Cyberscope

Review Report

CrossFi

February 2024

Repository https://github.com/crossfichain/mainnet

Reviewed by © cyberscope

Table of Contents

Review

Repository	https://github.com/crossfichain/mainnet
Commit	a04a0724f6b95c8c72b403205084bb5633ae8a65

Review Updates

Review Date	01 Feb 2024

Overview

Cross Finance (CrossFi) is a cutting-edge ecosystem that develops sophisticated and contemporary payment solutions. They employ their own proprietary CrossFi technology, which combines the security and transparency of blockchain technology with the stability of conventional finance.

Setup Process

Required Steps

The "README.md" file provides a set of commands for initializing and starting a CrossFi mainnet node:

1. Download and Extract Node Software:

   ```
   wget https://github.com/crossfichain/crossfi-node/releases/download/v0.1.1/mineplex-2-node._v0.1.1_linux_amd64.tar.gz && tar -xf mineplex-2-node._v0.1.1_linux_amd64.tar.gz
   ```

   This command downloads the specific version (v0.1.1) of the CrossFi node software (mineplex-2-node._v0.1.1_linux_amd64.tar.gz) and extracts the downloaded tar.gz file, making the node software available for use.

2. Clone the Mainnet Repository:

   ```
   git clone https://github.com/crossfichain/mainnet.git
   ```

   This command clones the mainnet repository from GitHub to the local machine, providing access to the required configuration files for starting the node.

3. Start the CrossFi Node:

   ```
   ./mineplex-chaind start --home ./mainnet
   ```

   This initiates the CrossFi node with "mineplex-chaind", the main executable and specified the directory where the node should look for configuration files (./mainnet directory, where the repository was cloned)

In-Depth Analysis

By executing these commands, a user sets up a node that connects to the CrossFi mainnet, effectively participating in the network's maintenance and consensus processes. This process involves downloading the node software, acquiring necessary configuration files, and initiating the node's operation. The combination of these configuration files ensures that the node operates both efficiently and securely, adhering to the network’s rules and contributing to its overall health and decentralization.

Configuration Files

"genesis.json" File

The genesis.json file serves as the foundational blueprint for the blockchain network, setting the initial parameters and configurations at the genesis (creation) of the blockchain. The following table includes key information from this file:

Field	Value	Description
genesis_time	2023-03-30T08:30:00Z	The timestamp for the genesis (creation) of the blockchain
chain_id	mineplex-mainnet-1	Unique identifier of the blockchain network
initial_height	1	The starting block height of the blockchain
consensus_params -> block	max_bytes: 22020096, max_gas: -1, time_iota_ms: 1000	Block size limit, gas limit, and minimum time between blocks in milliseconds
consensus_params -> evidence	Max_age_num_blocks: 100000, Max_age_duration: 172800000000000, max_bytes: 1048576	Parameters defining the handling of evidence of malfeasance on the network
consensus_params -> validator	pub_key_types: ["ed25519"]	Lists the public key types supported for validators
app_hash	(empty string)	Hash of the application state at genesis
auth -> params	max_memo_characters: 256, tx_sig_limit: 7 tx_size_cost_per_byte: 10, Sig_verify_cost_ed25519: 590, Sig_verify_cost_secp256k1: 1000	Parameters for transaction authentication, like maximum memo characters and signature verification costs

auth -> accounts	List of accounts	Initial accounts created at genesis, including their addresses and balances
bank -> params	send_enabled: [], default_send_enabled: true	Bank module parameters, including the status of token transfer capabilities
bank -> balances	List of balances	Initial token balances for the accounts, denoting the amount and denomination
crisis -> constant_fee	amount: 10000000000000000000, denom: "mpx"	The fixed fee for crisis transactions, denoting the amount and the denomination of the fee
distribution -> params	Various settings	Parameters for the distribution module, including proposer rewards, community tax, and withdrawal settings
genutil -> gen_txs	Various transaction details	Initial transactions included in the genesis block, such as validator creation transactions
genutil -> gen_txs -> MsgCreateValidator	Various settings	Details of a validator creation transaction including commission rates, delegation, and validator information
pubkey	Cosmos.crypto.ed25519.PubKey, key value	Public key of the validator
value	denom: "mpx", amount: large value	The amount and denomination of tokens staked by the validator
MsgCreateValidator (MinePlex2)	-	Details of the second validator's creation including commission, delegation, and validator info
commission (MinePlex2)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator

