0CHAIN

0Chain: A Fast, Secure, Scalable, & Free Blockchain

Tailored For High-Performance, Zero-Trust, Decentralized Storage

Saswata Basu, Thomas Austin, Siva Dirisala, & **0Chain** Team

Whitepaper V2.0 Last Updated Mar. 20, 2019 Subject To Further Review & Update

Table Of Contents

Motivation

When we started developing 0Chain in July 2017, we were driven by the idea of redefining the cloud in the context of privacy, transparency, and user control of data. In 2018, as we dove deeper into our development process, we identified a number of critical unresolved issues in the cryptocurrency space, such as: token security, token valuation metrics, governance, and token inflation. To address the aforementioned issues, we chose to conduct additional research and development. Today, we are not just redefining the cloud, but the blockchain and cryptocurrency landscape through our novel protocols.

Executive Summary

Several roadblocks exist in today's crypto market, as well as the underlying Distributed Ledger Technologies (DLT) that power them — hindering the maturation process of the blockchain industry. Some are regulatory and security concerns, some involve technical performance, while other issues are related to token economics, valuation metrics, and governance.

1. SECURITY

Regulators identify a few key shortcomings with regards to the security of crypto assets: poor wallet infrastructure, exchange transparency, and unreliable custody of assets. 0Chain is addressing all of these issues.

Today, hardware wallets are clunky and extremely hard to use for an average person, especially for daily utility. Software wallets are notoriously prone to hacking, unless you can memorize your private key. Our software-based secure wallet is the world's first to enable a 2-device authentication; it makes transactions simple and yet highly secure with just a mobile device and a laptop.

2. SPEED AND SCALABILITY

The finality for Bitcoin is about 1 hour; Ethereum is about 3-10 minutes — something that is still too slow for a micropayment transaction and verification. We aim to address this problem. 0ChainNet is a high speed blockchain with block finality achieved within . 5 to 1.5 seconds (depending on network latency), highly scalable with throughput rates of over 1,000 transactions per second, and energy efficient by way of our unique Proof of Stake protocol. We have proven these results on a public test network of 10 worldwide data centers and have completed over 6 billion transactions to verify reliability.

3. ARCHITECTURE

An Ethereum node mines a block, stores the block, and stores associated unstructured data, all on the same node. This same node handles transactions and queries from the same client. This architecture makes the node very expensive, slow and unscalable. 0Chain architecture allows for separation of duties to specific designated nodes, called miners, sharders, and blobbers. Miners receive transactions from users and they generate blocks via the consensus protocol, Sharders store these blocks and respond to queries on transactions and blocks, and Blobbers store unstructured data. This architecture allows for off-the-shelf cheap hardware (making it inexpensive to support the network as a node), faster response rates, and better scalability.

4. ZCN TOKEN

The value of ZCN is mathematically related to the data stored and other services on the network, unlike other cryptocurrency.

4.1 Addressing Cost, Value, And Volatility

Expensive transaction fees and the volatility of native blockchain cryptocurrencies is an unresolved problem for today's public blockchain networks. To address this pain point, 0Chain's native cryptocurrency (ZCN) is uniquely programmed as an asset-backed token.

When a user locks their ZCN, these tokens collect an "*interest"* which can be used toward payment of transactions or data services. Through this interest-bearing feature, the transaction fee is absorbed by the interest generated on the locked tokens. Additionally, 0ChainNet's core service — data storage — is also enabled by locked tokens. Upon locking, a storage service can be activated and the extent of service is determined by the number of tokens locked by the user (for more details refer to section 3.2). In both of these services, the initial locked tokens are fully redeemable upon unlocking, thus facilitating free services on the network. It's free because the network mints these tokens and is part of the inflation and underlying token economics.

Storage and transaction services have a quantifiable, real world market value. Conversely, it's frequently argued that other popular native blockchain cryptocurrencies (such as BTC, ETH, etc.) in their current state lack such an economic value beyond raw speculation. This creates a challenging process for accurate price discovery of cryptocurrencies, resulting in price volatility.

0Chain has programmed value economics into the ZCN token to curb price volatility. The lower bound value of the ZCN token can be mathematically estimated based on the number of tokens locked relative to the demand or usage of data storage and transaction services (for more details refer to section 3.4).

In other words, ZCN is backed by an allocation of transaction and data storage services, thus injecting a non-speculative, integral value into the ZCN token. The asset-backed nature of ZCN can buttress its market value and reduce its price volatility unlike other popular native cryptocurrencies.

4.2 Token Supply

The ZCN token supply (at genesis block) can be broken down as follows:

A) **Public Sale**: 40m sold to the public in the private and pre-sale rounds.

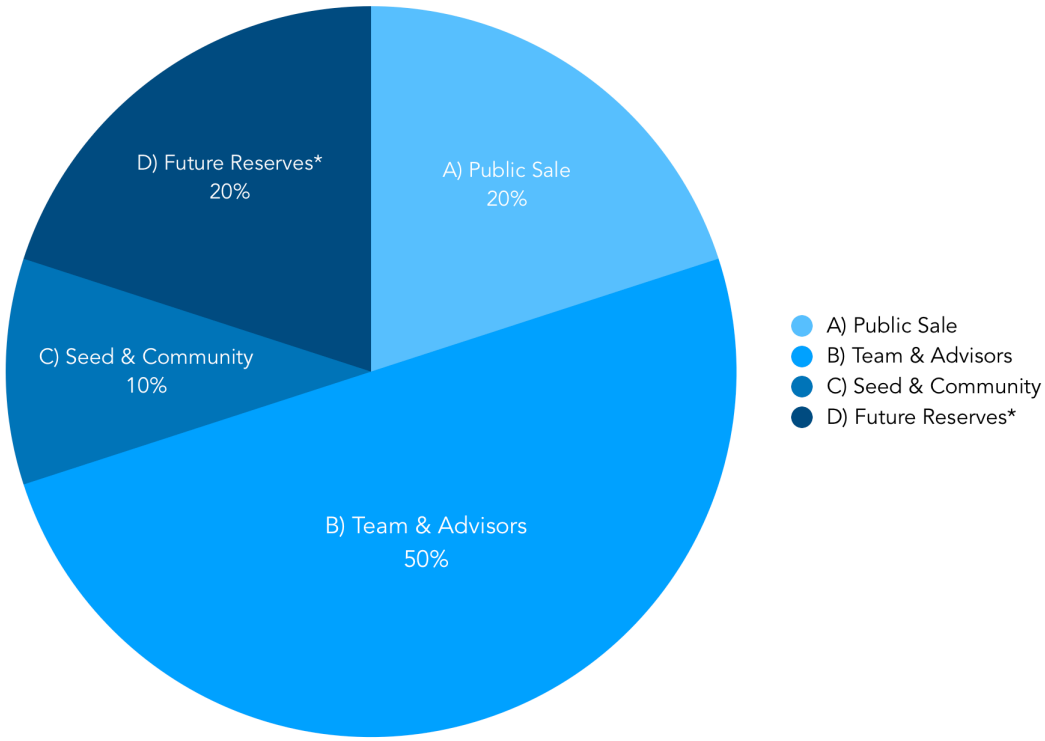
B) **Team & Advisors**: 100m allocated to team & advisors.

