Whitepaper:CRYT

Last Update: 25/07/2019 - v 1.11.15

Indice

1 Abstract

Bitcoin has proven that a peer-to-peer electronic cash system can indeed work and fulfill payments processing without requiring trust or a central mint. However, for an entire *electronic economy* to be based on a fully decentralized, peer-to-peer solution, it must be able to do the following: process transactions securely, quickly and efficiently, at the rate of thousands per hour or more; provide incentives for people to participate in securing the network; scale globally with a minimal resource footprint; offer a range of basic transaction types that launch cryptocurrencies past the core feature of a payment system alone; provide an agile architecture that facilitates the addition of new core features, and allows for the creation and deployment of advanced applications; and be able to run on a broad range of devices, including mobile ones. CRYT satisfies all these requirements.

2 Introduction and Overview

CRYT is a 100% proof-of-stake cryptocurrency, constructed in open-source Java. CRYTs unique proof-of-stake algorithm does not depend on any implementation of the coin age concept used by other proof-of-stake cryptocurrencies, and is resistant to so-called nothing at stake attacks. A total quantity of 1 billion available tokens were distributed in the genesis block. Curve25519 cryptography is used to provide a balance of security and required processing power, along with more commonly-used SHA256 hashing algorithms.

Blocks are generated every 20 seconds, on average, by accounts that are *unlocked* on network nodes. Since the full token supply already exists, CRYT is redistributed through the inclusion of transaction fees which are awarded to an account when it successfully creates a block. This process is known as *forging*, and is akin to the mining concept employed by other cryptocurrencies. Transactions are deemed safe after 10 block confirmations, **and CRYTs current architecture and block size cap allows for the processing of up to 1,101,600 transactions per day.**

CRYT transactions are based on a series of core *transaction types* that do not require any script processing or transaction input/output processing on the part of network nodes. These transaction primitives allow core support for:

- asset exchange
- alias registration
- encrypted messages
- digital goods store
- monetary system
- voting system
- phased transactions
- account control
- shuffling
- account properties
- cloud data

By leveraging these primitive transaction types, CRYTs core can be seen as an agile, base-layer protocol upon which a limitless range of services, applications, and other currencies can be built. This version of the whitepaper documents features and algorithms that are implemented in CRYT as of version 1.11.15. Future revisions will be made to reflect additional planned features and algorithm changes.

3 Core technologies

3.1 Proof of Stake

In the traditional Proof of Work model used by most cryptocurrencies, network security is provided by peers doing work. They deploy their resources (computation/processing time) to reconcile double-spending transactions, and to impose an extraordinary cost on those who would attempt to reverse transactions. Tokens are awarded to peers in exchange for work, with the frequency and amount varying with each cryptocurrency's operational parameters. This process is known as mining. The frequency of block generation, which determines each cryptocurrency's available mining reward, is generally intended to stay constant. As a result, the difficulty of the required work for earning a reward must increase as the work capacity of the network increases.

As a Proof of Work network becomes stronger, there is less incentive for an individual peer to support the network, because their potential reward is split among a greater number of peers. In search of profitability, miners keep adding resources in the form of specialized, proprietary hardware that requires significant capital investment and high ongoing energy demands. As time progresses, the network becomes more and more centralized as smaller peers (those who can do less work) drop out or combine their resources into pools.

In the Proof of Stake model used by CRYT, network security is governed by peers having a *stake* in the network. The incentives provided by this algorithm do not promote centralization in the same way that Proof of Work algorithms do, and data shows that the CRYT network has remained highly decentralized since its inception: a large number of unique accounts are contributing blocks to the network.

3.1.1 CRYT's Proof of Stake Model

CRYT uses a system where each coin in an account can be thought of as a tiny mining rig. The more tokens that are held in the account, the greater the chance that account will earn the right to generate a block. The total reward received as a result of block generation is the sum of the transaction fees located within the block. CRYT does not generate any new tokens as a result of block creation. Redistribution of CRYT takes place as a result of block generators receiving transaction fees, so the term forging (meaning in this context to create a relationship or new conditions) is used instead of mining.

Subsequent blocks are generated based on verifiable, unique, and almost-unpredictable information from the preceding block. Blocks are linked by virtue of these connections, creating a chain of blocks (and transactions) that can be traced all the way back to the genesis block.

Block generation time is targeted at 20 seconds.

The security of the blockchain is always of concern in Proof of Stake systems. The following basic principles apply to CRYTs Proof of Stake algorithm:

- A *cumulative difficulty* value is stored as a parameter in each block, and each subsequent block derives its new difficulty from the previous blocks value. In case of ambiguity, the network achieves consensus by selecting the block or chain fragment with the highest cumulative difficulty. This is covered in more detail in section 3.4.1
- To prevent account holders from moving their stake from one account to another as a means of manipulating their probability of block generation, tokens must be stationary within an account for 1,440 blocks before they can contribute to the block generation process. Tokens that meet this criterion contribute to an account's *effective balance*, and this balance is used to determine forging probability.
- To keep an attacker from generating a new chain all the way from the genesis block, peers allow chain re-organization of no more than 720 blocks behind the current block height. Any block submitted at a height lower than this threshold is rejected.
- Due to the extremely low probability of any account taking control of the blockchain by generating its own chain of blocks, transactions are deemed safe once they are encoded into a block that is 10 blocks behind the current block height.