min_self_delegation (MinePlex2)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex2)	mx1vxd9v5t7dffxp5kwjl6n3huqxw8uhh5v59hj69	Address of the delegator for MinePlex2
validator_address (MinePlex2)	mxvaloper1vxd9v5t7dffxp5kwjl6n3huqxw8uhh5vqwwvm6	Validator address for MinePlex2
pubkey (MinePlex2)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex2 validator
value (MinePlex2)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex2 validator
MsgCreateValidator (MinePlex3)	-	Details of the third validator's creation including commission, delegation, and validator info
commission (MinePlex3)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex3)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex3)	mx1le5gsa4qxweud3tuc4upn5p320nze8zdszyhv9	Address of the delegator for MinePlex3
validator_address (MinePlex3)	mxvaloper1le5gsa4qxweud3tuc4upn5p320nze8zdyfafd6	Validator address for MinePlex3
pubkey (MinePlex3)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex3 validator
value (MinePlex3)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex3 validator

MsgCreateValidator (MinePlex4)	-	Details of the fourth validator's creation including commission, delegation, and validator info
commission (MinePlex4)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex4)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex4)	mx16dp99m9kl83yeww6j7jyd9v3qzm2cy3qy9cdyj	Address of the delegator for MinePlex4
validator_address (MinePlex4)	mxvaloper16dp99m9kl83yeww6j7jyd9v3qzm2cy3qswpn9d	Validator address for MinePlex4
pubkey (MinePlex4)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex4 validator
value (MinePlex4)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex4 validator
MsgCreateValidator (MinePlex5)	-	Details of the fifth validator's creation including commission, delegation, and validator info
commission (MinePlex5)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex5)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex5)	mx1pfyz7tyk297p3cfl78fgt9esud4eclcegyf9np	Address of the delegator for MinePlex5
validator_address (MinePlex5)	mxvaloper1pfyz7tyk297p3cfl78fgt9esud4eclceu0smj7	Validator address for MinePlex5

pubkey (MinePlex5)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex5 validator
value (MinePlex5)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex5 validator
MsgCreateValidator (MinePlex6)	-	Details of the sixth validator's creation including commission, delegation, and validator info
commission (MinePlex6)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex6)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex6)	mx1u2y6apvc934zdss5rcgqstv99kshf9a38kh9wd	Address of the delegator for MinePlex6
validator_address (MinePlex6)	mxvaloper1u2y6apvc934zdss5rcgqstv99kshf9a3nawm0j	Validator address for MinePlex6
pubkey (MinePlex6)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex6 validator
value (MinePlex6)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex6 validator
MsgCreateValidator (MinePlex7)	-	Details of the seventh validator's creation including commission, delegation, and validator info
commission (MinePlex7)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex7)	1	Minimum self-delegation amount required for the validator

delegator_address (MinePlex7)	mx1xpsx8dkyqncey9kxkwumtu4n0mgg5vwh7s6kpe	Address of the delegator for MinePlex7
validator_address (MinePlex7)	mxvaloper1xpsx8dkyqncey9kxkwumtu4n0mgg5vwh2mrgqx	Validator address for MinePlex7
pubkey (MinePlex7)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex7 validator
value (MinePlex7)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex7 validator
MsgCreateValidator (MinePlex8)	-	Details of the eighth validator's creation including commission, delegation, and validator info
commission (MinePlex8)	rate: 0.1, max_rate: 0.2, max_change_rate: 0.01	Commission rates for the validator
min_self_delegation (MinePlex8)	1	Minimum self-delegation amount required for the validator
delegator_address (MinePlex8)	mx17400kwy96hlyv6lc4tjwc2kkspma2j74hgdq29	Address of the delegator for MinePlex8
validator_address (MinePlex8)	mxvaloper17400kwy96hlyv6lc4tjwc2kkspma2j74rr57t6	Validator address for MinePlex8
pubkey (MinePlex8)	/cosmos.crypto.ed25519.PubKey, key: [key value]	Public key of MinePlex8 validator
value (MinePlex8)	denom: "mpx", amount: 500000000000000000000000000	Staking amount for MinePlex8 validator
gov -> deposit_params	max_deposit_period: "172800s", min_deposit: [amount: 100000000000000000000, denom: "mpx"]	Governance parameters for depositing proposals, including maximum deposit period and minimum deposit amount and denomination

