**Block Verification**

Created by Hgminerva

Introduction

Xode Parachain operates under Kusama Parachain 3344 and is built using the Substrate Parachain Template. This documentation outlines the security mechanisms and block verification process to ensure transaction integrity, consensus safety, and chain finality.

**Block Verification**

**Transaction Validation**

Every transaction undergoes multiple validation steps before being included in a block:

* Signature Verification: Ensures transactions are signed by valid accounts using Schnorrkel (Sr25519) or Ed25519 cryptographic schemes.
* Nonce Checking: Prevents replay attacks by ensuring the transaction nonce is sequential.
* Balance Validation: Ensures the sender has sufficient balance for execution, including fees.
* Weight Calculation: Assigns a weight to the transaction to avoid exceeding block limits.

**Block Production**

* Blocks are produced using the AURA (Authority Round) mechanism:
* Validator-Driven Block Production: Validators take turns producing blocks in a round-robin fashion.
* Slot-Based Leader Selection: Validators are assigned slots to produce blocks, with the leader selected based on the AURA protocol.
* Fork Choice Rule: The longest valid chain is selected using the GHOST-based rule.

**Collator Node Verification**

* Collators: Prepare blocks by collecting transactions and executing extrinsics.
* State Transition Validation: Ensures the resulting state of a block is valid by executing the runtime logic.
* Proof-of-Execution Generation: Collators submit state proofs for verification by validators.

**Parachain-to-Relay Chain Communication**

* Collators Submit Blocks to Validators: Blocks are sent to the relay chain validators for cross-validation.
* Candidate Validation: Kusama relay chain validators verify parachain blocks using PVF (Parachain Validation Function).
* Consensus Agreement: Blocks are finalized upon agreement through GRANDPA.

**More Information**

<https://wiki.polkadot.network/docs/learn-parachains-protocol>