3.1.2 Contrast with Peercoin Proof of Stake

Peercoin uses a *coin age* parameter as part of its mining probability algorithm. In that system, the longer your Peercoins have been stationary in your account (to a maximum of 90 days), the more power (coin age) they have to mint a block. The act of minting a block requires the consumption of coin age value, and the network determines consensus by selecting the chain with the largest total consumed coin age.

When Peercoin blocks are orphaned, the consumed coin age is released back to the blocks originating account. As a result, the cost to attack the Peercoin network is low, since attackers can keep attempting to generate blocks (referred to as *grinding stake*) until they succeed. Peercoin minimizes these and other risks by centrally broadcasting blockchain checkpoints several times a day, to freeze the blockchain and lock in transactions.

CRYT does not use coin age as part of its forging algorithm. An account's chance to forge a block depends only on its effective balance (which is a property of each account), the time since the last block (which is shared by all forging accounts) and the base target value (which is also shared by all accounts).

3.2 Tokens

The total supply of CRYT is 1 billion tokens, divisible to eight decimal places. All tokens were issued with the creation of the *genesis block* (the first block in the CRYT blockchain), leaving the *genesis account* with an initial negative balance of 1 billion CRYT.

The existence of anti-tokens in the genesis account has a couple of interesting side effects:

- the genesis account cannot issue transactions of any kind, since its balance is negative and it cannot pay transaction fees. As a result, the private passphrase for the genesis account is free for anyone to use.
- any tokens sent to the genesis account are effectively destroyed, since that accounts negative balance will cancel them out.

The choice of the word *tokens* is intentional due to CRYTs intention to be used as a base protocol that provides numerous other functions. CRYTs most basic function is one of a traditional payment system, but it was designed to do far more.

3.3 Network Nodes

A *node* on the CRYT network is any device that is contributing transaction or block data to the network. Any device running the CRYT software is seen as a node. Nodes are sometimes referred to as "Peers".

Nodes can be subdivided into two types: *hallmarked* and *normal*. A hallmarked node is simply a node that is tagged with an encrypted token derived from an account private key; this token can be decoded to reveal a specific CRYT account address and balance that are associated with a node. The act of placing a hallmark on a node adds a level of accountability and trust, so hallmarked nodes are more trusted than non-hallmarked nodes on the network. The larger the balance of an account tied to a hallmarked node, the more trust is given to that node. While an attacker might wish to hallmark a node in order to gain trustworthiness within the network and then use that trust for malicious purposes; the barrier to entry (cost of CRYT required to build adequate trust) discourages such abuse.

Each node on the CRYT network has the ability to process and broadcast both transactions and block information. Blocks are validated as they are received from other nodes, and in cases where block validation fails, nodes may be blacklisted temporarily to prevent the propagation of invalid block data.

Each node features a built-in DDOS (Distributed Denial of Services) defense mechanism which restricts the number of network requests from any other node to 30 per second.

3.4 Blocks

As in other crypto-currencies, the ledger of CRYT transactions is built and stored in a linked series of blocks, known as a blockchain. This ledger provides a permanent record of transactions that have taken place, and also establishes the order in which transactions have occurred. A copy of the blockchain is kept on every node in the CRYT network, and every account that is *unlocked* on a node (by supplying the account private key) has the ability to generate blocks, as long as at least one incoming transaction to the account has been confirmed 1440 times. Any account that meets these criteria is referred to as an *active account*.

In CRYT, each block contains up to 255 transactions, all prefaced by a block header that contains identifying parameters. Each transaction in a block is represented by common transaction data, specific transaction types also include transaction attachment, and certain transactions may include one or more additional appendices. The maximum block size is 42KB. All blocks contain the following parameters:

- A block version, block height value, and block identifier
- A block timestamp, expressed in seconds since the genesis block
- The ID of the account that generated the block, as well as that accounts public key
- The ID and hash of the previous block The number of transactions stored in the block
- The total amount of CRYT represented by transactions and fees in the block
- Transaction data for all transactions included in the block, including their transaction IDs
- The payload length of the block, and the hash value of the block payload
- The block's generation signature
- A signature for the entire block
- The base target value and cumulative difficulty for the block

3.4.1 Block Creation (Forging)

Three values are key to determining which account is eligible to generate a block, which account earns the right to generate a block, and which block is taken to be the authoritative one in times of conflict: *base target value*, *target value* and *cumulative difficulty*.

3.4.1.1 Base Target Value

In order to win the right to forge (generate) a block, all active CRYT accounts compete by attempting to generate a hash value that is lower than a given *base target value*. This base target value varies from block to block, and is derived from the previous block base target multiplied by the amount of time that was required to generate that block using a formula that ensures 20 seconds average block time.