gov -> tally_params	quorum: 0.334, threshold: 0.5, veto_threshold: 0.334	Parameters for proposal tallying in governance, including quorum, threshold, and veto threshold percentages
gov -> voting_params	voting_period: "172800s"	The voting period length for governance proposals
mint -> params	Various settings	Parameters for minting, including reward per block for different periods
slashing -> params	downtime_jail_duration: "600s", min_signed_per_window: 0.5, signed_blocks_window: 100, slash fractions	Parameters related to validator slashing for downtime and double signing
staking -> params	bond_denom: "mpx", historical_entries: 10000, max_entries: 7, max_validators: 64, unbonding_time: "1296000s"	Staking parameters including bond denomination, validator limits, and unbonding time
transfer -> params	receive_enabled: true, send_enabled: true	Parameters for enabling or disabling token transfers.
treasury -> params	owner: "mx1anctd79de6ry2m0u2va9x8tcls9jeflhg3r6ys"	Treasury parameters including the owner address

"config.toml" File

The "config.toml" file is an important component in configuring the operational parameters of a blockchain node. It provides a structured format to define various settings that influence the node's performance, networking, consensus mechanisms, security features, and other critical operational aspects. The following table includes key information from this file:

Field	Value	Description
abci	socket	ABCI connection type used by the blockchain node
db_backend	goleveldb	Database backend for storing blockchain data
fast_sync	true	Indicates if fast synchronization is enabled for the node
genesis_file	config/genesis.json	Path to the genesis file
log_format	plain	Format of the logging output
log_level	info	Level of logging detail
moniker	seed	Name identifier for the node
consensus -> create_empty_blocks	true	Whether to create empty blocks when there are no transactions
consensus -> timeout_commit	5s	Time to wait before committing a block
fastsync -> version	v0	Version of the fast sync algorithm used

mempool -> max_tx_bytes	1048576	Maximum size of a single transaction
p2p -> max_num_inbound_peers	40	Maximum number of inbound peers.
p2p -> max_num_outbound_peers	10	Maximum number of outbound peers
p2p -> persistent_peers	[List of peers]	List of persistent peers to maintain connections with
rpc -> max_open_connections	900	Maximum number of simultaneous connections (including WebSocket and RPC)
rpc -> laddr	tcp://0.0.0.0:26657	RPC server listen address
statesync -> enable	true	Whether state sync is enabled, allowing faster catching up of a node
statesync -> trust_height	500000	Block height at which the trusted header was fetched to initialize state sync
tx_index -> indexer	kv	Transaction indexing mechanism

"app.toml" File

The "app.toml" file contains configurations specific to the application layer of the blockchain node. This configuration file focuses on aspects like API and gRPC server settings, database options, pruning strategies, and telemetry settings. The following table includes key information from this file:

Field	Value	Description
app-db-backend	(empty)	Specifies the backend database used for the application
halt-height	0	Block height at which the node will automatically halt
iavl-cache-size	781250	Size of the IAVL tree cache
minimum-gas-prices	"10000000000000mpx"	Minimum gas prices for processing transactions
pruning	"default"	Pruning strategy used for the state database
[api] -> address	"tcp://0.0.0.0:1317"	Address for the API server
[api] -> enable	true	Whether the API server is enabled
[api] -> max-open-connections	1000	Maximum number of open connections to the API server
[grpc] -> address	"0.0.0.0:9090"	Address for gRPC server
[grpc] -> enable	true	Whether the gRPC server is enabled
[grpc-web] -> address	"0.0.0.0:9091"	Address for the gRPC-Web server

[grpc-web] -> enable	true	Whether the gRPC-Web server is enabled
[rosetta] -> enable	false	Whether the Rosetta API server is enabled
[state-sync] -> snapshot-interval	0	Interval at which state sync snapshots are taken
[state-sync] -> snapshot-keep-recent	2	Number of recent state sync snapshots to keep
[streamers.file] -> write_dir	(specified path)	Directory where streamer files will be written
[telemetry] -> enabled	false	Whether telemetry is enabled to collect and expose metrics

"client.toml" File

The client.toml file is crucial for setting up the client-side configurations of the blockchain node. It specifies how the client interacts with the blockchain network and the node it connects to. The following table includes key information from this file:

Field	Value	Description
broadcast-mode	"block"	Determines the broadcast mode for transactions
chain-id	"mineplex-mainnet-1"	Identifier for the blockchain network the client connects to
keyring-backend	"file"	Specifies the backend for storing keys
node	"tcp://localhost:26657"	Address of the node to connect to for blockchain operations
output	"text"	Format for command output (e.g., text, JSON).

Summary

Cross Finance (CrossFi) implements a blockchain infrastructure. This report has provided an in-depth analysis of this blockchain, spotlighting the technical nuances and operational mechanics as defined by its key configuration files.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io