C) **Seed & Community**: 20m allocated to seed & community (eg bounties)

D) **Future Reserves**: 40m reserved for future use if the per unit token price of ZCN exceeds $10. This reserve unlocks in two 20m tranches in Jan 2020 and Jan 2022 if the $10/ZCN threshold is met.

ZCN Supply Breakdown

Category	ZCN	% of Supply
A) Public Sale	40,000,000	20%
B) Team & Advisors	100,000,000	50%
C) Seed & Community	20,000,000	10%
D) Future Reserves ($10/ZCN vesting provision)	40,000,000	20%
Total	***200,000,000***	***100%***

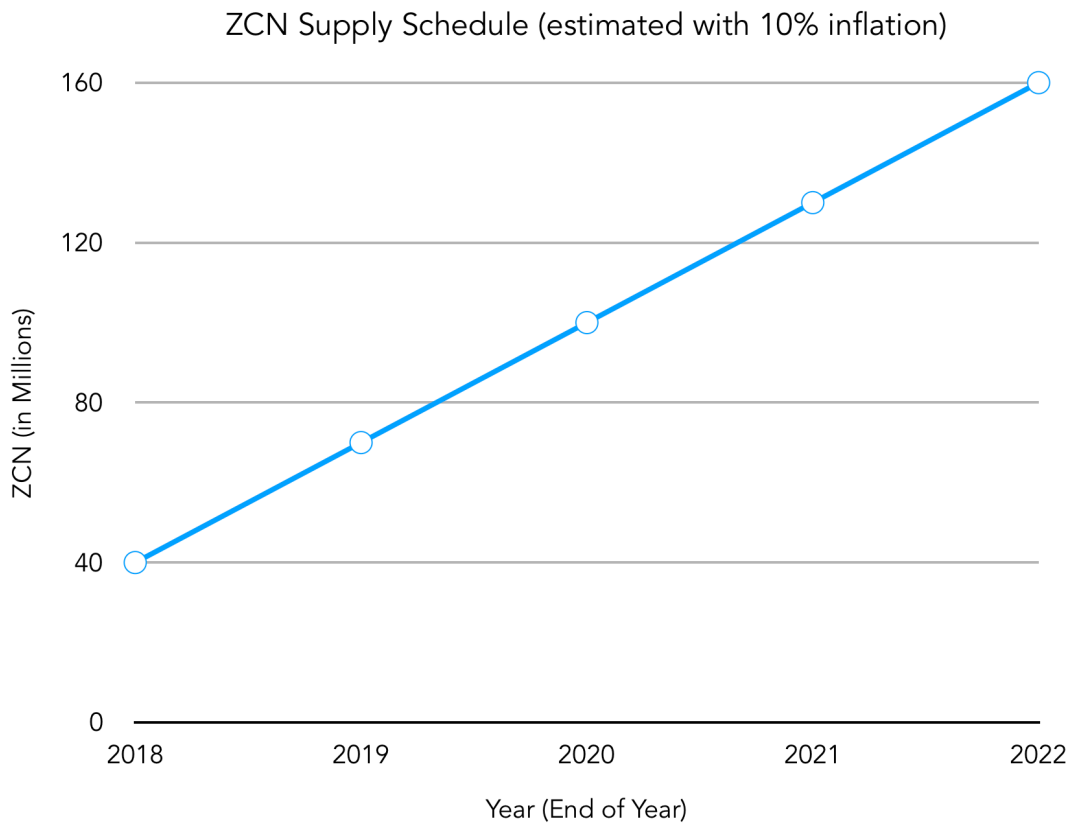


**Future Reserves unlocks only if the per unit market price of ZCN exceeds $10.*

4.2.1 Token Supply Schedule

The current inflation rate of ZCN will run at 10% per year of the outstanding token supply. The 120m tokens from quadrants “B) Team & Advisors” & “C) Seed & Community” are vested linearly over a 4 year period beginning in January 2018. To summarize, at the end of 2019 there will be 70m ZCN outstanding, 100m in 2020, 130m in 2021, 160m in 2022, etc. The 40m ZCN from quadrant “D) Future Reserves” will fully vest after four years (if, and only if, price exceeds $10/ZCN).

The chart below depicts the aforementioned schedule:



**Future Reserves excluded. Future Reserves unlocks only if the per unit token price of ZCN exceeds $10. This reserve unlocks in two 20m tranches in Jan 2020 and Jan 2022 if the $10/ZCN threshold is met.*

5. GOVERNANCE

Upgrades in the context of a protocol are not just unavoidable, but critical. The mechanisms being used by public blockchains to initiate protocol upgrades leave a lot to be desired. Today, changes are slow, and occasionally result in a contested fork; which can have dire implications, as evidenced by the divisive chain splits seen on the Ethereum and Bitcoin networks. 0Chain governance enables a fast but fair approach to all issues, ranging from configurable changes to major code upgrades.

6. ENTERPRISE MARKET

Today's blockchain solutions for the enterprise market are disjointed. While HyperLedger, Corda, and Ethereum platforms provide a barebones blockchain (or block-less in the case of Corda), they do not solve issues regarding governance, profit sharing, addition/removal of consortium members, and verifiable storage of data. Most of the latter issues need to be developed, verified, and agreed upon by all parties so that they can trust the blockchain system. Thus, it is not simple for enterprises to transition to a blockchain system and roll-out new products on it.

In addition, the notion of a private blockchain in a datacenter under the control of a central party (or developed by such) always has the nagging complaint of being a glorified version of a traditional centralized system. And so, it is likely that a centralized blockchain system is a good transition market for enterprise products before they move to the public chain in the future. 0Chain provides a suite of protocols that addresses all of these issues, and so enterprise private chains can use 0Chain to abstract out the infrastructure and protocols, in order to develop and market new applications at a faster pace.

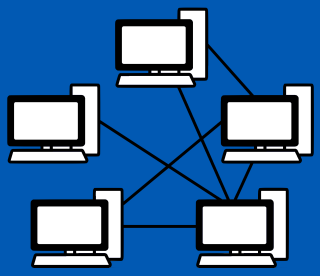
Section 1
An Introduction To ØChain

ØWallet

Secure Wallet

ØBox

Storage App



ØChainNet

Fast Blockchain Infrastructure

0Chain aims to introduce new products and services based on 0ChainNet, as it conducts ongoing research, development, and code releases for 0ChainNet.

0ChainNet is a fast, secure, public enterprise-grade blockchain powered by an innovative and original consensus protocol. It enables decentralized applications such as 0Box to abstract infrastructure and zero-trust protocols and build a fiat application on 0Chain. 0ChainNet is secured by a native cryptocurrency (ZCN), which enables a fast, secure, and free way to store and transfer value.

0Box: transparent, zero-trust dStorage for your data.

0Box is the first product 0Chain will unveil on 0ChainNet. Similar to how a browser facilitated wide adoption of Internet protocols, the intention of **0Box** is to facilitate wide adoption of public blockchain protocols. **0Box** is a platform to protect data privacy and provide transparent content monetization for both the consumer and the enterprise.