The calculation is based on the following constants:

- MAXRATIO=27 - max ratio by which the target is decreased when block time is larger than 20 seconds.
- MINRATIO=13 - min ratio by which the target is increased when block time is smaller than 20 seconds.
- GAMMA=0.64

And the following variables:

- S - average block time for the last 3 blocks
- Tp - previous base target
- Tb - calculated base target

The base target is calculated as follows:

- If S>20

```
Tb = (Tp * Min(S, MAXRATIO)) / 20
```

- Else

```
Tb = Tp - Tp * GAMMA * (20 - Max(S, MINRATIO)) / 20;
```

The idea is to make target adjustments gradual using the MIN and MAX ratio constants and increase the target, thus reducing block times, at a faster rate than decreasing the target, using the GAMMA constant, since the block time is bounded by 0 from below but can be infinitely large.

3.4.1.2 Target Value

Each account calculates its own target value, based on its current effective stake. This value is:

$$T = T_b \times S \times B_e$$

where:

T is the new target value

T_b is the base target value

S is the time since the last block, in seconds

B_e is the effective balance of the account

As can be seen from the formula, the target value grows with each second that passes since the timestamp of the previous block. The maximum target value is $1.53722867 \times 10^{17}$ and the minimum target value is one half of the previous blocks base target value.

This target value and the base target value are the same for all accounts attempting to forge on top of a specific block. The only account-specific parameter is the effective balance parameter.

3.4.1.3 Cumulative Difficulty

The cumulative difficulty value is derived from the base target value, using the formula:

$$D_{cb} = D_{pb} + \frac{2^{64}}{T_b}$$

where:

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

3.4.1.4 The Forging Algorithm

Each block on the chain has a *generation signature* parameter. To participate in the block forging process, an active account digitally signs the generation signature of the previous block with its own public key. This creates a 64-byte signature, which is then hashed using SHA256. The first 8 bytes of the resulting hash are converted to a number, referred to as the account *hit*.

The hit is compared to the current target value. If the computed hit is lower than the target, then the next block can be generated. As noted in the target value formula, the target value increases with each passing second. Even if there are only a few active accounts on the network, one of them will eventually generate a block because the target value will become very large. Therefore, you can calculate the time it will take any account to forge a block by comparing the account hit value to the target value.

The last point is significant. Since any node can query the effective balance for any active account, it is possible to iterate through all active accounts in order to determine their individual hit value. This means it is possible to predict, with reasonable accuracy, which account will next win the right to forge a block. A *balance shifting attack* cannot be mounted by moving stake to an account that will generate the next block, since CRYT stake must be stationary for 1440 blocks before it can contribute to forging (via the effective balance value). Interestingly, the new base target value for the next block cannot be reasonably predicted, so the nearly-deterministic process of determining who will forge the next block becomes increasingly stochastic as attempts are made to predict future blocks. This feature of the CRYT forging algorithm helps form the basis for the development and implementation of the Transparent Forging algorithm.

When an active account wins the right to generate a block, it bundles up to 255 available, unconfirmed transactions into a new block, and populates the block with all of its required parameters. This block is then broadcast to the network as a candidate for the blockchain.

The payload value, generating account, and all of the signatures on each block can be verified by all network nodes who receive it. In a situation where multiple blocks are generated, nodes will select the block with the highest cumulative difficulty value as the authoritative block. As block data is shared between peers, forks (non-authoritative chain fragments) are detected and dismantled by examining the chains cumulative difficulty values stored in each fork.

A node which receive a valid block representing a chain with larger cumulative difficulty than it's own, will determine the highest common block between it's own chain and the chain represented by the new block, then remove it's own blocks from the chain down to the common block and undo any side effects of these blocks then build it's own chain based on blocks received from other nodes.

3.4.1.5 Balance leasing

Since the ability for an account to forge is based on the effective balance parameter, it is possible to loan forging power from one account to another without giving up control of the tokens associated with the account. Using a leaseBalance transaction, an account owner may temporarily reduce an accounts effective balance to zero, adding it to the effective balance of another account. The targeted account forging power is increased for a certain number of blocks specified by the original account owner, after which the effective balance is returned to the original account.

Leasing is advised for large stake holders since the lessor account, which leased its forging power, does not need to reveal its passphrase in order to participate in forging new blocks. Only the lessee account need to reveal its passphrase and this account can poses much smaller balance so that in case its passphrase is stolen the lose is minimal.

Leasing balance does not affect the functionality of the lessor account except its ability to forge. Balance changes to the lessor account affects the forging power of the lessee account after 1440 blocks.

3.4.2 Accounts

CRYT implements a *brain wallet* as part of its design: all accounts are stored on the network, with *private keys* for each possible account address directly derived from each accounts *passphrase* using a combination of SHA256 and Curve25519 operations.

Each account is represented by a 64-bit number, and this number is expressed as an *account address* using a Reed-Solomon error-correcting notation that allows for *detection* of up to four errors in an account address, or *correction* of up to two errors. This practically eliminates the risk that a typo in account address would result in lose of funds. Account addresses are always prefaced by an *CRYT-* prefix, making CRYT account addresses easily recognizable and distinguishable from address formats used by other blockchains.

The Reed-Solomon-encoded account address associated with a secret passphrase is generated as follows:

1. The secret passphrase is hashed with SHA256 to derive the accounts *private key*.
2. The private key is encrypted with Curve25519 to derive the accounts *public key*.
3. The public key is hashed with SHA256 to derive the *account ID*.
4. The first 64 bits of the account ID are the *visible account number*.
5. Reed-Solomon encoding of the visible account number, prefixed with CRYT-, generates the *account address*.

