KEKCOIN AND MEMECHAIN

October 1, 2019

ABSTRACT

Kekcoin is a censorship-resistant proof-of-stake (PoS) cryptocurrency designed for the global meme community. It forms an efficient and scalable base-layer infrastructure and currency for a *meme economy*. A second-layer protocol called the MemeChain enables an extensible set of functionalities such as provable content authorship, ownership, and trading. These functionalities can be monetized using smart-contracts. A variety of applications concurrently access MemeChain functionality including a MemeChain desktop browser and the community's telegram chat-bot known as 'Oracle of Kek'. This whitepaper forms the specification for both Kekcoin and the MemeChain and elaborates on key design decisions that have been made to-date.

As a community-driven enterprise with no central governance authority, open-source contributions define the trajectory of this innovative technology at each layer of operation (cryptocurrency, MemeChain protocol, and user-facing applications).

***K*eywords** Cryptocurrency · Second-Layer Protocols · Meme Economy

1 Introduction

Kekcoin is an open project in that it is publicly accessible, transparent, and welcomes anonymous contributions. Kekcoin is global and border-less, it operates the same regardless of which jurisdiction it is accessed from. Kekcoin is neutral, which means the system does not require users to provide a reason why it is being used. Kekcoin is censorship-resistant because the validation network is decentralized and denying access to anyone requires subverting a majority of the stakeholders on the network. Finally, Kekcoin is public, in that each transaction is openly recorded on the system's blockchain, and verifiable by anyone.

Numerous factors limit access to internet content, including strict regulations and inadequate internet infrastructure in developing countries. While internet censorship and surveillance have become prevalent in authoritarian regimes such as China, Saudi Arabia, and North Korea, memes have become a potent tool for disseminating libertarian ideas and for transmitting cross-cultural content. This motivates the need for a border-less and homogenized environment for the dissemination of content. Memes, a common form of transmitted internet content, are a powerful tool that enable a rapid and cross-cultural mechanism for spreading important ideas. This is the motivation that drives the Kekcoin and MemeChain project - meme sovereignty and the digital freedom of speech.

Cryptocurrencies enable a new wave of internet applications where users control their own data. They also have a built-in incentive structure that binds participants together and simultaneously creates an infrastructure layer that acts as an anchor of trust for second layer protocols. Disseminating memes through a protocol based on a decentralized network currency creates an entirely new digital economy, a *meme economy*, protected by smart contracts that intermediate transactions of currency and content.

While Bitcoin [1] is demonstrably the most secure public blockchain for storing digital value, it is not suitable for rapid payments nor micro-payments. It is thus not suitable to form the basis of a meme economy. The question then becomes, how best to approach creating the infrastructure and incentive structure for this meme economy? Solutions based on the lightning network [2, 3], and bitcoin-backed side-chains [4] were considered, but were found to be limited by an important factor. They each required an initial investment from the community before any live demonstration that the technology works. Funds would need to be deposited by each participant before they could begin participating in the nascent meme economy. Rather than take this approach, a new cryptocurrency which supported faster micro-payments was created. An initial supply of that currency was air-dropped to those who wanted to participate, free-of-charge. The market would then determine the value of those tokens based on the collective beliefs on the prospects of the project. This approach served to bootstrap a community without exposing participants to any monetary risk.

The purpose of Kekcoin is not to compete with Bitcoin, rather the intent is for Kekcoin to augment the functionality offered by Bitcoin. Bitcoin has a strong security model which somewhat limits performance, and this enables extremely robust custody solutions for digital value. Kekcoin was constructed with a weaker security model, enabling the performance necessary for the meme economy. By enabling atomic swaps [5] between Bitcoin and Kekcoin, users can be offered the best of both systems, storing the majority of their digital value in Bitcoin, and swapping into Kekcoin when participating in the meme economy.

