Munhumutapa©

Munhumutapa Alphablockchain Technical Paper

www.munhumutapa.c1.biz

Architecture

Munhumutapa Alphablockchain is made up of a two tier model;

3. Local Chains

These types of chains are built on top of betachains and are built for small local communities to allow them to create local ledgers for various uses and information storage and tracking. An example is the use of the identity betachain based local chain which may help universities issue digital student ID's which are loaded with a currency and are used to pay for food and library fees.

Such local chains will help small organisations to create structures and systems based on blockchain technologies minus the complexities and need for expert services.

2. Betachains

These are child chains built on top of Munhumutapa which will offer different, economic specific services to users. betachains are designed with client based properties advanced on the alpha-chain while utilizing the infrastructure and software system of Munhumutapa.

Transactions executed on these betachains are confirmed and validated by a network of permissioned nodes called binders which combine the betachain transactions into betablocks.

For each betachain is a betablock submitted to the Munhumutapa alpha-chain.

All the betachain services are integrated onto the alphablockchain. This innovative mechanism allows addresses to connect and disconnect from chains seamlessly for security and privacy.

Each child chain will have a specific core function where apps built on it have access only to a chain's core service.

1. Alpha-chain

The main blockchain, Munhumutapa, will be were alpha-transactions are added to the forged blocks. The alpha-transactions are grouped together with transactions on the alpha-chain and a block is forged and added to the Alphablockchain.To create a block, one of the nodes is randomly selected through a Proof-Of-Stake protocol to forge a block which is then added to the Munhumutapa blockchain.
Alpha-chain transactions are limited to Munhumutapa Energy transfers only.

Accounts & Addresses

Munhumutapa adopts 25519 Elliptic Curve and SHA 256 as its underlying cryptographic base. For a user to connect to the blockchain, an account has to be initially created. At the first instance of account creation, a passphrase is generated for the user which is private and only

visible to the user. This passphrase is an account's private key and is solely the user's responsibility and consequently, its safe-keeping. Losing the passphrase will result in the loss of funds which will be forever locked in the account.

However, an account is not activated i.e added to the blockchain until an outgoing transaction is executed and effectively confirmed.

Munhumutapa accounts take the form of a 64 -bit number address which is generated by a series of operations from the account's passphrase. The operation is thus summarised as;

The private key is hashed from the account passphrase using SHA256 and subsequently encrypted into a 256-bit public key through the use of Curve25519.

To create an account ID, the public key undergoes another SHA256 hashing and encoding to give an address of the format: MUNHU- xxxx-xxxx-xxxx-xxxxx

Account States

An account is a user's gateway to the Munhumutapa blockchain and is composed of varying degrees of states depending on its transactions. These states are:

A. Account Balance

- Active account balance i.e Munhumutapa Coin, Munhumutapa Energy Coin, mCurrencies, Mutapa Energy Coins and totems available for transactions.

- Transitional account balance i.e balances of unconfirmed transactions. If confirmed, the active and transitional balances are adjusted accordingly.

- Forge Rewards account balance i.e balances acquired from forging blocks.

B. Associated data files

- files include identity documents, mAUTH documents associated with the account.

C. Single or multi-signatory state and/or associations

D. Forging History

E. Leasing History

Address Collisions

As more accounts are created on the blockchain, instances of similar addresses being generated are very much possible. These instances are destroyed by ensuring public keys generated first are the only one permitted and the rest are rejected.

Transactions on Munhumutapa

To send and receive Munhumutapa and/or totems, assets and files, a user needs to have an active account.

An account also needs to hold a sufficient amount of energy fees payable only in Energy to execute an outgoing transaction. There is no energy fees paid for an incoming transaction.

Transaction types on Munhumutapa:

- Payments and Funds Transfers

These are transactions which involve the transfer of funds from one to another.

- Account Control Transactions

These are set and limited types which include:

Vaulting in Munhumutapa Bank
Lending in Munhumutapa Bank
Balance leasing for forging on the Munhumutapa alphachain.

-*Messaging*

Plain or Encrypted

Users can send plain text or encrypted messages from one account to the other.

Voting

As a democratic economy, users can vote on Munhumutapa on different given polls. Voting can be initiated for appointments, funds use etc.

Transaction Process

For an outgoing transaction, an account with a sufficient balance of MUNHU/MUTAPA and token/coin sends its transaction to a node which processes that transaction and broadcasts it to other nodes in the network.

