Prizm Whitepaper

Revision - March, 2017

This document describes the original concept of Prizm.

Bitcoin is the world's first decentralized digital currency, allowing the easy storage and transfer of cryptographic tokens, using a peer-to-peer (P2P) network to carry information, hashing as a synchronization signal to prevent double-spending, and a powerful scripting system to determine ownership of the tokens.Bitcoins are fungible, acting as a neutral medium of exchange. Bitcoins can then have special properties supported by either an issuing agent or by public agreement, and have value independent of the face value of the underlying bitcoins. Prizm (PZM) is the new generation cryptographic currency, fully decentralized peer-to-peer electronic cash system, that really works without the need for a third party.
It can 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; and be able to run on any of devices.The key advantage of Prizm is the unique ParaMining technology is not found in any of existing cryptocurrencies. But more on that later.

Contents

1 Introduction

Prizm is a 100% proof-of-stake (PoS) cryptocurrency, based on Next core, constructed in open-source Java. Prizm's 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 10 million 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 60 seconds, on average, by accounts that are unlocked on network nodes. PZM 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 Next's current architecture and block size cap allows for the processing of up to 367,200 transactions per day. Prizm includes the implementation of a Transparent Forging feature which will allow a transaction processing capacity increase of two orders of magnitude using a deterministic block generation algorithm, coupled with additional network security mechanisms.

2 Core technologies

2.1 Proof of Stake

In the traditional Proof of Work model used by most cryptocurrencies, net-work 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".

Bitcoin’s creator, Satoshi Nakamoto, intended for the bitcoin network to be fully decentralized, but nobody could have predicted that the incentives provided by Proof of Work systems would result in the centralization of the mining process. This leads to possible vulnerabilities. The GHash.io bitcoin pool has reached 51% of the bitcoin mining power in the past, and the top five bitcoin mining pools make up 70% of the Bitcoin network’s hashing power . The concept of decentralization is at risk of being completely lost.

In the Proof of Stake model used by Prizm, 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 Prizm network has remained highly decentralized since its inception: a large (and growing) number of unique accounts are contributing blocks to the network, and the top five accounts have generated 35% of the total number of blocks.

2.1.1 Prizm’s Proof of Stake Model

Prizm uses a system where each "token" 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. PZM does not generate any new tokens as a result of block creation. Redistribution of PZM 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 59 seconds, but variations in probabilities have resulted in an average block generation time of 80 seconds, with occasionally very long block intervals.

The security of the blockchain is always of concern in Proof of Stake systems.

The basic principles of Prizm's 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 block's 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 2.4.1 on page 5.

- 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, the network only allows chain re-organization 720 blocks behind the current block height. Any block submitted at a height lower than this threshold is rejected. This moving threshold may be viewed as PZM's only fixed checkpoint.

- 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.

2.2 Tokens

The initial issue of PZM is 10 million. Tokens were issued with the creation of the *genesis block* (the first block in the PZM blockchain). Pre-mining is implemented in all countries of the world at nominal value, in limited batches, to achieve Prizm starting decentralization.

The total issure of PZM will be 6 billion tokens.

Genesis account generates tokens via Paramining signals (the signal to send tokens to a definite wallet).

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

- Any tokens sent to the genesis account are effectively destroyed, since that account's negative balance will cancel them out.
- Prizm's most basic function is one of a traditional payment system, but it was designed to do far more.

2.3 Network Nodes

A node on the Prizm network is any device that is contributing transaction or block data to the network. Any device running the Prizm software is seen as a node.

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's private key; this token can be decoded to reveal a specific PZM 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 PZM required to build adequate trust) discourages such abuse.

Each node on the Prizm 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 built-in DDOS (Distributed Denial of Services) defence mechanisms which restrict the number of network requests from any peer to 30 per second.

2.4 Blocks

As in other cryptocurrencies, the ledger of PZM 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 Prizm network, and every account that is *unlocked* on a node (by supplying that account's 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 Prizm, each block contains up to 255 transactions, all prefaced by a 192-byte header that contains identifying parameters. Each transaction in a block is represented by a maximum of 160 bytes, and the maximum block size is 32KB.

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 account's public key
- The ID and hash of the previous block
- The number of transactions stored in the block
- The total amount of PZM 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

2.4.1 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.*

Base Target Value

In order to win the right to forge (generate) a block, all active Prizm 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's base target value multiplied by the amount of time that was required to generate that block.

Target Value

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

$$T = Tb \times S \times Be$$

where:

T is the new target value

Tb is the base target value

S is the time since the last block, in seconds

Be 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 x 1017 and the minimum target value is one half of the previous block's 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.