When an account is accessed by a secret passphrase for the very first time, it is not secured by a public key. When the first outgoing transaction from an account is made, the 256-bit public key derived from the passphrase is stored on the blockchain, and this secures the account. The address space for public keys (2^{256}) is larger than the address space for account numbers (2^{64}), so there is no one-to-one mapping of passphrases to account numbers and collisions are possible. These collisions are detected and prevented in the following way: once a specific passphrase is used to access an account, and that account is secured by a 256-bit public key, no other public-private key pair is permitted to access that account number.

3.4.2.1 Account Balance Properties

For each CRYT account, several different types of balances are available. Each type serves a different purpose, and many of these values are checked as part of transaction validation and processing.

- The *effective balance* of an account is used as the basis for an account's forging calculations. An account's effective balance consists of all tokens that have been stationary in that account for 1440 blocks. In addition, the Account Leasing feature allows an account's effective balance to be assigned to another account for a temporary period. The account effective balance is calculated from the *confirmed balance* by reducing all balance additions during the last 1440 blocks.
- The guaranteed balance of an account consists of all tokens that have been stationary in an account for 1440 blocks. Unlike the effective balance, this balance cannot be assigned to any other account.
- The *confirmed balance* of an account accounts for all transactions that have had at least one confirmation.
- The *unconfirmed balance* of an account is the one that is displayed in CRYT clients. It represents the confirmed balance of an account, minus the tokens involved in unconfirmed, sent transactions or locked by specific transaction types such as CurrencyReserveIncrease and Shuffling transactions or locked by phased transactions not applied or cancelled yet.
- The *forged balance* of an account shows the total amount of CRYT that have been earned as a result of successfully forging blocks.
- Confirmed and unconfirmed asset quantities and currency units are also tracked by each account holdings.

3.4.3 Transactions

Transactions are the only means CRYT accounts have of altering their state or balance. Each transaction performs only one function, the record of which is permanently stored on the network once that transaction has been included in a block.

3.4.3.1 Transaction Fees

Transaction fees are the primary mechanism through which CRYT are recirculated back into the network. Every transaction requires a minimum fee. When an CRYT account forges a block, all of the transaction fees included in that block are awarded to the forging account as a reward. Unlike with other blockchains, minimum transaction fees are enforced by the blockchain therefore transactions which does not specify a fee larger than the minimal fee for this transaction type won't be accepted by nodes.

3.4.3.2 Transaction Confirmations

All CRYT transactions are considered *unconfirmed* until they are included in a valid network block. Newly-created blocks are distributed to the network by the node (and associated account) that creates them, and a transaction that is included in a block is considered as having received one confirmation. As subsequent blocks are added to the existing blockchain, each additional block adds one more confirmation to the number of confirmations for a transaction.

If a transaction is not included in a block before its deadline, it expires and is removed from the transaction pool.

3.4.3.3 Transaction Deadlines

Every transaction contains a deadline parameter, set to a number of minutes from the time the transaction is submitted to the network. The default deadline is 1440 minutes (24 hours). A transaction that has been broadcast to the network but has not been included in a block yet is referred to as an *unconfirmed transaction*.

If a transaction has not been included in a block before the transaction deadline expires, the transaction is removed from the network.

Transactions may be left unconfirmed until their deadline expire, because they are permanently invalid or malformed, or because they do not meet certain temporary conditions such as sufficient balances, or because blocks are being filled with transactions that have offered to pay higher transaction fees.

3.4.3.4 Transaction Types

Categorizing CRYT transactions into types and subtypes allows for modular growth and development of the CRYT protocol without creating dependencies on other base functions. As features are added to the CRYT core, new transaction types and subtypes can be added to support them.

Multiple transaction types and associated subtypes are supported by CRYT. Each type dictates a given transactions required and optional parameters, as well as its processing method. A complete list of all transaction types and sub types is out of the scope of this document.

3.4.3.5 Transaction Creation and Processing

The details of creating and processing an CRYT transaction are as follows:

1. The sender specifies parameters for the transaction. Types of transactions vary, and the desired type is specified at transaction creation, but several parameters must be specified for all transactions:
   - private key for the sending account
   - specified fee for the transaction
   - deadline for the transaction
   - an optional referenced transaction
2. All values for the transaction inputs are checked. For example, mandatory parameters must be specified; fees cannot be less than the minimum fee for this transaction type; 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:
   1. The public key for the generating account is computed using the supplied secret passphrase
   2. 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 *unconfirmed balance* must not be lower than the transaction amount plus the transaction fee
4. If the sending account has sufficient funds for the transaction:
   1. 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 object
   2. The transaction is signed using the sending account's private key
   3. The encrypted transaction data is placed within a message instructing network peers to process the transaction
   4. The transaction is broadcast to all peers on the network
   5. The server responds with a result code:
      - the transaction ID, if the transaction creation was successful
      - an error code and error message if any of the parameter checks fail.