The rest of this paper is organized as follows. Section 2 describes Kekcoin, the base-layer cryptocurrency which serves as the incentive layer and infrastructure for the meme economy. Section 3 covers the MemeChain protocol design goals and specification.

2 Kekcoin

The Kekcoin-core code is a fork of Bitcoin with alterations in the consensus protocol (detailed in section 2.1) as well as in the blockchain parameters. To better suit its use as a base-layer protocol for the meme economy, Kekcoin has a faster block time than Bitcoin at an average of one block per minute. This places Kekcoin at a different balance in the three-way trade-off (see the CAP theorem [6]) between consistency, availability, and partition tolerance. Kekcoin gains faster consistency by slightly reducing partition tolerance. This enables faster confirmations of transactions, including transactions that relate to MemeChain commands (see Section 3). This comes at the risk of creating more short-term forks at the chain tip, since the propagation time of blocks across the network becomes comparable to network latency in information propagation which may cause network nodes to briefly disagree [7].

At the time of writing, Kekcoin is based on bitcoin-core 0.13, which means it has support for Segregated Witness. This soft-fork activated on the 7th of May 2018 and predominantly fixes transaction malleability issues that hinder off-chain protocol design such as Lightning Network and Atomic Swaps. The next step is to re-base the source code to bitcoin-core version 0.18+. Not only will this bring a lot of new features, it will better prepare Kekcoin for future maintenance of the protocol, since it will be easier to port new updates from Bitcoin Core development.

2.1 Consensus Protocol

A consensus protocol is the mechanism by which (honest) nodes running the same protocol specification maintain a common view of the blockchain state, as a set of heterogeneous participants propose and accept updates to the state in an adversarial context, where some nodes may behave arbitrarily. In particular, the security properties *persistence* and *liveness* should be guaranteed [8]. (Typically there is eventual consistency of state, due to the capacity for short-term forks.)

While Kekcoin is a proof-of-stake (PoS) cryptocurrency (specifically PoS version 3.0 [9]), there was an initial phase where proof-of-work (PoW) [10] was used to generate the supply for the initial air-drop. The choice to rely on PoS as the consensus protocol was made because cryptocurrencies that rely on PoW are subject to attacks from large mining pools for other cryptocurrencies with the same PoW algorithm. Bitcoin as the dominant cryptocurrency needs not worry about such attacks. Further, the arguments for less energy intensive consensus protocols are compelling, at least for use-cases such as creating a vibrant meme economy. The majority of Bitcoin mining operations rely on sustainable energy sources because the intensely competitive mining industry is forcing efficient business models compared with other cryptocurrencies whose mining industries are more nascent.

The security of consensus protocols is a hotly debated topic. On one hand, the use of formal modeling methodologies such as the Universal Composability framework enable reasoning about the security of complex protocols, and have been used to demonstrate security guarantees offered by both PoW [8, 11, 12] and PoS [13, 14] systems. On the other hand, no model can reasonably be expected to perfectly match the complexity of a real protocol implementation and environment. This leads some experts to rely on a system demonstrating its security in practice by continuing to function in the face of myriad attack vectors. Based on both theory and practice, PoS seems a viable consensus protocol for the use-case of Kekcoin.

The four independent processes which lead to the emergence of consensus in open blockchain systems are:

- Independent validation of transactions by each *full node* (one which maintains a full copy of the distributed ledger) in the network
- Independent aggregation of validated transactions by *security providers* (e.g. miners/ stakers), to verify these transactions through proof-of-work (in the case of mining) or proof-of-stake (in the case of staking).
- Independent validation of new blocks by full nodes, and addition of validated blocks into the local copy of the blockchain

- Independent choice, by each node, of the chain that demonstrates the most cumulative cost (and thus the highest security) through proof-of-work and/ or proof-of-stake.