Cumulative Difficulty

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

$$Dcb = Dpb + 264 / Tb$$

where:

Dcb is the difficulty of the current block

Dpb is the difficulty of the previous block

Tb is the base target value for the current block

The Forging Algorithm

Each block on the chain has a generation signature parameter. To participate in the block forging process, an active account cryptographically 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 gives a number, referred to as the account's 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. The corollary of this is that you can estimate the time that will be required for any account to forge a block by comparing that account's 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 shuffling attack could be mounted by moving stake to an account that will generate the next block, which is another reason why a PZM 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 PZM 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.

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 transaction of the "account control" type, an account owner may temporarily reduce an account's effective balance to zero, adding it to the effective balance of another account. The targeted account's forging power is increased until the end of a time period specified by the original account owner, after which the effective balance is returned to the original account.

Accounts with leased forging power generate blocks more often and earn more transaction fees, but those fees are not automatically returned to lease accounts. With a bit of coding, however, this system allows for the creation of nearly-trustless forging pools that can make payouts to participants.

2.4.2 Accounts

Prizm 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 account's 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 format was implemented in response to concerns that a mistyped account address could result in tokens, aliases, or assets being irreversibly transferred to erroneous destination accounts. Account addresses are always prefaced by “PRIZM-”, making Next account addresses easily recognizable and distinguishable from address formats used by other cryptocurrencies.

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 account’s private key.

2. The private key is encrypted with Curve25519 to derive the account’s 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 “PRIZM-”, 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 (2256) is larger than the address space for account numbers (264), 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.

Account Balance Properties

For each Prizm 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 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 *basic balance* of an account accounts for all transactions that have had at least one confirmation.
- The *forged balance* of an account shows the total quantity of PZM that have been earned as a result of successfully forging blocks.
- The *unconfirmed balance* of an account is the one that is displayed in Prizm clients. It represents the current balance of an account, minus the tokens involved in unconfirmed, sent transactions.
- Guaranteed asset balances lists the guaranteed balances of all the assets associated with a specific account.
- Unconfirmed asset balances lists the unconfirmed balances of all the assets associated with a specific account.

Wallet.dat

Bitcoin and related currencies often use an encrypted file, called a *wallet*, to store generated addresses for receiving tokens. The core design of Next, using by Prizm, does not mimic this functionality, but also does not preclude it. It is possible for client developers to implement a system where a group of private keys for Prizm accounts are stored in an encrypted, offline file.

2.4.3 Transactions

Transactions are the only means by which Prizm accounts can change 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.

Transaction Fees

Transaction fees are the primary mechanism through which PZM are recirculated back into the network. Every transaction requires a minimum fee of 0,5% transaction amount, (minimum 0,05 PZM).

Transaction Confirmations

All Pzm 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.

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 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 because they are invalid or malformed, or because blocks are being filled will transactions that have offered to pay higher transaction fees. In the future, features such as multi-signature transactions may be able to take advantage of deadlines as a means of enforcing an expiry date.

Transaction Creation and Processing

The details of creating and processing a PZM transaction are as follows:

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:

- the private key for the sending account
- a deadline for the transaction
- an optional referenced transaction

2.5 Paramining

Paramining technology is the key advantage of Prizm over other crypto-currencies. The Prizm developers have created the unique, linear-retrograde mechanism for determining the reward for the store of value, aimed at economic attractiveness and the gradual replacement of all world existing financial instruments by the mass of PZM. This is in addition to the basic forging mechanism, which is not increases the amount of funds in the system. Paramining technology allows to generate new tokens, according to the metrics of the standard mathematic of a normalized financial system development in context of global economy. According to our calculations, only this format for the growth of the tokens' mass can ensure a gradual and confident replacement of all existing economic instruments.

The working mechanism

According to its characteristics, Paramaming is a MLM 2.0 system, excluding everything that pushes an ordinary person away from network marketing, but at the same time involves him in the network development to increase the tokens mining speed in a personal wallet.

The speed of new tokens' generation via Paramaming mechanism is based on two basic parameters, this is the number of tokens in a personal wallet and the number of tokens in wallets of followers up to 888 levels.

1. The number of tokens in the personal wallet

Daily growth of coins number, %	The number of tokens in the personal wallet
0,12%	от 1 до 99
0,14%	от 100 до 999
0,18%	от 1000 до 9999
0,21%	от 10000 до 49999
0,25%	от 50000 до 99999
0,28%	от 100000 до 499999
0,33%	от 500000 до 1000000

2. The number of coins in the downline wallets of 888 levels down.

Multiplying factor	The number of coins of the whole structure
2,18	от 1000 до 9999

2,36	от 10000 до 99999
2,77	от 100000 до 999999
3,05	от 1000000 до 9999999
3,36	от 10000000 до 99999999
3,88	от 100000000 до 999999999
4,37	от 1000000000

Paramining principle is based on the fundamental laws of physics, from the field of “Visible radiation”. Like the model of our universe, the system is constantly expanding with greater speed.

0.12% * 2.18 = 0.26% in 24 hours.

TOTAL: More than 8% of new tokens per month.

0.18% * 2.77 = 0.49% in 24 hours.