3.5 Cryptographic Foundations

Key exchange in CRYT is based on the Curve25519 algorithm, which generates a shared secret key using a fast, efficient, high-security elliptic-curve Diffie-Hellman function. The algorithm was first demonstrated by Daniel J. Bernstein in 2006. CRYTs Java-based implementations were reviewed by DoctorEvil in March, 2014.

Message signing in CRYT is implemented using the Elliptic-Curve Korean Certificate-based Digital Signature Algorithm (EC-KCDSA), specified as part of IEEE P1363a by the KCDSA Task Force team in 1998.

Both algorithms were chosen for their balance of speed and security for a key size of only 32 bytes.

3.5.1 Encryption Algorithm

When Alice sends an encrypted plaintext to Bob, she:

1. Calculates a shared secret:
   - shared_secret = Curve25519(Alice_private_key, Bob_public_key)
2. Calculates N seeds:
   - $seed_n$ = SHA256($seed_{n-1}$), where $seed_0$ = SHA256(shared_secret)
3. Calculates N keys:
   - key_n = SHA256(Inv($seed_n$)), where Inv(X) is the inversion of all bits of X
4. Encrypts the plaintext:
   - ciphertext[n] = plaintext[n] XOR key_n

Upon receipt Bob decrypts the ciphertext:

1. Calculates a shared secret:
   - shared_secret = Curve25519(Bob_private_key, Alice_public_key)
2. Calculates N seeds (this is identical to Alices step):
   - $seed_n$ = SHA256($seed_{n-1}$), where $seed_0$ = SHA256(shared_secret)
3. Calculates N keys (this is identical to Alices step):
   - key_n = SHA256(Inv($seed_n$)), where Inv(X) is the inversion of all bits of X
4. Decrypts the ciphertext:
   - plaintext[n] = ciphertext[n] XOR key_n

Note: If someone guesses part of the plaintext, he can decode some part of subsequent messages between Alice and Bob if they use the same key pairs. As a result, it's advised to generate a new pair of private/public keys for each communication.

4 Core Features

4.1 Advanced JavaScript client

A second-generation, user-friendly client application is built into the CRYT core software distribution, and can be accessed through a local web browser. The client provides full support for all core CRYT features, implemented such that users private keys are never exposed to the network. It also includes an advanced administrative interface and built-in javadoc documentation for CRYTs low-level Applications Programming Interface.

4.2 Agile architecture

First-generation cryptocurrencies were primarily designed as payment systems. CRYT recognizes that decentralized blockchains can enable a broad range of applications and services, but is not prescriptive about what those services should be or how they should be built. By design, CRYT strips away unnecessary complexity in its core, leaving only the most successful components of its predecessors intact. As a result, CRYT functions like a low-level, foundational protocol: it defines the interfaces and operations required to operate a lightweight blockchain, a decentralized communication system, and a rapid transaction processing framework, allowing higher-order components to build on those features.

Transactions in CRYT make simple adjustments to account balances instead of tracing sets of input or output credits. In addition, the core software does not support any form of scripting language. By providing a set of basic, flexible transaction types that can quickly and easily be processed, CRYT creates a foundation that does not limit the ways in which those transaction types can be used, and does not create significant overhead for using them. This flexibility is further amplified by CRYTs low resource and energy requirements, and its highly readable, highly organized object-oriented source code.

4.3 Basic Payments

The most fundamental feature of any cryptocurrency is the ability to transmit tokens from one account to another. This is CRYTs most fundamental transaction type, and it allows for basic payment functionality.

4.4 Alias System

The CRYT Alias System allows any string of text to be permanently associated with a specific CRYT account. Since its inception, a convention for the format of these strings, using JSON notation, has been formalized. As a result, an alias can currently be human-friendly text alias for an account address or a Uniform Resource Identifier (URI).

The ability to store any URI on the CRYT blockchain enables the creation of any number of decentralized services that rely on small, persistent strings of text, such as a distributed Domain Name Server (DNS) system.

4.5 Arbitrary Messages

Arbitrary strings of data up to 1000 bytes in length can be stored on the CRYT blockchain using the Arbitrary Messages feature, and these strings may optionally be AES-encrypted. These messages are intended to be removable, in the future, when blockchain size needs to be reduced; nonetheless, they form a critical building block for a number of next-generation features.

At the basic level, the system can be used to transmit human-readable messages between accounts, creating a decentralized chat system. However, advanced applications can use this feature to store structured data, such as JSON objects, that can be used to trigger or facilitate services built on top of CRYT.

4.6 Asset Exchange

An entire class of CRYT transactions is used to implement a fully-decentralized and automated asset exchange that operates on the CRYT blockchain. Using the colored coins concept, CRYT assets may be issued and tracked on the CRYT ecosystem, supported by transactions and processing that allow for asset transfer, bid and ask order placement, and automatic order matching.

By combining the features of the CRYT Asset Exchange with other features such as the Arbitrary Messaging System, value-added services can be created. Most notably, another feature of the CRYTServices layer is a system for the automated calculation and disbursement of dividends based on the performance of existing CRYT assets.