Split-Key Wallet: the convenience of a software wallet; the security of a hardware wallet.

User A can receive transactions.
To send transactions, User A needs laptop & mobile

The **Split-Key Wallet** is 0Chain's breakthrough innovation for crypto wallets and passwordless login security. No more software wallet hacks. More secure than traditional 2FA, and more convenient than a hardware wallet. All you need are 2 devices and your PIN to send a transaction.

ØCHAIN

Section 2
Products

2. Products

0ChainNet is designed to serve both the crypto and enterprise industries, with the ability for enterprises and blockchains to leverage 0Box for their decentralized storage requirement. Both industries have the ability to integrate with 0Box and use the 0ChainNet infrastructure as a high-performance zero-trust cloud service.

2.1 0ChainNet

2.1.1. MULTI-SIG WALLET FOR THE CORPORATE ENVIRONMENT

The 0Chain platform has a built in multi-sig wallet, and unlike other implementations, is cryptographically secure to enable m-of-n keys to execute a smart contract. Exchanges and corporations can use it to dispense funds based on several signatures, instead of one signature. It creates protection from multiple bad actors, and hacks on multiple servers. By having the multi-signature feature built into the platform, any smart contract can use this feature for a highly secure execution of value transfer.

2.1.2. SPEED

The speed of a transaction is ultimately determined by when that transaction is fully processed and confirmed, also known as finality. These are the definitions of finality:

1) **Steady-state block finality** is based on the rate at which a finalized block is created every block generation round, and can be directly correlated with transactions per second. So, if the block size averages 1000 transactions, and steady-state finality is a second, then the transactions per second (TPS) is 1000.

2) **End-to-end block finality** is defined as the time taken from the generation of a block to the finalization of that block; this timeframe is typically about two to three times that of steady-state finality.

3) **Transaction finality** is based on the time the transaction is sent by the user, to the miner, to the time a block with that transaction is finalized. Transaction finality is typically longer than block finality by about a 1.5 to 2 times multiple, because the transaction depends on the internet speed quality of the end user. Even if the connection is perfect, the transaction submitted could miss the block generation event. In our experiments, which included 130 world-wide nodes (100 miners, 30 sharders) with an average network latency of 700 ms, the steady-state finality was roughly ~1.2s, block finality was ~4.6s, and transaction finality was ~5.4s.

4) **Deterministic finality** is when you have 100% certainty that there will be no rollbacks. This lags all finality by a multiple that will be discussed in a blog in the future.

In another context, a 0Chain self-fork could have a distribution of nodes strictly within the U.S., and then the network latency would be about 200ms. Additionally, if the transactions are less than 100 per block, the steady-state finality would clock in at 300ms, with block finality at 900ms, and transaction finality at about 1s. With such speed, a decentralized exchange would be extremely fast, rivaling today's centralized exchanges and solving the problem of token custody & transparency.

2.1.3. SCALABILITY

It's expected for every chain on 0ChainNet to accommodate at least 1,000 transactions per second without a hitch. As the number of cores, memory, SSD, and IO capability increase, scaling as high as 100,000 transactions per second is possible; but this will incur high costs, and is unnecessary for any app today, including VISA. A more economical solution is scaling vertically, through additional chains based on demand. The best way to scale is to have a self-forking feature, whereby a forked chain can be formed with all associated protocols, features, and abstracted infrastructure in a trustless fashion.

2.1.4. COMPETITION

Several public blockchain solutions exist on the market today (e.g., Ethereum, Dfinity, EOS, Tezos, Tron, etc.), as well as private ones (e.g., HyperLedger and Corda), all with their own advantages. 0Chain is carving out its own identity as the first enterprise-grade dStorage blockchain platform with unparalleled transaction speeds, scalability and an integrated asset-backed token economy, that can integrate with existing blockchain solutions.

2.1.5. GO-TO-MARKET STRATEGY

Building a robust community and a strong developer environment is not a simple task. One of the core strategies is to have a constant supply of bounty and competition-issued tokens to promote growth of the network. 0Chain expects to be a participant in mining, and some of the accumulated interest proceeds will go toward such marketing efforts. The following strategies will be considered to educate the market:

- Bounty-issued locked and unlocked tokens for transaction and storage, based on shared link referral.
- Bounty-issued locked and unlocked referral tokens; referrals verified by the team or smart contract code.
- Bounty-issued locked and unlocked tokens for bug fixes on our governance protocol.
- A dApp “Starter Package” of locked and unlocked tokens, rewarding early applications built on the 0ChainNet platform.
- Token incentives for researchers at Universities to conduct research on our protocols. Winner selected using the 0ChainNet governance protocol.
- App competition. Winner package including locked and unlocked tokens, selected via the governance protocol.

As 0Chain LLC raises additional capital, we expect to spend a large portion on marketing efforts in the following areas:

- University initiatives
- Meetup groups
- Accelerator engagements
- 0Chain Conferences
- Hackathons

2.1.7. PARTNERS

As a small team, 0Chain must be selective with our partners. We are working closely with the following impactful opportunities:

2.1.7.1. PollGateWay – eVoting/eOpinion platform

A startup in India addressing their customers' issue of transparent, authentic, and anonymous voting processes via the blockchain. However, most notably, the core issue being solved is efficiency: long lines, hours of waiting, retries, recounts, and setting up booths are all inconveniences solved by a more efficient voting process.

2.1.7.2. MyntCoinz – White labeled loyalty platform

A U.S. startup offering a BaaS (Blockchain as a Service) platform for the "*loyalty rewards*" space. MyntCoinz offers a unique solution to transform various customer engagement issues, such as a lack of flexibility and liquidity in the loyalty market.

2.1.7.3. Department of Homeland Security (DHS) – Identity fraud prevention platform

DHS seeks technical solutions that can serve the needs of U.S. Customs and Border Protection (CBP), U.S. Citizenship and Immigration Services (USCIS), and Transportation Security Administration (TSA). DHS is interested in using blockchain to address the challenges of interoperable digital entitlement attestations that support individual control and accountability of data release, while incorporating digital counter-fraud technologies and tactics, enterprise lifecycle management, and a high degree of usability across service delivery modalities.

2.1.7.4 AWS – Infrastructure and BaaS partner

We have been working with AWS for over 6 months and have been granted $100k of cloud resources, which has been valuable in rolling out our Blockchain as a Service platform. We also passed their technical requirement for disaster recovery and cloud service processes to get accepted in their Advanced Solution program. Moving forward, we'll work with their sales team on opportunities.

2.1.7.5 Oracle – Infrastructure and BaaS partner

We are a partner in Oracle's standard program, and plan to bring our BaaS to their customers. One of their requirements to do so is to have a platform up on their datacenter. Given our limited resources, we have decided to wait until the AWS opportunity and 0ChainNet are both live in the marketplace — something we expect for Q2 2019.

