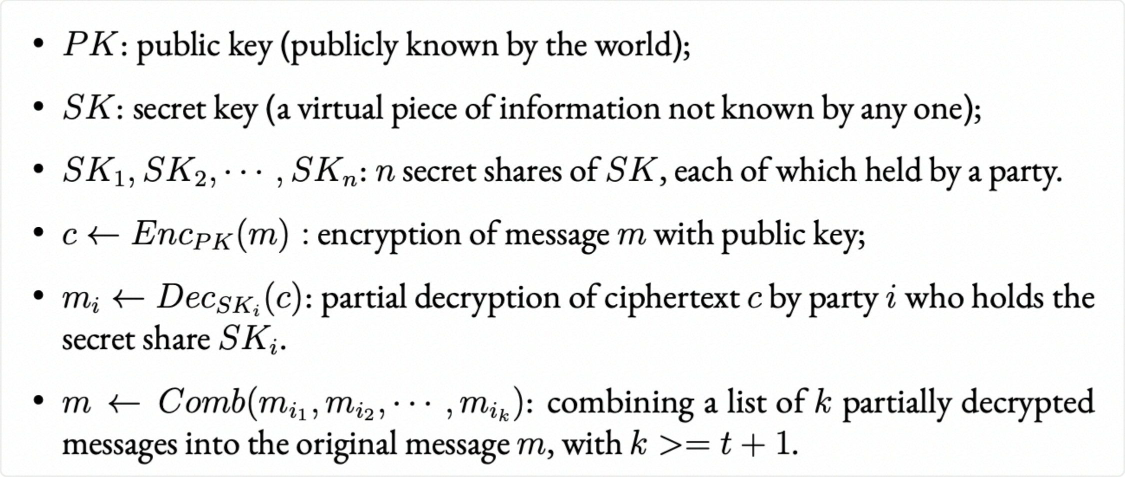
To address the challenges of privacy and verifiability, our system leverages specialized hardware provisioned with enhanced security features for both CPU and GPU to fortify computation nodes. Some nodes in the NES network are designated as enhanced nodes equipped with TEEs, which offer a secure enclave for processing sensitive data. This setup is particularly adept at protecting the user's input from exposure while still enabling computation.

Complimentary to the hardware requirement, Nesa relies on a threshold cryptosystem, because the user submits a request that is publicly accessible, hence the request should contain encrypted information that can be decrypted (in secure memory) by a randomly selected committee. Basically, in a (t, n)-threshold cryptosystem, we have:



Nesa's Hardware Composite. Each participating node in the inference is equipped with advanced hardware with confidential computing capability. A user can submit ciphertext in the request transaction.

The ciphertext can be decrypted only inside secure memory in the committee of nodes, through threshold decryption. The decryption reveals the plaintext in the secure memory, enabling each node to continue performing subsequent inference computation.



- PK: public key (publicly known by the world);
- SK: secret key (a virtual piece of information not known by any one);
- $SK_1, SK_2, \cdots, SK_n$: n secret shares of SK, each of which held by a party.
- $c \leftarrow Enc_{PK}(m)$: encryption of message m with public key;
- $m_i \leftarrow Dec_{SK_i}(c)$: partial decryption of ciphertext c by party i who holds the secret share SK_i.
- $m \leftarrow Comb(m_{i_1}, m_{i_2}, \cdots, m_{i_k})$: combining a list of k partially decrypted messages into the original message m, with $k >= t + 1$.

Here is a step-by-step breakdown of the protocol employed by our system that utilizes hardware-based solutions to ensure privacy-preserving computation shown above:

1. Inference Request Preparation: The user begins by encrypting his input data *m*,

by

$$c = Enc_{PK}(m),$$

using a public key that is part of a public-secret key pair associated with the enhanced nodes. This key pair is specially designed so that the corresponding secret key is not held by any single node but is rather distributed amongst the computation nodes using secret sharing techniques. The user then creates an inference request transaction, embedding the encrypted input (ciphertext) rather than plaintext data. The user also specifies that this is an inference request with enhanced privacy so that only nodes with proper hardware will participate in the committee selection.