4.7 Digital Goods Store

The CRYT Digital Goods store gives account owners the ability to list assets for sale in an open, decentralized market place. Goods can be purchased, discounted, delivered, refunded, and transferred, using a dedicated class of transaction types that manage and secure store listings on the decentralized blockchain.

4.8 Device Portability

Due to its cross-platform, Java-based roots, its Proof of Stake hashing and its future ability to reduce the size of the block chain, CRYT is extremely well suited for use on small, low-power, low-resource devices. Android and iPhone applications are currently in development, and the CRYT software has been ported to low-powered ARM devices such as the RaspberryPi and CubieTruck platforms.

The ability to implement CRYT on low-powered, always-connected devices such as smartphones allows us to envision a scenario where the majority of the CRYT network is supported on mobile devices. The low cost and resource consumption of these devices significantly reduce network costs in comparison with traditional Proof of Work cryptocurrencies.

5 Concerns

5.1 Proof of Stake Attacks

5.1.1 Nothing at Stake

In a nothing at stake attack, forgers attempt to build blocks on top of every fork they see because doing so costs them almost nothing, and because ignoring any fork may mean losing out on the block rewards that would be earned if that fork were to become the chain with the largest cumulative difficulty.

While this attack is theoretically possible, it is currently not practical. The CRYT network does not experience long blockchain forks, and the low block reward does not provide a strong profit incentive; further, compromising network security and trust for the sake of such small gains would make any victory pyrrhic.

As part of CRYTs development roadmap, a feature called Economic Clustering will provide further protection against attacks of this nature by forcing transactions to include hashes of previous blocks, and by grouping nodes into clusters that can detect unusual behavior on the network and impose penalties (in the form of temporary loss of the ability to forge).

5.1.2 History Attack

In a history attack, someone acquires a large number of tokens, sells them, and then attempts 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. Extreme forms of this attack involve obtaining the private keys from old accounts and using them to build a successful chain right from the genesis block.

In CRYT, the basic history attack generally fails because all stake must be stationary for 1440 blocks before it can be used for forging; moreover, the effective balance of the account that generates each block is verified as part of block validation. The extreme form of this attack generally fails because the CRYT blockchain cannot be re-organized more than 720 blocks behind the current block height. This limits the time frame in which a bad actor could mount this form of attack.

5.2 Distribution

Because blocks may only be generated based on existing stake, at least some of the token supply must be available when a Proof of Stake network is bootstrapped. As a result, CRYT issued and distributed its full supply of tokens with the creation of the genesis block.

The initial supply of CRYT was distributed to:

- 10% to NXT Holders, snapshot @ NXT block 2640000. (100M CRYT)
- 63% to old CRYT Holders. (630M CRYT)
- 10% to CRYT Faucet. (100M CRYT)
- 17% to Team, for Development, Campaign, Advertising. (170M CRYT)

It will never be possible for CRYTs proponents to dispel the distribution concerns raised by the wider community. Relative to the levels of profit achieved by early investors in IBM, Apple, Google, Facebook, and Bitcoin, the amount of inequality present in the CRYT blockchain is not out of line.

5.3 Transaction Fees

As the value of CRYT increases, the cost of minimum transactions fees, expressed in fiat terms, also increases. Plans are underway to reduce the minimum fee, scaled according to transaction byte-size, in order for micro-transactions to be practical.

6 Additional CRYT-related Papers and Resources

- Official Website
  - https://crytrex.com/cryt

7 Bibliography

Bitcoin: a Peer-to-Peer Electronic Cash System. (n.d.). Retrieved July 06, 2014, from https://bitcoin.org/bitcoin.pdf

Bitcoin Is Broken. (n.d.). Retrieved July 06, 2014, from http://hackingdistributed.com/2013/11/04/bitcoin-is-broken/

Bitcoin Miners Ditch Ghash.io Pool Over Fears of 51% Attack. (n.d.). Retrieved July 06, 2014, from http://www.coindesk.com/bitcoin-miners-ditch-ghash-io-pool-51-attack/

Bitcoin needs to scale by a factor of 1000 to compete with Visa. Heres how to do it. (n.d.). Retrieved July 06, 2014, from http://www.washingtonpost.com/blogs/the-switch/wp/2013/11/12/bitcoin-needs-to-scale-by-a-factor-of-1000-to-compete-with-visa-heres-how-to-do-it/

Bitcoin security guarantee shattered by anonymous miner with 51% network power. (n.d.). Retrieved July 06, 2014, from http://arstechnica.com/security/2014/06/bitcoin-security-guarantee-shattered-by-anonymous-miner-with-51-network-power/

Cohen, R. (2013, November 28). Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers, Combined! Retrieved July 06, 2014, from http://www.forbes.com/sites/reuvencohen/2013/11/28/global-bitcoin-computing-power-now-256-times-faster-than-top-500-supercomputers-combined

Crypto Review of Curve25519.java & Crypto.java. (n.d.). Retrieved July 06, 2014, from https://gist.github.com/doctorevil/9521116

Eyal, I., & Gun Sirer, E. (2013). Majority is not Enough: Bitcoin Mining is Vulnerable. Unpublished manuscript. Retrieved July 06, 2014, from http://arxiv.org/pdf/1311.0243v5.pdf