Kekcoin relies on the same validation procedure for transactions as Bitcoin. The difference lies in how aggregations of valid transactions in blocks are verified and validated, which is through proof-of-stake. Finally, Kekcoin nodes consider the chain that demonstrates the most cumulative proof-of-stake to be the reliable state. The security of the system thus depends on internal network activity (i.e. the distribution of stake among participants) as compared with proof-of-work which concretely links security to actions external to the network activity (i.e. the distribution of compute power among participants). This, it could be argued, isolates the attack surface of the consensus protocol from physical and political attacks on security providers (e.g. mining operations being banned in some jurisdictions).

2.2 Crypto-economics

2.2.1 Coin Emission

The total supply of Kekcoin is capped at 21 million units (KEKs). A brief phase of PoW was used to bootstrap the network by enabling a portion of the supply of coins to be offered to the community in an airdrop. The mint rate was chosen to gradually reduce according to the following schedule.

Block Verification Method	Time (Approx.)	Block Range	Reward (KEKs)
PoW	1h 40min	1-100	100000
PoS	7 Days	101-10000	50
PoS	14 Days	10001-20000	25
PoS	21 Days	20001-30000	10
PoS	70 Days	30001-100000	5
PoS	140 Days	100001-200000	2.5
PoS	19 Years	200001-9770000	1
PoS	> 38 Years	> 9770000	0

2.2.2 Initial Distribution

A community-oriented distribution model was designed to align the incentives of the code contributors, content creators, and community participants. A total of 10,000,000 KEKs from the PoW phase were distributed in the following proportions:

An *airdrop* (35%) of funds went to registrants who each received 10,000 free KEKs for demonstrating a unique identity on the popular Bitcointalk forum. Participants used these funds to test the infrastructure, and began tipping each other in the community chat on Telegram.

A *Meme Bounty Fund* (30%) was kept to reward winners of meme bounty competitions which served both as marketing campaigns and as a fair way to distribute funds to potential community members who missed the initial airdrop.

Early contributors (20%) have funded the deployment of the technical infrastructure required to properly secure a global and distributed network. This includes the set-up and continual maintenance of a geographically distributed set of full nodes.

The *development fund* (10%) has been (and will continue to be) used to finance upgrades to the Kekcoin project for the foreseeable future.

The *team and founders* (5%) put in a great deal of effort in setting up, testing and deploying Kekcoin, in addition to building an early-stage community. The Team and Founders Reward is compensation for the work carried out in building the foundations of this meme economy.

Following the initial distribution, the majority of KEKs are minted via the PoS consensus protocol. Early adopters will benefit from being part of the Kekcoin economy while simultaneously securing the blockchain network. Combined, this forms a flexible and efficient ecosystem where network participants and contributors are rewarded on balance while stimulating mass adoption and dissemination via ongoing reward bounties. Kekcoin was listed on public exchanges with KEK/BTC, KEK/LTC and KEK/ETH trading pairs.

3 MemeChain

3.1 Introduction

The MemeChain is a second layer protocol which builds upon the functionality of the Kekcoin blockchain. The idea is to define a type of blockchain transaction that updates the state of the MemeChain. This is called a *meme transaction*, or, MemeTx. In order to optimize data storage and reduce the burden on the performance of the blockchain, the data itself is distributedly stored on the InterPlanetary File System (IPFS) and only a unique identifier for the data is embedded in the MemeTx.

A *MemeChain node* operates as both a Kekcoin node and an IPFS node, and contains additional semantic logic for the MemeChain protocol (e.g. to add a meme to the chain, to read who authored the meme or to transfer the meme to a new owner). The MemeTxs are authenticated commands which update the MemeChain state. The set of commands is extensible to add new functionality to the MemeChain. These commands are time-stamped and support public verification, properties derived from the distributed ledger.

The MemeChain protocol could support any data format, but the current implementation supports a few picture data formats, including JPG, PNG and GIF because the use-case is for memes rather than generic data.