2. Transaction Submission: The inference request transaction, containing the encrypted input, is submitted to the blockchain and queued according to the priority system outlined in Chapter 3.
3. Committee Selection: Upon reaching the head of the queue, an inference committee S is selected using the VRF technique, ensuring a fair and random choice of enhanced nodes from among those equipped with TEEs.
4. Threshold Decryption: The selected committee nodes (withsize|S|>t) retrieve the encrypted input from the transaction. Inside the secure enclaves provided by their TEEs,

committee nodes collaborate to perform a threshold decryption operation. Basically, they perform

$$m_i = Dec_{SK_i}(c),$$

followed by a threshold combining

$$m = Comb(\{m_i\}_{i \in S}).$$

This process ensures that only a collective effort by the committee can reconstruct the secret key and decrypt the input, thus no single node has access to the plaintext data.

5. Inference Execution: Once decrypted, the plaintext input remains within the protected memory space of the TEEs, where the inference computation is securely executed. The TEE ensures that the computation process is isolated from the rest of the system, providing a safeguard against potential leaks or attacks.
6. Result Encryption and Submission: The output of the inference computation, still within the TEE, is encrypted with a public key provided by the user.

An encrypted inference result is then generated and submitted to the blockchain as a transaction. This result can only be decrypted by the user, maintaining the confidentiality of the data. This hardware-based solution provides strong security guarantees for user input by combining encryption, threshold decryption within TEEs, and secure computation.

The protocol addresses the dual challenges of confidentiality and verifiability by ensuring that the sensitive data is never exposed in an unencrypted form outside of the secure enclaves and that the computation is performed within an environment that is resilient to tampering.

By maintaining the secrecy of the data throughout the process and leveraging the trusted hardware, the system provides a robust framework for performing privacy-preserving computations on a decentralized network.

TEE Setup and Attestation Protocol

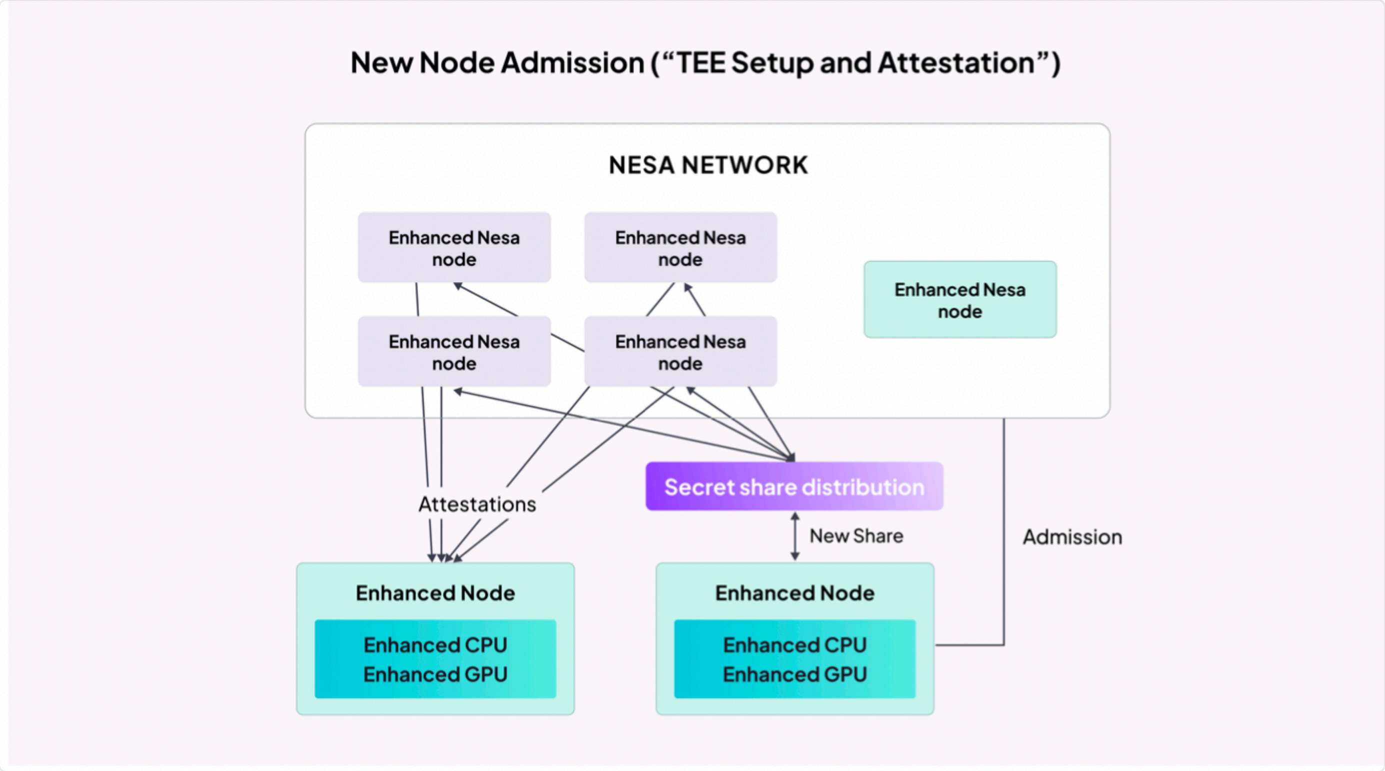
To ensure that enhanced nodes with TEEs are trustworthy and properly configured, we implement an attestation protocol similar to the decentralized trust mechanisms present in