Learn Cryptography 51% Attack. (n.d.). Retrieved July 06, 2014, from http://learncryptography.com/51-attack/

Losing to win. (2014, June 23). Retrieved July 03, 2014, from http://www.economist.com/blogs/schumpeter/2014/06/bitcoin

Peercoin. (n.d.). Retrieved July 06, 2014, from http://www.peercoin.net/whitepaper

Qin, W., & Zhou, N. (2010, 12). New concurrent digital signature scheme based on the computational Diffie-Hellman problem. The Journal of China Universities of Posts and Telecommunications, 17(6), 89-100. doi: 10.1016/S1005-8885(09)60530-6

The Well Deserved Fortune of Satoshi Nakamoto, Bitcoin creator, Visionary and Genius. (n.d.). Retrieved July 06, 2014, from http://bitslog.wordpress.com/2013/04/17/the-well-deserved-fortune-of-satoshi-nakamoto/

Yung, M., Dodis, Y., Kiayias, A., Malkin, T., & Bernstein, D. J. (2006). Curve25519: New Diffie-Hellman Speed Records. Public Key Cryptography, 2006, 207-228. doi: 10.1007/11745853_14

8 Appendix: Bitcoin Problems Addressed by CRYT

CRYT was created as a cryptocurrency 2.0 response to Bitcoin. CRYT adopts features that have proved to work well in Bitcoin, and addresses aspects that are cause for concern.
This appendix addresses issues with the Bitcoin protocol and network that are mitigated by CRYT technology.

8.1 Blockchain Size

The Bitcoin blockchain is the complete sequential collection of generated data blocks containing the electronic ledger book for all Bitcoin transactions occurring since its launch in January 2009. Four years later in January 2013, the size of the Bitcoin blockchain stood at 4 gigabytes (GB) about the amount of data required to store a two hour movie on a DVD disk. Eighteen months later, in July 2014, the size of the Bitcoin blockchain had swelled by almost a factor of five to 19 gigabytes (GB).
The Bitcoin blockchain is undergoing exponential growth and modifications to the original Bitcoin protocol will be required to deal with it.

8.1.1 CRYT Solutions

CRYT block size is currently capped at 32KB.
In the future, CRYT will implement a Blockchain Pruning feature that will reduce blockchain size by selectively removing information on permanent blocks, and by deleting other non-persistent data, such as Arbitrary Messages.

8.2 Transactions per Day

In late 2013, the number of transactions being processed on the Bitcoin network was peaking at 70,000 per day, which is about 0.8 transactions per second (tps). The current Bitcoin standard block size of one megabyte, generated every ten minutes (on average) by full node clients, limits the maximum capacity of the current Bitcoin network to a about 7 tps. Compare this with the VISA network's capacity to handle 10,000 tps and you will see that Bitcoin cannot compete as it exists today.

Increasing public use of the Bitcoin system will cause Bitcoin to soon hit its transaction-per-day limit and halt further growth. To forestall this, Bitcoin software developers are working on the creation of thin clients[27] that employ simplified payment verification (SPV)[28]. To handle greater throughput in the same 10-minute-average time, SPV thin clients will not perform a full security check on the larger blocks they process. They will instead examine multiple hashed blockchains from competing miners and assume that the blockchain version generated by the majority of miners is correct. In the words of Bitcoins Mike Hearn, Instead of verifying the entire contents, [SPV] just trusts that the majority of miners are honest.… As long as the majority is honest, [SPV] works… [However],the full node does give you better security. If you're running an online shop for example, it makes sense to run a full node.

8.2.1 CRYT Solutions

In its current state, the CRYT network can process up to 1,000,000 transactions per day more than 30 times Bitcoins current peak values. The planned implementation of Transparent Forging will allow for near instant transaction processing, drastically increasing this limit.

8.3 Transaction Confirmation Time

Transaction confirmation times for Bitcoin ranged from 5 to 10 minutes for most of 2013. After the late 2013 announcement that Chinese banks would not be allowed to process Bitcoins, the average Bitcoin transaction time significantly increased to 8 to 13 minutes, with occasional peaks of 19 minutes. Confirmation times have since resettled in the 8 to 10 minute range. Nonetheless, since multiple verifications are required to finalize a Bitcoin transaction (six confirmations is generally preferred), one hour can easily pass before a sale of assets paid for by Bitcoin is complete.

8.3.1 CRYT Solutions

The average block generation time for CRYT has historically been shown to be about 20 seconds, putting the average transaction processing time at the same value.
Transactions are deemed safe after ten confirmations, meaning that transactions are permanent in less than 4 minutes.

The implementation of Transparent Forging will allow for nearly-instant transactions, which will further reduce this time.

8.4 Centralization Concerns

The increasing difficulty and combined network hashrate for Bitcoin has created a high barrier to entry for newcomers, and diminished returns for existing mining rigs. The block reward incentive employed by Bitcoin has driven the creation of large, single-owner installations of dedicated mining hardware, as well as the reliance on a small set of large mining pools. This has resulted in a centralization effect, where large amounts of mining power are concentrated in the control of a decreasing number of people. Not only does this create the kind of power structure that Bitcoin was designed to circumvent, but it also presents the real possibility that a single mining operation or pool could amass 51% of the network's total mining power and execute a 51% attack. Attacks requiring as little as 25% of total network hashing power also exist.