3.2 Design

The MemeChain can be described as a deterministic state-machine which derives its current state from information embedded in the Kekcoin blockchain. It consists of two components; a cryptographic hash-chain of memes that is embedded in the Kekcoin blockchain, and a distributed data store utilising the IPFS network. Inspiration for this design was drawn from numerous sidechain projects, coloured coins, and other layer 2 protocols.

MemeChain nodes establish consensus about the state of the MemeChain by ensuring that the methods for both injecting and extracting MemeChain data from the blockchain are deterministic. That way, it is trivial to ensure that each MemeChain node is interacting with the same view of the state. This removes uncertainty that may arise in a non-deterministic set-up, and removes the need for MemeChain nodes to converse with each other about anything other than the state of the blockchain itself, which is a task that is already completed through the Kekcoin core protocol.

3.2.1 IPFS Integration

IPFS is a new hypermedia storage and transfer protocol which will form part of the “permanent web”. IPFS comprises a peer-to-peer network which redundantly stores and shares data packets. The redundancy enables permanent hosting of content. Navigating the content hosted on IPFS is made trivial through the content-addressable method of storing the data in a distributed file system.

Utilizing the resilient IPFS network, consistent availability of memes is ensured. That is to say, memes that pass protocol validation and are in the MemeChain state will remain there indefinitely. Moreover, IPFS enables high volumes of data to be distributed with high efficiency. A MemeChain node will automatically run an IPFS node and thus contribute to the storage and sharing service of the network.

3.2.2 Hash-chain

The hash-chain is created using a special kind of blockchain transaction, one with an OP_RETURN script. This is a script which allows users to take advantage of the immutability and irreversibility of the blockchain. The script enables 80 arbitrary bytes of data to be embedded in the blockchain while rendering the transaction unspendable. In order to create a hash-link between MemeChain data in the blockchain, a specific data format will be defined. To illustrate an example of the formatting required, consider the following:

|–Identifier–|–Command code–|-arguments-|–Sha256 Hash(current meme hash+prev meme hash)–|

With 80 bytes in total, the first bytes are allocated as an identifier for the MemeChain parsing module. This is followed by a few bytes which indicate a specific command whose semantics are defined by the MemeChain protocol. This is designed to be extensible with new commands added in future upgrades as new use-cases are demanded by different apps using the protocol. The next string of bytes form the argument for the command. If the command is to add a meme, then the argument is the hash of the transaction’s meme (the same hash as is used on IPFS). The final string of bytes consists of the hash of the concatenated hashes of the current meme and the previous confirmed meme. This final string is what links the MemeChain transactions and thus constitutes the building block for the hash-chain. A transaction with this formatting is a MemeTx.

Due to the blockchain property of eventual consistency, MemeChain nodes will only confirm blocks of transactions once they are at a certain depth in the chain. This is because, in theory (albeit with low probability), the latest blocks are subject to change through a brief fork as staking nodes concurrently hold a different view about the chain tip. For similar reasons, a new MemeChain transaction must not be hash-linked with a previous MemeChain transaction which has yet to be confirmed. It is important to note that the chain will form a directed acyclic graph (DAG) rather than a single sequence of memes.

3.2.3 MemeChain API

The MemeChain software runs as a daemon (a background process) and handles the complexity of MemeChain transactions and IPFS content management. The interface that connects the MemeChain daemon with any generic web application is in the form of an API. A web application is able to run their own MemeChain node, with local client HTTP API support, or it will be able to connect to a remote client which is exposing their API links publicly.

Any user or web app running the MemeChain software can share the same view of the MemeChain, and multiple applications update this state concurrently. An emphasis is put on making the API simple so that any developer can easily integrate the MemeChain into their software. Examples are provided on the project's Github to showcase how the API is used and integrated in various different use cases and applications. Applications built on top of the MemeChain are encouraged to remain open-source.