Chainlink's Oracle network. This attestation process is critical as it not only assures that the TEE is genuine and secure but also serves as the initiation protocol for new nodes entering the system. The node admission procedure is shown and discussed below:

1. Node Initialization: When a new computation node equipped with a TEE wishes to join the network, it must first perform a local attestation of its TEE. This step involves generating an attestation report that certifies the authenticity of the TEE and provides details about the environment's configuration. This is called a quote Qte(hw, env).
2. Network Attestation: The new node then broadcasts its attestation report Qte (hw,env) to the existing nodes in the network. A threshold number (larger than t) of established nodes must validate the newcomer's attestation report. This is a crucial step as it ensures that no single node can unilaterally admit a new member into the network, thereby decentralizing trust.
3. Secret Share Distribution: Upon successful verification of the attestation report, the existing nodes collaborate to spin up a new share of the secret key for the new node. This process utilizes secure multi-party computation protocols to ensure that the new node receives its share without any single node ever having possession of the complete secret key. The threshold secret sharing is n evaluations of a randomized polynomial

$$p(x) = s + r_1 x + r_2 x^2 + \cdots + r_t x^t,$$

with s being the secret, and x = 1,2,⋯ , n. Hence, to accept a new node with an index n + 1, a threshold of the existing nodes collectively evaluates the polynomial



New node admission. To admit a new node into the network, the new node has to attest to a number of existing nodes. Upon successful attestations, the existing nodes will collectively generate a new share of the secret key for the new node, enabling it to become a member that is capable of participating in the hardware-based secure inference procedure shown in Nesa's Hardware Composite

on a new point p(n + 1). Basically, the way to perform such an evaluation in a decentralized way is by using a distributed version of the Lagrange interpolation. Suppose we have t + 1 evaluations:

$$(i_1, a_{i_1}), \cdots, (i_{t+1}, a_{i_{t+1}}),$$

where ai =p(i) is the secret shared by node i. By applying the Lagrange interpolation formula, we have:

$$p(n+1) = \sum_{j=1}^{t+1} a_{i_j} \lambda_j,$$

where λj is the Lagrange interpolation coefficient:

$$\lambda_j = \prod_{k=1, k \neq j}^{t+1} \frac{i_k - n - 1}{i_k - i_j}.$$

Note that *aij λ j can be computed locally by node ij.* Therefore each node can compute a Lagrange interpolation term and send it to the new node that will aggregate and obtain p(n + 1) as a new share of the secret.

4. Key Share Integration: The new node integrates the received share of the secret key into its TEE. With the combination of the key share and the TEE's secure en- clave, the node is now capable of participating in threshold decryption operations as part of an inference committee.
5. Node Admission: The node is now considered a trusted member of the network and can participate in inference tasks. For ongoing assurance, the attestation process can be periodically repeated to confirm the integrity of the TEE.

The attestation and key share distribution processes are essential for maintaining a secure and decentralized environment, preventing any single point of failure in the security model. Each node's TEE serves as a strong anchor of trust, with the assurance that it has not been tampered with and is running the correct software.

By distributing the responsibility of attesting and generating secret shares across multiple nodes, we ensure that the system remains resilient and can dynamically adapt as new nodes join or leave the network.

This subsection outlines how TEE attestation is integral to ensuring that all computation nodes meet the stringent security standards required to protect the confidentiality and integrity of user data, resisting any form of participant collusion.

Previous
Split-Flow

Next
Composite 2: Cryptography

Last updated 1 month ago