2.2. **0BOX**

The second product that we're developing in parallel is 0Box, a data privacy and transparent decentralized storage (dStorage) platform. 0Box will be used to accomplish two core objectives:

1. Showcase a native dApp that runs on 0ChainNet as a model for other dApps.
2. Facilitate use of decentralized storage on 0ChainNet to "*seed*" adoption of blockchain.

2.2.1. HOW IT WORKS

Users lock tokens or directly allocate required tokens for a specific storage **allocation** (e.g. 1TB). The allocation is created one time when the miner randomly chooses the least expensive blobbers. The list of blobbers are on the blockchain, since each service provider of 0Chain needs to register with the network. If the blobbers offer the same price, then they are randomly picked. The client collects all signed contracts from the blobbers and sends a transaction with the list to the blockchain. Then it starts uploading files to the blobbers. Once the file is committed, people can read from the blobbers. The health (reliability, availability) of the file is checked on a regular basis through challenges, and if anything is amiss, then the repair protocol takes care of it in the background.

Users can upload as many files as they wish. Files can be private or public and can be shared via links on any platform. When a user clicks on the link, it invokes the 0Chain mobile or desktop app to download the file.

2.2. **0BOX** (cont'd)

Regarding competition, we do not directly compete with DropBox, Box, or other data sync solutions, since our target market are users that value data privacy and transparency — for which a practical solution does not exist. These consumers care about their personal data, especially their images, posts, and videos. 0Box enables them to protect their images and videos by placing them on 0Box, and pasting a link on their social wall for their friends to download and view those images and videos. Users will be able to monitor a file's downloads on the blockchain, and be assured of their privacy.

Consumer content platforms such as YouTube, Spotify, Apple, and Pandora do not address small artists. New rules alienate small publishers. 0Box can accommodate both of the following types of content creators:

1) A well-known publisher, with a large social media following (1m followers or more)

2) A small publisher, with a small social media following (1k followers or less)

The small publisher can host content on YouTube and share the video for free, or host it on 0Box (for free via locked tokens) and charge a nominal fee to view their content. In the latter case, the user generates some revenue instead of none. If the well-known artist has a million followers, they can share the 0Box link and price it lower than they expect to get from ad clicks. The artist then promotes the content on social networks for people to watch it. The average cost per thousand impressions (CPM) for YouTube is $9.68 and $3.21 for CPC (2018). A video with 1000 views would earn them 9.68. If the artist sells the video for $0.25c to their fans, then only 40 out of 1000 followers need to download it to achieve the same result, but the artist gets to keep all of the revenue. For a user's perspective, they can get access to the content, free of ads, with their interest tokens, and still directly support the artist. 0Box provides a win-win solution for content creators and consumers. The monetization feature of 0Box will be offered later in the roadmap.

2.2. **0BOX** (cont'd)

For bloggers and other content creators, they can utilize 0Box share links for monetized content, and get paid via a share link. They can simply write a teaser article on Medium, Twitter, Facebook, or brand name journal publication, and have people download the full content via the share link.

2.2.2. COMPETITION

While several providers of storage and data sync solutions exist in today's market, 0Box is addressing an unserved need for data privacy, security, and a transparent dStorage.

2.2.3. GO-TO-MARKET STRATEGY

Similar to the 0ChainNet strategy, the 0Box viral marketing strategy for building out a strong network will be accomplished through a combination of the following:

- Bounty-issued (locked and unlocked) ZCN tokens for creative artists.
- Bounty-issued (locked and unlocked) ZCN tokens for 0Box based on referrals.
- Reward-issued (locked and unlocked) ZCN tokens for content competitions (eg, "best creative content" contest, with winner determined via the 0Chain governance protocol)
- Bounty-issued (locked and unlocked) ZCN tokens based on most downloaded content.

2.2.2. GO-TO-MARKET STRATEGY *(CONT'D)*

We also expect to allocate substantial funding towards marketing efforts in the following areas:

- Specific Content groups (art, graphics, music, video, clips, gifs, blogs, reports)
- Privacy meetup groups
- Accelerator engagements using API using 0Box
- 0Chain Conferences

2.2.3. TELECOM CHANNEL PARTNERS

We are working with three telecom operators to leverage our 0Box solution to provide a better cloud service for their customers, and increase their top line with higher data rates. Our proposal is a freemium data model for bandwidth related to cloud usage. In doing so, subscriber acquisition cost reduces substantially, increasing their top line over time with specific data plan introductions built around 0Box.

2.2.4. CYBERSECURITY CHANNEL PARTNERS

We are selectively pursuing customers and partners that can most effectively leverage 0Box to address the issues of data privacy and data breach, particularly within the healthcare sector. With 0Box's hassle-free integration platform for existing websites, internet processes, and healthcare companies, a 0Box solution is practical for both the consumer and enterprise.

2.3. **0WALLET**

Our 0Wallet product allows for secure value transfer without the need for hardware wallet technology. A 0Wallet user can keep their secret phrase in a safe place, and use it to generate “split-keys”, which are factor authenticators via a device of the user’s choosing, such as a laptop and a mobile phone. Since the pin code is used on the second device, even if one of the devices or both devices are compromised, the user needs the pin code to send a transaction. There is no need to remember back-up codes for traditional 2FA, nor a need for hardware wallets.

Split-key wallet technology can be used for different use cases like password-less logins for websites and applications. It can be also be used with hardware devices such as RSA type keys used in banks. Additional use cases are keyless entry for cars and buildings.

For more details, reference Appendix 5 in the Appendices section.

Section 3
0Chain Architecture

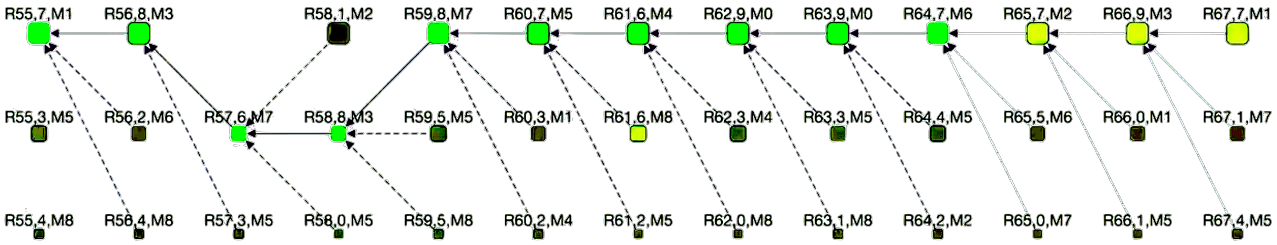
3. 0Chain Architecture

The value of 0ChainNet is tied to the protocol design and implementation. We believe we have an unrivaled, innovative design not seen in today's Distributed Ledger Technology (DLT) space, especially with regards to network security and asset-based service solutions. Many of these protocols have been rigorously tested and demonstrated on private and public test networks.

3.1. CONSENSUS PROTOCOL

0ChainNet offers a fast, secure, and scalable blockchain through a proof-of-stake consensus protocol that extends existing work of Dfinity protocol in several ways. The 0Chain protocol assigns various parties in the system with specialized roles: 0Chain has multiple generators produce blocks to prevent DDoS attacks, generate random numbers, and verify blocks; sharders store the blockchain history and respond to queries about that history; and blobbers store data needed for dApps.