1. The sender sets the conditions and parameters for the transaction. The parameters and conditions set include;

   A private key for the account
   A sufficient energy fee for the transaction i.e transaction fee.
   Execution time period limit.
   A transaction reference message (optional)

2. All values for the transaction inputs are checked. For example, mandatory parameters must be specified; fees cannot be less than or equal to zero; a transaction deadline cannot be less than one minute into the future; if a referenced transaction is specified, then the current transaction cannot be processed until the referenced transaction has been processed.

3. If no exceptions are thrown as a result of parameter checking:

The public key for the generating account is computed using the supplied secret passphrase

Account information for the generating account is retrieved, and transaction parameters are further validated:

   The sending account's balance cannot be zero
   The sending account's balance must not be lower than the transaction amount plus the transaction fee.

4. If the sending account has sufficient funds for the transaction:

   A new transaction is created, with a type and subtype value set to match the kind of transaction being made. All specified parameters are included. A unique transaction ID is generated with the creation of the transaction request.
   The transaction is signed using the sending account's private key
   The encrypted transaction data is placed within a message instructing network peers to process the transaction
   The transaction is broadcast to all peers on the network.

The transaction is then included in a block and added to the blockchain. This process is however only for transactions which are executed on the alphachain i.e transactions that alter mCurrencies, Munhumutapa Coin and Munhumutapa Energy Coin balances.

Betachain Transactions

Binding

Betachain transactions are not processed on the alphachain as are Munhumutapa Coin and Energy. Instead, betachain transactions are grouped into betablocks by eligible nodes through the process of binding. The betablock only carries the transactions of a particular betachain and these transactions are denominated in the chain own energy token/coin. Transaction fees for betachains are paid in their native coins to the binders who will pay the transaction fees in MUNHUe to the forgers. The betablocks are then included as a single beta-transaction to forgers. Betablocks can have a maximum of 100 betachain transactions.

Binding Pricing

Each binder has the freedom to set their own binding fee provided it is not less than the minimum default fee. Binders will broadcast their fees to users who also can choose their own preferred binder based on fees charged or trust. Users can also run personal binders for themselves especially those with frequent transaction submissions.

Forging

Forging Rewards

Munhumutapa Energy balances are the accepted as stake on Munhumutapa. The fees paid for each transaction by binders are the rewards received by a forger when they successfully forge a block.The blockchain already has a minimum acceptable energy fee which ensures transactions are guaranteed confirmation. However due to high level traffic in some time periods, forgers may prioritise including transactions offering more rewards in their forged blocks. In such cases, a user might opt to increase the amount of fees associated with their transactions for them to be confirmed faster.

Balance Leasing

Munhumutapa allows accounts to lease each other balances for forging. With balance leasing on the blockchain, a user's balance is not transferred to the leasee's account but only its forging power. This means the balance leaser can spend and control their funds norm-Storey even when they are leasing their balance. The obvious use of this feature is the creation and running of forging pools where accounts put together their forging power to boost their forging probability and share the rewards. The shared reward for each pool contributer will depend on the agreed conditions and the trust of network users.

Transaction Validations

Nodes

Nodes are any devices running the Munhumutapa blockchain at any given time.Transactions are processed by nodes on the network.

When online, each node can process and validate transactions as well as information from forged blocks. Nodes also participate in the forging process as long as they meet the set requirements and receive and broadcast information which includes valid transactions and blocks. In a proposed (currently being developed sys) system, nodes will have a global trust value based on the Eigen Trust Algorithm

Blocks

A block is a collection of valid transactions executed on Munhumutapa alphachain and binded betachain transactions. Blocks on Munhumutapa contain a maximum of 10 transactions and a block is forged every 60 seconds. Only one betablock per betachain can be included in a forged block. These blocks are produced in succession in a chain-like organisation from which term, blockchain is derived. Once a block is added to the chain, it is permanent and cannot be altered in any way. Each node in the network thus stores this blockchain.