TOTAL: More than 15% of new tokens per month.

For example: If someone has 99 PZM in the personal wallet and 100,000 PZM in 888 levels of his structure, the percentage of the tokens amount growth is 0.12% and the multiplier is 2.77, which allows to generate 3.3 new tokens daily. It is enough to make any transaction to get these tokens on the personal balance.

The Paramining system, while processing transactions in the wallet, makes a note in the Blockchain containing information of tokens amount of the wallet owner and the number of tokens in the wallets of his followers. At this moment new coins are generated in the wallet balance.

Thus, we get a system with the compound interest that stimulates users to make transactions to increase their funds capitalization, to connect new wallets’ holders, thereby increasing the turnover of the structure. According to the most conservative calculations, the monthly token amount growth for the account holder is 10% at least.

Referral connections

For the first time, the system of establishing referral connections without using any referral links has been applied. After the creation of a new wallet, the system captures the first transaction from the account in the Blockchain and forever sets up a referral chain that can not be changed. It makes easy to build global MLM networks and increase the speed of new tokens generation.

Technical implementation of Paramining technology has not been described in detail, because the main thing, for all of us, is to create not a hundred of "dead" tools, but the one, with good support and working well. If our

know-how is revealed, then someone will definitely try to repeat it and this involuntarily will lead to the dispersion of attention and use of this idea not for noble and important for our planet purposes, but for purposes not known to us and not always differing in positive coloring intent.

For starting the generating of new PZM tokens, it is enough to have the balance of just one token in the personal wallet, which in turn will automatically launch the Paramining. This process allows to increase the number of tokens in the wallet without extra electricity consumption.

Paramining technology works in every Prizm wallet and stops automatically after reaching the balance of 1 million tokens in the wallet.
The Paramaming system is the perfect tool for promotion and popularization, as it has no analogues in any modern crypto currency.

The key advantage of Paramining is uniqueness of situation where nobody can interfere with this mechanism and falsify new coins, because all PZM users can monitor the number of tokens issued by the system in the real-time.

2.6 Cryptographic Foundations

Key exchange in Prizm 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.

Message signing in Prizm 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.

2.6.1 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:

seedn = SHA256(seed n-1), where seed 0 = SHA256 (shared_secret)

3. Calculates N keys:

- keyn = SHA256(Inv(seedn)), where Inv(X) is the inversion of all bits of X

4. Encrypts the plaintext:

- ciphertext[n] = plaintext[n] XOR keyn

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 Alice's step):

- seedn = SHA256(seed n-1), where seed0 = SHA256(shared_secret)

3. Calculates N keys (this is identical to Alice's step):

- keyn = SHA256(Inv(seed n)), where Inv(X) is the inversion of all bits of X

4. Decrypts the ciphertext:

- plaintext[n] = ciphertext[n] XOR keyn

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.

3 Core Features

3.1 Advanced JavaScript client

A second-generation, user-friendly client application is built into the Prizm core software distribution, and can be accessed through a local web browser. The client provides full support for all core Prizm 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 Prizm's low-level Applications Programming Interface.

3.2 Basic Payments

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

3.3 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, Prizm is extremely well suited for use on small, low-power, low-resource devices. Android and iPhone applications are currently in development, and the software has been ported to low-powered ARM devices such as the RaspberryPi and CubieTruck platforms.

The ability to implement Prizm on low-powered, always-connected devices such as smartphones allows us to envision a scenario where the majority of the Prizm 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.

4 Concerns

4.1 Proof of Stake Attacks.

4.1.1 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 Prizm 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.

4.1.2 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 Prizm, 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 Prizm 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.

Appendix: Solutions to some Bitcoin Problems, used by Prizm

Prizm adopts features that have proved to work well in Bitcoin, and addresses aspects that are cause for trouble. This appendix addresses some issues with the Bitcoin protocol and network that are mitigated in Prizm by used alternatives.

Blockchain Size

Bitcoin 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” that employ simplified payment verification (SPV). 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 Bitcoin’s Mike Hearn, “Instead of verifying the entire contents, [SPV] just trusts that the majority of miners are honest.

Prizm Transactions per Day

In its current state, the Prizm network can process up to 367,200 transactions per day – more than nine times Bitcoin's current peak values. Implementation of Transparent Forging allows for near instant transaction processing, drastically increasing this limit.

Bitcoin 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.

Prizm Transaction Confirmation Time

The average block generation time for PZM has historically been shown to be about 80 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 14 minutes.

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

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 46 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.

Prizm Solutions

The incentives provided by Next's Proof of Stake algorithm, using by Prizm, 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 Prizm network has remained highly decentralized since its inception: a large (and growing) number of unique accounts are contributing blocks to the network, and the top five accounts have generated 35% of the total number of blocks.

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.

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.

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.

Proof of Work's Resource Costs Pertaining to Coin holders

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. This produces 10% inflation in the total Bitcoin supply a year. 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.