0Chain's Block Generation Process



This design allows for more specialized machines to be used for each of these roles. Otherwise queries will bog down miners, and storing large data files will slow down miners, if they were to do all the tasks. 0Chain also introduces a squared staking approach for Sybil resistance, by which miners and sharders are probabilistically chosen based on the square of the number of tokens they have staked; this design incentivizes miners and sharders to stake their coins in a single account, and thus risk greater penalties should they fail to perform their duty.

Finally, the 0Chain consensus protocol makes very mild assumptions about the network latency to allow for faster confirmation time, because nodes do not need to wait a fixed time in order to progress, but can instead progress shortly after they receive their expected messages.

3.2. **STORAGE PROTOCOL** [1]

A key distinction of our data storage system from other blockchain storage solutions is that we divorce the role of mining from that of providing storage. Computers that provide storage are referred to as blobbers. Blobbers are neither responsible nor required for mining. In this manner, we lighten the load on our mining network and enable fast transactions on a lightweight blockchain. As the client and blobber interact, the client generates special signed receipts called markers. These markers act like checks that the blobber can later cash in with the blockchain.

Once the interaction between client and blobber has concluded, the blobber writes an additional transaction to the blockchain, which redeems the markers for 0Chain tokens and commits the blobber to a Merkle root matching the data stored. The leaves of the Merkle tree must match markers sent from the client, preventing either the client or the blobber from defrauding each other.

After a file has been stored, a challenge protocol ensures both that the blobber continues to store the file and continues to be paid for that work. The mining network posts a transaction, challenging the blobber to prove that it still possesses the data that it was paid to store. The blobber must provide that data, the relevant system metadata, and the client-signed marker to prove that the right data is stored. The blobber is then rewarded or punished accordingly.

With our design, the majority of the work between clients and blobbers happens off-chain. The mining network is only involved enough to ensure that clients pay blobbers for their work and that the blobbers are doing the work that they have been paid to do. Our design assumes that the client is using erasure codes to ensure greater resiliency. While this is not a strict requirement, it does enable a client to recover if a blobber proves to be unreliable.

[1] Will be presented and published at IEEE Dappcon in Apri 2018. Early publication at https://0Chain.net/research

3.3. **SPLIT-KEY WALLET PROTOCOL [2]**

The split-key wallet protocol uses a BLS signature scheme to split keys and let users interact using crypto keys via a blockchain. Since the cryptocurrency balance is maintained against these keys, it's very important to protect the private key. The private key is split into two secondary keys, storing each of the secondary key on a different device. Signing requires individual signatures from each device. Hence, losing any one device can still protect the primary key. In addition, if desired, one of the secondary keys can be further split into two parts; only one of which is stored on the device and the other is a simple PIN that the user has to enter each time. This provides an extra layer of protection in case both devices are compromised. The split-key wallet protocol makes it easy to generate as many split keys as desired providing the ability for the user to periodically rotate the split keys and in the process change the PIN.

3.4. **TOKEN REWARD PROTOCOL [3]**

When clients lock tokens, they are rewarded with an "interest". The interest is newly generated ZCN tokens, intended (but not required) for payment of services on the network. These services can be miner compensation for transaction processing, blobber compensation for storage, or transmitted to any other client in exchange for a service; facilitating a lucrative market for building and running distributed applications. In the event of network congestion, a client may also offer to lock a greater amount of tokens to ensure that their transaction is accepted by the mining network. The token reward protocol creates an economy where ZCN tokens can be used to receive services for "*free*" — meaning, the client does not lose their initial stake, but still adequately compensates the service provider.

[2] Will be presented and published at IEEE Dappcon in Apri 2018. Early publication at https://0Chain.net/research
[3] Will be presented and published at IEEE Dappcon in Apri 2018. Early publication at https://0Chain.net/research

3.5. **GOVERNANCE PROTOCOL**

Our governance protocol enables simple implementation of a variety of lightweight, non-controversial changes, while still supporting more extensive and potentially controversial changes. Our protocol provides this flexibility by supporting different thresholds for different types of changes, which we divide into configuration changes, moderation, and feature requests. We expect that moderation will be relatively non-controversial, that feature requests are likely to be highly controversial, and that configuration changes might be either.

Our design builds on our token-locking reward model. Essentially, clients who own 0Chain tokens may temporarily lock tokens to produce token rewards for service providers. In our voting protocol, these token rewards are treated as votes; by locking more tokens, clients may dedicate more votes to a proposal that they favor. Similarly, they may allocate token rewards against any proposals that they oppose. A critical aspect of our design is that our voting mechanism can measure both whether a proposal has broad community support and the degree of support or opposition from different parties for a specific proposal.

One concern in voting protocols is that a wealthy supporter of a proposal could sweep in during the last moment of voting to pass or defeat a measure before other members of the community can react. This is a particular concern for our protocol, since clients have an economic cost associated with voting for a proposal. Our design addresses this issue by having multiple rounds of voting. If a proposal passes, it is followed by a review period where the community may veto the proposal. If a proposal is vetoed, the community may vote to override the veto. The override itself may be vetoed, which may also be overridden, and so on. Eventually, one side or the other will exceed a threshold that the other side cannot match, and the issue will be settled. To minimize the back and forth votes, the vetoing/overriding faction must exceed the threshold by a fixed buffer amount.

3.5. **OTHER PROTOCOLS**

*Additional protocols are in the works related to View Change, Self-Forking, Proxy-Reencryption, and Economic protocols, as well as The Equilibrium Price of ZCN, and The Mathematical Valuation of ZCN Token. The overarching objective with 0Chain's library of protocols is to provide a complete "a-to-z" platform for a dApp, such as 0Box, to seamlessly operate on our chain without the need to worry about infrastructure and protocols.

Section 4
Appendices

Appendix 1: Team

The inception of **0Chain** began in July 2017, which led to a white paper in December 2017 and funding in February 2018. We have a world-class team, headquartered in downtown San Jose, heart of Silicon Valley. We think out of the box and are impassioned by blockchain technology.

For more details visit https://0chain.net/team.

SASWATA BASU

CEO & FOUNDER

Our founder and CEO, Saswata Basu, has 20 years of experience with startups and corporate product development. His first startup, InSpan, had a successful exit in 2000 to CommScope. He has worked on startup initiatives at Intel and Harris in mobile, wireless backhaul, IoT and energy efficiency sectors.

TOM AUSTIN

CO-FOUNDER

Our co-founder, Tom Austin, is an Associate Professor at San Jose State University, and is a well-renowned cyber security and programming language expert. He is the inventor of our storage and token reward protocols.