Block Composition

A block also carries other information on top of the transaction records. It has its own identification parameters stored on it for references and validation. The block contains;

- ➢ A block version, height and block identification tag
- ➢ A block's timestamp
- ➢ A hash of the previous epoch and its timestamp
- ➢ Slot identity tag
- ➢ The public key and identification of its forger
- ➢ The ID and hash of the previous block
- ➢ The number of transactions stored in the block
- ➢ The total amount of MUNHU carried by the block.
- ➢ transaction data of entries in the block and respective transaction identification
- ➢ The payload length and hash value of its payload
- ➢ The block’s generation signature
- ➢ A signature for the entire block

Consensus: Proof-Of-Stake (PoS)

Proof of Stake Protocols

Munhumutapa blockchain will initially use a naive Proof-of-Stake protocol to reach consensus before upgrading to the Munhumutapa PoS protocol after six months. The change will coincide with the launch of the mCurrency system. Although the Nxt's proof -of-stake algorithm is relatively secure, Munhumutapa PoS will significantly enhance the security of the alphablockchain.

Pure Proof of Stake

Munhumutapa currently uses a Target and Hit random selection algorithm to select a forger. To be determined as the forger, a candidate's Hit value has to be less than the Target value.

A Hit value is determined by:

1. A forger's Public key

2. Signature hash of the previous block.

The Target for a candidate forger is determined by:

A. Uniform parameters;

1. *Base Target*

The base target is a value of the the time period between the forging of the previous two blocks. This parameter is the same for every forging eligible account.

2. TimeSinceLastBlock

Simply the time elapsed since the last forged block.

B. Non-Uniform parameter;

1. Effective balance of candidate.

The Effective balance of an account is the MUNHU an account holds which has been static for 1440 blocks or more.

2. Number of Connected Peers
The number of peers the node is currently connected to.

Formula
To get the target value, the following formula is used:

Target =

*Base Target * Time Since Last Block * Effective balance * ConnectedPeersCount*

Computing the Hit Value

Each block on the Munhumutapa blockchain has a generation signature parameter. During the block forging process, a forger cryptographically signs the block with its public key to create a 64-byte signature, which is then hashed using SHA256. The first 8 bytes of the resulting hash is the hit value.

Picking A Forger

The hit value and target value are subsequently compared and if the condition:

Target value > Hit value,

is satisfied then the account gets to forge the next block.

Cumulative Difficulty

To determine an authoritative block in an instance when two or more blocks are forged at once, a cumulative difficulty value is used.

The cumulative difficulty value is derived from the base target value as follows:

Let,

D cb is the difficulty of the current block

D pb is the difficulty of the previous block

T b is the base target value for the current block,

Then

$$Cumulative\ Difficulty = Dpb + (2^{64})/Tb$$

So when multiple blocks are forged and proposed simultaneously, the block with the highest cumulative difficulty is taken as the progressive and authoritative one.

Security - Proof of Stake Attacks

Nothing at Stake

In a nothing at stake attack, forgers attempt to build blocks on top of every fork they find regardless of validity and since costs are minimal this attack is theoretically possible. However, the low block reward does not warrant such an a attack as the financial gain is almost meaningless.

History Attack

In a history attack, an attacker tries to double-spend by attempting to create a successful fork from just before the time when their tokens were sold or traded. If the attack fails, the attempt costs nothing because the tokens have already been sold or traded; if the attack succeeds, the attacker gets their tokens back.To successfully carry out such an attack needs one to obtain the private keys from old accounts and use them to build a chain starting from the genesis block.

Shuffling Attack

On Munhumutapa, a history attack should fail because of the static stake rule which enforces the condition that funds must be static for 1440 blocks before an account can forge and the coupled with the verification of the balance of the account that generates each block.This is also circumvented on Munhumutapa as the blockchain cannot be re-organized more than 1440 blocks behind the current block height.

51% Attacks

Under Munhumutapa, the distribution and value of MUNHU makes such an attack very expensive and ultimately lead to an immeasurable loss. For an attacker to acquire 51% of the entire stake, they would have to spend in excess of $300 million provided they convinced MUNHU holders to sell. Even if such an amount is acquired, successfully carrying out the attack could not possibly match the value expended on the attack.

Grinding Attacks

In such an attack, a dishonest party influences the forger picking algorithm to select their own choice and consequently influence the chain's continuation path. Since Munhumutapa initially uses a hit and target algorithm, each account can only generate a value unique to itself which is determined by the effective balance of an account subject to the 1440 static stake rule thereby isolating accounts from outside influence.

Selfish Forging

In this type of attack, a forger may withhold blocks and stifle chain progress. It is countered on Munhumutapa by having the target continuously increasing until a new forger is selected and thus skipping the selfish forger making their withheld block invalid and will be discarded. Also, the chain can only be reorganised only 720 blocks deep into the chain which discourages such a form of attack.