In early January, 2014, GHash.io began voluntarily decreasing its own mining power because it was approaching the 51% level. After a few days, the pool's mining power was reduced to 34% of the total network power, but the rate immediately began to increase again, and once more reached dangerous levels in June 2014.

8.4.1 CRYT Solutions

The incentives provided by CRYTs Proof of Stake algorithm provide a low Return on Investment of approximately 0.1%. Since no new coins are generated with each block, there is no additional mining reward that incentivizes combining efforts to generate blocks.
Data shows that the CRYT network has remained highly decentralized since its inception: a large (and growing) number of unique accounts are contributing blocks to the network.

8.5 Proof of Work's Resource Costs

Confirming transactions for existing Bitcoins, and creating new Bitcoins to go into circulation, requires enormous background computing power that must operate continuously. This computing power is provided by so-called mining rigs operated by miners. Bitcoin miners compete among themselves to add the next transaction block to the overall Bitcoin blockchain. This is done by hashing - bundling all Bitcoin transactions occurring over the past ten minutes and trying to encrypt them into a block of data that also coincidentally has a certain number of consecutive zeros in it. Most trial blocks generated by a miner's hashing effort don't have this target number of zeros, so they make a slight change and try again. A billion attempts to find this winning block is called a gigahash, with a mining rig being rated by how many gigahashes it can perform in a second, denoted by *GH/sec*. A winning miner who is first to generate the next needle-in-a-haystack, cryptographically-correct Bitcoin block currently receives a reward of 25 newly-mined Bitcoins - a reward worth, at the time of this writing, around $15,750USD. This competition among miners, with its hefty reward, repeats itself over and over and over every ten minutes or so. By early 2014 over 3500 bitcoins per day are generated, worth around $2.2 million US dollars per day.

With so much money at stake, miners have supported a blistering arms race in mining rig technology to better their odds of winning. Originally Bitcoins were mined using the central processing unit (CPU) of a typical desktop computer. Then the specialized graphics processing unit (GPU) chips in high-end video cards were used to increase speeds. Field programmable gate array (FPGA) chips were pressed into service next, followed by mining rigs specialized application specific integrated circuits (ASIC) chips. ASIC technology is the top of the line for Bitcoin miners, but the arms race continues with various generations of ASIC chips now coming into service. The current generation of ASIC chips are the so-called 28nm units, based on the size of their microscopic transistors in nanometers. These are due to be replaced by 20nm ASIC units by late-2014. An example of an upcoming state-of-the-art mining rig would be a Butterfly Labs Monarch 28nm ASIC card, which is to provide 600GH/sec for an electricity consumption of 350 watts and a price of $2200USD[38].

The mining rig infrastructure currently in place to support ongoing Bitcoin operations is astounding. Bitcoin ASICs are like autistic savants - they are able to do only the Bitcoin block calculation and nothing more, but they can do that one calculation at supercomputer speeds. In November 2013, Forbes magazine ran an article entitled, Global Bitcoin Computing Power Now 256 Times Faster Than Top 500 Supercomputers, Combined!. In mid January 2014, statistics maintained at blockchain.info showed that ongoing support of Bitcoin operations required a continuous hash rate of around 18 million GH/sec. During the course of one day, that much hashing power produced 1.5 trillion trial blocks that were generated and rejected by Bitcoin miners looking for one the magic 144 blocks that would net them $2.2 million USD. Almost all Bitcoin computations do not go towards curing cancer by modeling DNA or to searching for radio signals from E.T.; instead, they are totally wasted computations.

The power and cost involved in this wasteful background mining support of Bitcoin is enormous. If all Bitcoin mining rigs had Monarch levels of capability as described above - which they will not, until they are upgraded - they would represent a pool of 30,000 machines costing over $63 million USD and consuming over 10 megawatts of continuous power while running up an electricity bill of over $3.5 million USD per day[39]. The real numbers are significantly higher for the current, less-efficient mining rig pool of machines actually supporting Bitcoin today. And these numbers are currently headed upward in an exponential growth curve as Bitcoin marches from its current one transaction per second to its current maximum of seven transactions per second.

8.5.1 CRYT Solutions

Analysis of the cost and energy efficiency of the CRYT network shows that the entire CRYT ecosystem can be maintained for about $60,000USD per year, which is currently almost 2,200 times less expensive than the cost of running the Bitcoin network.

8.6 Proof of Work's Resource Costs Pertaining to Coinholders

In addition to massive electrical costs, there is a hidden fee for simply holding Bitcoins. For each block found, the entity that generates the block receives a stipend. At the time of writing, this stipend is 12.5 BTC, producing 10% inflation in the total Bitcoin supply this year alone. For each $1000USD worth of Bitcoin someone owns, that person is paying $100USD per Bitcoin this year to pay miners for keeping the network secure.

8.6.1 CRYT Solution

Since the complete supply of CRYTs 1 billion coins was created with the genesis block, there is no inflation in CRYT. Deflationary pressures are likely to affect CRYT in the future, and a planned feature called Antideflation will address that problem.