SIVA DIRISALA

CTO & VP OF ENGINEERING

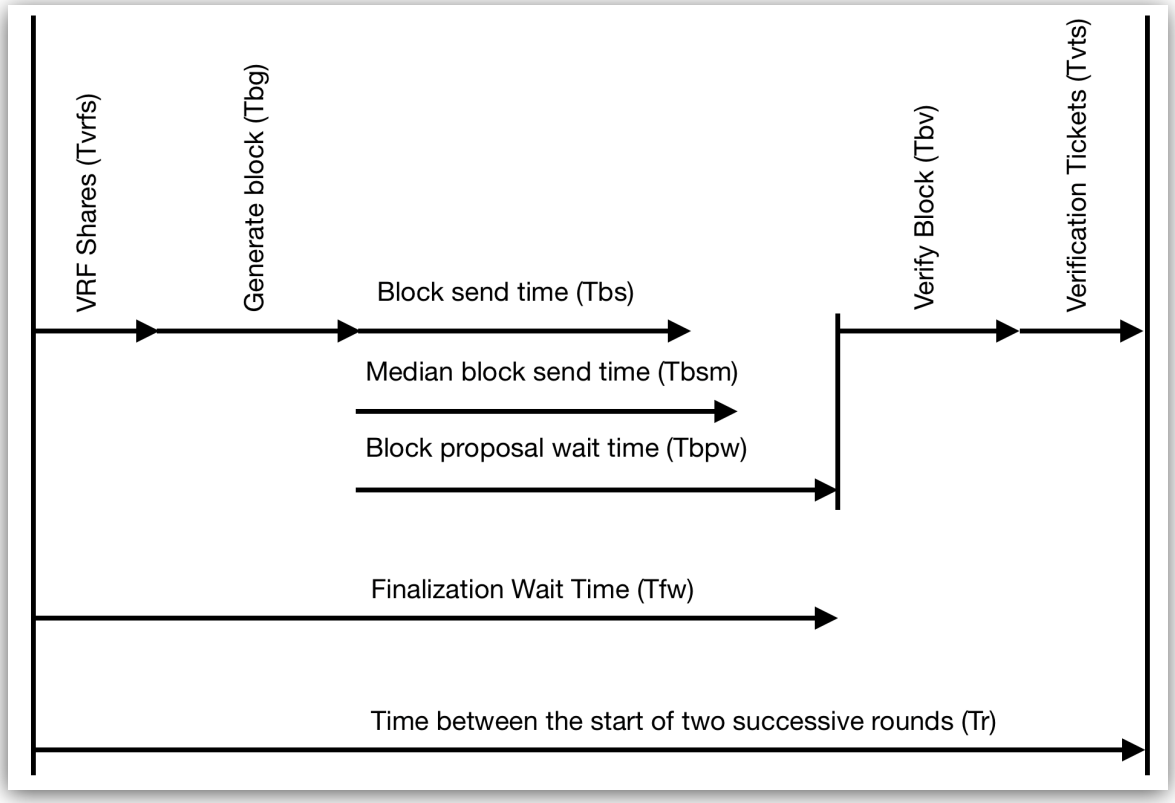
Our Chief Technical Officer, Siva Dirisala, has 20 years of experience as a computer architect and product developer at Service Now & Oracle, and has been solving complex algorithmic and implementation problems his entire career.

Appendix 2: Understanding 0Chain Finality

The 0Chain consensus protocol has been tested on a large cluster of 100 miners with 4 block generators per round and 30 sharders with 6 block replicators per round. The testing is done in a non-Byzantine condition but realistic network conditions by spreading the 100 miners across 14 different data center zones spanning the world (California, Canada, Frankfurt, Ireland, London, Mumbai, Ohio, Oregon, Paris, Seoul, Singapore, Sydney, Tokyo and Virginia).

These experiments and associated data points provide empirical evidence to how the finality time is related to the various underlying parameters. The following diagram illustrates a full cycle of what happens between two successive rounds getting started.

1. **CONDITIONS**



When a notarized block is first discovered in a round, a miner does two things:

1) Start the next round.

2) Wait to start the finalization process for the previous round.

These two processes are independent and are shown as two separate timelines above. We define the steady state finality of a block is the time between invoking finalization between two successive rounds. So, as long as the finalization wait time, Tfw, is less than the total time to generate, send and verify a block referred to as round processing time, Tr, then the steady state finality of a block is given by Tr.

1. **CONDITIONS *(CONT'D)***

```
Tr = VRF Share send time + Block Generation time + Block Proposal Wait time + Block
Verification Time + Verification tickets send time.
```

Note that this is the ideal time where all of these are assumed to happen simultaneously at all the miners, but in reality there will be some variance of when each of these steps are started at each miner.

The above formula gives the following key insights

1) If the block proposal wait time is sufficiently large, every node receives all the blocks being proposed for the round before the verification starts. This ensures only one block is signed (for honest nodes)

2) If the block proposal wait time is too little, most blocks will arrive after the wait time and hence the chance of signing lower ranked blocks increases resulting in several blocks getting notarized in a given round.

Another option is to use a dynamic block proposal wait time. Under this scheme, as long as there is at least one generator whose network time is less than the median network time, a miner will choose that as the block proposal wait time. If in a round, there are no generators whose network time is less than the median, then the configured block proposal wait time will be used. If there are multiple generators whose network time is less than the median, then among all these generators whoever has the higher ranked block, their network time is selected as the block proposal wait time. This scheme tries to balance between waiting arbitrarily too long and too short striving to speed up the finality and reduce number of notarized blocks in a given round by signing fewer blocks.

1. **CONDITIONS *(CONT'D)***

We define steady state finality (time between running two successive finalizations) and start-to-finish block finality (time between the starting of a block and eventually finalizing it). Note that the finality here is a local view of the individual miner.

- Steady state block finality time = Tr (when Tfw < Tr)
- Start-to-finish block finality time = Finality Lag * Tr + Tfw
- Where finality lag is the number of blocks by which the finality lags.

2. **RESULTS**

In our experiments, we found that the block generation time and verification times are significant and comparable to network times and even the network time itself is different for short messages such as sending VRF shares and verification tickets vs sending a large message like a block. For example, the block message time was 3 times or more than the short message time. Based on these observations, in our protocol, every miner maintains the time it takes to send a small message and a large message to every other miner and can be configured to use these data points to dynamically optimize the finality time.

2. **RESULTS *(CONT'D)***

In our experiments, we found that the block generation time and verification times are significant and comparable to network times and even the network time itself is different for short messages such as sending VRF shares and verification tickets vs sending a large message like a block. For example, the block message time was 3 times or more than the short message time. Based on these observations, in our protocol, every miner maintains the time it takes to send a small message and a large message to every other miner and can be configured to use these data points to dynamically optimize the finality time.

The experiments were conducted by varying the block proposal wait time, wait mode (static vs dynamic) and the finality wait time.

The below table shows the steady state block finality, start to finish block finality and percent of rounds notarized with 1, 2, 3 and 4 blocks (as we have 4 generators). The Tfw, the finalization wait time, is twice the Network Relay time indicated in the table. In our experiments, the finality lag is 3 blocks. Hence, Start-to-finish finality time = 3 * Tr + Tfw.