The current set of MemeChain API commands includes:

API Call	**Argument**
Get Info	/
Get MemeChain Height	/
Get Meme Data by Height	Height
Get Meme Data by Range	Range
Get Meme Data by Hash	Hash
Get Meme Image by Height	Height
Get Meme Image by Hash	Hash
Add Meme	Meme

3.2.4 Incentives for MemeChain nodes

The incentives for running a Kekcoin node are robust - stakers earn rewards for processing transactions. The incentive structure for MemeChain nodes must be extended to ensure that files are adequately supported on IPFS.

Since there is no custom peer-to-peer communication protocol for MemeChain clients, creating a protocol that relies on proofs-of-storage and proofs-of-replication is not feasible. Instead, a simpler approach must be taken.

Initially, app developers will be forced to set-up sufficient infrastructure to maintain the distributed data stores for the set of memes in the MemeChain. They are incentivised to do this since, if they do not, their apps will not run.

Additionally, the apps can be configured to run MemeChain nodes by default. While users could always modify their app to not support IPFS file-hosting, this would take some technical knowledge and effort, which acts as a natural dis-incentive.

It remains an open and active research question of how to incorporate stronger incentives into the system. Time will demonstrate whether what has been proposed so far is sufficient.

3.2.5 Illegal Content Management

The purpose of the MemeChain is not to persistently host illegal media. On the contrary, a mechanism should be in-place to enable any MemeChain participant refrain from implicating themselves in a crime.

Content filters should be applied by any MemeChain node who is publicly exposing their API, so as not to implicate themselves by hosting incriminating content. It remains an open question how best to approach filtering content. Automated filters can be used as a first line of defence, while manual filters are reserved for anything that bypasses the automated filter.

MemeChain nodes should also be weary of other MemeChain nodes adding illegal content to the MemeChain state, and should pass any content not authored by them through their filtering process while syncing with the MemeChain state. If incriminating content is discovered, ideally it should never be saved and should be dropped immediately.

If some illegal content makes its way through all filters in use by a MemeChain node who is hosting a user-facing application, the users may be confronted with the content. Users should have a mechanism to report this content and the application host should handle this by immediately hiding the content from all users until it can be manually filtered. To ensure this reporting mechanism is not abused, multiple reports could be required from different users.

Given this construction, any MemeChain node has sovereignty over the data they store. At the same time, they have full responsibility for the content they expose to other clients through apps they are hosting. Content can be censored locally and independently, but to achieve global censorship requires each MemeChain node to filter the content.

References

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *https://bitcoin.org/bitcoin.pdf*, (Accessed Online October 1, 2019), 2008.

[2] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9212:3–18, 2015.

[3] Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. *https://lightning.network/lightning-network-paper.pdf*, (Accessed Online October 1, 2019), 2016.

[4] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains, 2014.

[5] Maurice Herlihy. Atomic Cross-Chain Swaps. *http://arxiv.org/abs/1801.09515*, (Accessed Online October 1, 2019), 2018.

[6] Eric A. Brewer. Towards robust distributed systems (abstract). In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*, PODC '00, pages 7–, New York, NY, USA, 2000. ACM.

[7] C Decker and R Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10, Sept 2013.

[8] The Bitcoin backbone protocol: Analysis and applications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9057:281–310, 2015.

[9] Proof-of-stake protocol v3.0. `https://forum.bitbay.market/uploads/default/original/1X/a82b35edf21dee2cded2624d82fad28e6c1f4682.pdf`.

[10] Adam Back. Hashcash - A Denial of Service Counter-Measure. *http://www.hashcash.org/papers/hashcash.pdf*, (Accessed October 1, 2019), 2002.

[11] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the Blockchain Protocol in Asynchronous Networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673, Cham, 2017. Springer International Publishing.

[12] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 291–323, Cham, 2017. Springer International Publishing.

[13] Ouroboros: A provably secure proof-of-stake blockchain protocol. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10401 LNCS:357–388, 2017.

[14] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 66–98, Cham, 2018. Springer International Publishing.