Network Relay (ms)	Block Proposal Wait (ms)	Block Proposal Wait Mode	Steady state block Finality	Block Start To Finish	Rounds	Blocks Notarized = 1	Blocks Notarized = 2	Blocks Notarized = 3	Blocks Notarized = 4
200	200	Static	1238.57 ±143.22 ms	4497.22 ±362.72 ms	25805	24539 (95.09%)	1262 (4.89%)	5 (0.01%)	0 (0.00%)
600	200	Static	1251.13 ±159.85 ms	5338.80 ±365.94 ms	28762	27363 (95.13%)	1391 (4.83%)	9 (0.03%)	0 (0.00%)
400	400	Static	1267.28 ±140.92 ms	5013.01 ±357.77 ms	27842	26525 (95.26%)	1318 (4.73%)	0 (0.00%)	0 (0.00%)
200	800	Static	1424.32 ±119.25 ms	5046.32 ±330.31 ms	25598	25454 (99.43%)	145 (0.56%)	0 (0.00%)	0 (0.00%)
200	200	Dynamic	1261.21 ±150.29 ms	4576.53 ±376.98 ms	25988	24804 (95.44%)	1178 (4.53%)	7 (0.02%)	0 (0.00%)
400	400	Dynamic	1262.96 ±153.61 ms	4995.64 ±365.48 ms	26189	24966 (95.32%)	1220 (4.65%)	4 (0.01%)	0 (0.00%)
200	800	Dynamic	1260.01 ±142.89 ms	4564.89 ±359.27 ms	26149	25088 (95.93%)	1059 (4.04%)	3 (0.01%)	0 (0.00%)
200	1600	Dynamic	1306.84 ±186.77 ms	4710.29 ±409.20 ms	25726	24963 (97.03%)	763 (2.96%)	1 (0.00%)	0 (0.00%)

3. **OBSERVATIONS**

The following observations have been made based on the aforementioned results:

1. No rounds with 4-notarized blocks were observed in any of the scenarios.
2. With static block proposal wait time, the steady state finality increases little from 1238.57 ms to 1267.28 ms when the wait time increases from 200 to 400 but increases much higher to 1424.32 when the wait time is 800. When the wait time is too short, it is still required to wait to receive the first block. So, there is a minimum built-in wait time. Hence, any block proposal wait time below this minimum wait time will not reduce the stead state finality. Similarly, any wait above the time when all blocks are received will result in delaying the verification process and hence directly contribute to the rond processing time, Tr.
3. As the static wait time increased, the percent of rounds with single notarized blocks increased showing very high percent at 800 ms wait confirming that majority of the nodes are able to send a block within this time.
4. Increase of the finality wait time (Tfw) from 400 to 1200 didn't impact the steady state finality much by only increasing it from 1238.57 to 1251.13. This is expected because, as long as the finality wait time is less than, Tr, the time between two successive rounds, the steady state finality only depends on Tr. Since, Tr is 1238.57 when the finality time was 400, increasing it to 1200 didn't significantly increase the steady state finality.
5. However, increase in Tfw, does increase the start to finish block finality linearly. For example, the finality increased from 4497.22ms to 5338.80ms and 4497.22+3*(1251.13-1238.57)+(1200-400) = 5334.90 which is close to the observed value of 5338.80 and agrees with the start to finish finality time formula given above.
6. While static wait of 800ms resulted in 99.43% rounds with single notarized blocks, 800ms dynamic wait resulted in only 95.93% but steady state finality reduced from 1424.32ms to 1260.01ms. This is the expected trade-off between improving finality and ending up with multiple notarized blocks in a round.
7. Unlike the static wait, steady state finality is not impacted much with increasing the block wait proposal time under dynamic wait. This is again expected because most of the time the wait will be within the median network time and only occasionally it will exceed that but still be the network time of one of the generators that is smaller than the dynamic wait time. Hence, even after increasing the dynamic wait time to 1600ms, the steady state finality hardly had any impact.

These experimental results provide valuable insights for us to fine tune the key parameters as needed. In dFinity, the process timings such as block generation and verification times are assumed to be 0. The network time is considered the same for a small message and a large message. In 0Chain protocol implementation, the process timings are explicitly considered and also the time for small and large messages is treated separately.

Appendix 3: Consensus Protocol

The 0Chain Consensus Protocol

Jonathan Katz[1⋆], Thomas Austin[2], Siva Dirisala[3], and Saswata Basu[3]

[1] Dept. of Computer Science, University of Maryland.
[2] 0Chain LLC and San Jose State University.
[3] 0Chain LLC.

Abstract. We describe the 0Chain blockchain ecosystem, including a new consensus protocol offering fast finality. We provide proofs of security for the protocol, along with experiment results validating its efficiency under realistic network conditions.

1 Introduction

Since the advent of Bitcoin [Nak09], the blockchain has revolutionized the world of cryptocurrencies and distributed computation. Ethereum [Woo14] further developed this promise by integrating Turing-complete smart contracts into the blockchain for building distributed applications (dApps).

Despite the promise of blockchain protocols, they have been held back by their slow consensus times. For example, in Bitcoin a transaction is not considered finalized until it is six blocks deep in the chain, a process which takes roughly one hour. Newer protocols have attempted to address this limitation by introducing consensus algorithms with faster finalization times.

One such protocol, Dfinity [HMW18], uses a randomness beacon (implemented via a *verifiable random function*, or VRF) for ranking different proposed blocks. The designers also introduce the concept of *notaries* who sign the highest-ranked block in each round. The authors describe notarization as "optimistic consensus"; in most rounds, only one block will be notarized, and in that case the unique notarized block will be finalized soon thereafter. Importantly, only notarized blocks will be accepted as part of a chain by miners; this prevents both selfish mining [ES18] and the "nothing-at-stake" problem [Poe15].

0Chain offers a "fast, flexible, free" platform for dApp development through a proof-of-stake consensus protocol that extends previous work in several ways. First, the 0chain protocol assigns various parties in the system specialized roles: at any given time, a subset of the clients (referred to as the "active set") serve as *miners* running the consensus protocol; in turn, a subset of the miners act as *generators* proposing new transactions. *Sharders* store the blockchain history and respond to queries about that history; and *blobbers* store data needed for dApps. This design allows for more specialized machines to be used for each of these roles; by reducing the number of parties running the consensus protocol at any point in time, it also reduces network latency thus improving finalization

⋆ Work done as part of a consultancy agreement with 0Chain LLC.

Appendix 4: Storage & Token Reward Protocol

Lock and Load: A Model for Free Blockchain Transactions through Token Locking

Paul Merrill and Thomas H. Austin
Research Group / Department of Computer Science
0Chain LLC / San José State University
San Jose, United States

Jenil Thakker and Younghee Park
Department of Computer Engineering
San José State University
San Jose, United States

Justin Rietz
Department of Economics
San José State University
San Jose, United States

***Abstract*—Bitcoin introduced the world to blockchain-based cryptocurrencies, and Ethereum highlighted their value in building distributed applications (dApps). However, the development of blockchain-based applications has been held back by high transaction fees.**

In this paper, we introduce a model for free transactions on the blockchain. Rather than spending tokens for transaction fees, a token owner (known as a *client*) locks tokens to generate new tokens as a reward for the miner who includes the transaction in a block. This *token-locking reward model* eases congestion on the blockchain in the same manner as fees do in protocols like Bitcoin and Ethereum, but without forcing clients to sacrifice their tokens.

This same design can be used to incentivize service providers. We show how a client can lock their tokens to generate new tokens for storage providers, and how this reward mechanism can help to facilitate an audit of the storage provider.

***Index Terms*—blockchain, storage, cryptocurrency economics**

I. Introduction

Our blockchain introduces a *token-locking reward model*; rather than spending tokens, clients lock tokens to pay for services. This act creates more tokens, which can be used to reward miners or other entities for their work.

This model is similar to Bitcoin's design [1]. Early in Bitcoin's history, miners were primarily incentivized to generate blocks through *coinbase transactions* that rewarded miners with newly created bitcoins. As interest in Bitcoin has skyrocketed, clients must offer transaction fees to motivate miners to include their transactions. This design also serves to ease congestion; when demand for transactions increases, clients can raise these fees to increase the odds of their transactions being accepted.

Our model can be seen as a blend of these two mechanisms. Clients lock tokens to generate new tokens for miners, similar to coinbase transaction rewards; but as with transaction fees, a client may offer to lock a greater amount of tokens to ensure that their transaction is accepted by the mining network.

When clients lock tokens, they can give the newly generated tokens to any other client, facilitating a market for creating distributed applications (dApps). In this manner, tokens in our network can be used to buy services for "free", in the sense that the client does not lose their tokens, but still gives something of value to the service providers.

Using our token-locking reward model, we build a sample dApp for storage, the blockchain-observable storage system (BOSS). With BOSS, storage providers are rewarded for their work by clients who lock tokens. We also show how BOSS can ensure that neither clients nor storage providers can cheat one another; this process relies on special, signed *markers* that ensure public agreement between the two parties. Additionally, this agreement can be publicly validated; third parties can verify that the storage provider is storing the agreed-upon data using nothing more than public transactions on the blockchain and signed messages from the client.

A key property of our design is that the clients may give themselves the newly generated tokens (hereafter referred to as interest). An alternate design could restrict a client to only reward service providers. However, that approach would incentivize clients to feign services in order to mint new tokens; we legitimize this behavior and eliminate such shenanigans.

An interesting economic consequence of this design is that it reduces the opportunity cost of holding a token versus holding a fiat currency in an interest bearing bank account. The interest paid at least partially offsets a possible reduction in the token value. If the level of interest paid moved inversely with the token price the interest payment might substantially offset changes in token value, which could be a stabilizing factor. If the token price decreases, people will lock more tokens in expectation of receiving a higher interest rate, and this locking of tokens in effect reduces supply, creating upwards pressure on the token price.

Our paper makes the following contributions:

- We present our token-locking reward model, which enables clients to reward service providers by locking tokens, without needing to sacrifice their tokens.
- We demonstrate how our model can be used to incentivize miners to accept transactions and generate blocks.
- We use our token-locking reward model to build a storage dApp, allowing clients and storage providers to negotiate an agreement for service. As with incentivizing miners, we can reward storage providers for their work without requiring the client to sacrifice tokens.
- We show how signed markers and the blockchain can be used to validate a storage provider's work.
- We provide an economic analysis of our system, showing how our model can reduce the price instability typically associated with cryptocurrencies.

Appendix 5: Split-Key Protocol

0CHAIN

Splitting and Aggregating Signatures in Cryptocurrency Protocols

S. Sharmila Deva Selvi[1], Arinjita Paul[1], C. Pandu Rangan[1], Siva Dirisala[2], and Saswata Basu[2]

[1]Department of Computer Science and Engineering, IIT Madras, India
Email: {sharmila,arinjita,prangan}@cse.iitm.ac.in

[2]0chain LLC, San Jose, USA
Email: {siva,saswata}@0chain.net

ABSTRACT

The blockchain technology and a vast amount of cryptocurrency related activities have generated an unprecedented level of interest among the public. However, even at the entry level, cryptocurrency users need to deal with the complex task of key management. In this paper, we propose a simple way to manage a user's private key, under a reasonable assumption that the user has two devices at his disposal (say a laptop and a mobile phone). We refer to our strategy as *key splitting*. Since these cryptographic keys are used for generating digital signatures, we should take a closer look at the signature schemes that would perform best under key splitting. At the operational level, scalability is one of the main challenges faced by the users and developers. While there are fundamental issues like consensus that challenge scalability, we focus on the computational efficiency in a block formation. Aggregation of signatures is one of the effective solutions to this problem. To this end, we observe that none of the existing signature schemes work well for BOTH key splitting and aggregation. The current popular schemes such as the ones used in Bitcoin or Schnorr's scheme implemented over Elliptic curves are neither suitable for aggregation nor can their keys be split in a convenient and meaningful way. A detailed theoretical and empirical analysis shows that the BLS short signature scheme is best suited for achieving both key splitting and aggregation.

***Index Terms*—Blockchain, key management, wallet, signature, scalability.**

I. INTRODUCTION

The real-world as well as the academic studies on cryptocurrencies and the block chain technology are among the most significant and trendy developments of Information Technology. Block-chain technology is witnessing an exponential growth in interest and technical advancement at this point of time. While these areas are witnessing an unprecedented growth and attention, their deployments face major hurdles at several fronts. One of the major concerns related to this technology is scalability and in general efficiency/reliability of the whole operation. For instance, every user in this community, sooner or later, directly or indirectly, is forced to deal with challenges of maintaining and managing the cryptographic keys that are used. The subtleties and challenges involved in key generation, maintenance and management are well known in security industry and both cryptographic and policy based solutions have been devised in the past. However, in the context of cryptocurrencies, we still do not have satisfactory solutions that would help scalability or ease of use. The second major concern is related to computational efficiency of the tasks performed during the execution of the protocols. One of the most computationally intense and most frequently used cryptographic primitives in blockchain technology is digital signatures. The users need to generate every transaction with appropriate authentication done on the transaction and the minors or validators need to verify/validate the same multiple number of times. In this paper, we focus on the signing process at the users end and verification process at the block formation/validation end.

In order to handle the challenges and complexities of key management, a number of techniques were proposed and deployed in different cryptocurrencies. In Bitcoin core, the keys are maintained in local storage. A typical user will have an access to a wallet software of his choice and use the same to authenticate transactions he is generating. As wallets generate the digital signature, it requires an access to the private key of the user. While this speeds up the wallet operations, the presence of a key for a long time in a system that is online increases its vulnerability. Off-line storage and air gapped storages are used by systems such as Armory [1]. Password protected wallets are deployed by certain systems but they do not provide any security against a malware that might read the key strokes etc. Third party hosted wallets are also suggested to remove the pains of key management to a novice user but then it requires enormous amount of trust in a third party. A detailed analysis on various techniques that are currently used in practice together with limitations in their usability is reported in [7].

In view of the shortcomings of the existing systems, we take a fresh look at key generation and management using two systems that may be available with a typical user. Our proposal is simple, easy to implement, secure and offers protections against theft/loss of the systems. Given that a typical user may have at his disposal several devices(atleast two, say a laptop