BigTangle Technical Whitepaper

Kai Cui, Jianjun Cui,
Maximilian Hensel, Maximilian Lowin

April 28, 2018

Abstract

Recent cryptocurrencies have tackled a number of key issues preventing widespread adoption and practical usage, one of which is the scalability issue. In this technical paper, a novel big data approach to scalable cryptocurrencies employing state-of-the-art big data technologies such as Apache Spark and Apache Kafka in combination with the highly scalable Tangle DAG base architecture is proposed. This feeless and scalable Proof-of-Work approach with mining is projected to solve issues of other recent financial cryptocurrency solutions by providing inter alia native support for token issuances, resulting in attractive use-cases for banks, stock exchanges and companies and achieving high performance in terms of transaction throughput by parallelizing computations on a directed acyclic block graph instead of a blockchain.

I. Introduction

Since the inception of Bitcoin in 2009 [4], new cryptocurrencies have been rising in popularity and adoption rate and are predicted to potentially rise to rival fiat currencies in the future. Transitioning to such a digital currency offers various advantages, of which the authors believe the infrastructure cost savings in the financial sector via revolutionizing highly complex and opaque traditional transaction systems from current banking and stock market technologies to be the most attractive. To attain this attractivity, two key properties are required: scalability and the absence of fees in general. Additionally, low transaction confirmation times, ease of use, security and anonymity are other baseline properties required for widespread adoption. Traditional Proof-of-Work-based blockchain approaches fail to meet these requirements, while Proof-of-Stake-based approaches might face other problems such as voter politics, mining regulation etc.

In this paper we propose BigTangle, an alternative scalable Proof-of-Work-based cryptocurrency to recent Proof-of-Stake-based blockchains. The proposed solution's architecture extends the Tangle [1], a DAG generalization of blockchains capable of scaling to infinite transactions per second while remaining feeless. Beyond the decentralization of payment processing, the network can be utilized as a basic service layer for the decentralization of markets in general, transfer and ownership management or authenticity proofs for assets of any kind. It supports all functions supported by Bitcoin, including but not limited to escrow transactions, bonded contracts, third-party arbitration, multiparty signatures etc.

In contrast to most other recent high performance cryptocurrencies, BigTangle employs a Proof-of-Work-based mining process where in principle every transaction is rewarded and helps securing the network. This mining process incentivises users to run full nodes and secure the network, providing advantages such as inflationary currency models and value through economic coupling to the real world. Full node operators and miners are therefore compensated with a time-normalized amount of new BigTangle tokens proportional to their contribution to network validation.

BigTangle begins as a fully decentralized solution which does not rely on a centralized coordinator to finalize consensus and sees itself as a successor to Bitcoin, focusing on economically important key use-cases and differing from other current Tangle implementations in a number of key design points as mentioned further below. Projected practical use cases include payment processing, decentralized stock exchanges, supply chains and product integrity tracking.

In the authors' opinion, a transaction system capable of supporting the global transaction volume of banks and stock markets requires sufficiently performant computer clusters to work as full nodes. BigTangle will therefore employ established industrial grade big data technologies such as Apache Kafka, Phoenix and Spark. By utilizing local approximations to previously non-scalable graph computations of the asynchronous Tangle in addition to big data computation technologies, it is believed that the proposed solution suffices the properties mentioned before. Minimal end-user client requirements are a side effect, allowing end-users to participate and issue transactions without having to store and handle huge amounts of data but instead providing hashing power.

II. Overview

The protocol maintains a public ledger of transactions which are contained within blocks. In contrast to blockchain-based solutions, blocks reference and approve two previous blocks to form a directed acyclic graph (DAG). As per Tangle protocol, a block is valid if all referenced blocks are valid and no conflicts exist in the set of the block itself and all of its approved blocks. Contributing blocks therefore helps securing and validating the Tangle by approving all referenced blocks. A detailed explanation of the base architecture and important existing concepts can be found in [4], [1] and [2].

Participants are connected in a standard peer-to-peer network via a gossip protocol as fall-back solution. In addition, BigTangle will employ Apache Kafka data streaming to achieve sufficient scalability and propagation speed. The network is split into two different archetypes: clients and network nodes. Generic end-users and miners can participate as clients that issue transactions with the aid of a network node, using their hashing power to create blocks and solve the low-difficulty Proof-of-Work themselves, while network nodes maintain a copy of the graph and provide validated tip pairs to build upon.

The network nodes can derive account balances and transaction states from the graph and provide the maintained states to clients in exchange for e. g. hashing power. Furthermore, a network node operator might choose to also participate in the mining process himself. This consists of using available computational power to solve Proof-of-Works for new (possibly empty) blocks.

The server-side validation as described in [1] can be effectively computed by utilizing local approximations. The network nodes create their local view of the Tangle by maintaining helper constructs such as local confirmed block snapshots or UTXO tables. Validation is then performed by building upon this snapshot. This means that additional validity constraints are added over the standard Tangle validity constraints, resulting in a stricter validity evaluation than before. Additional information can be found in chapter 3.

Baseline properties provided by BigTangle are high hash power security due to mining, infinite scalability, sufficiently fast transaction confirmation times, full decentralization, trustlessness and permissionlessness, feeless transactions, in-principle quantum security and normalized inflation. To make use of these properties, BigTangle will natively support custom token issuances and decentralized token exchanges.

Popularly quoted future solutions to scalable cryptocurrencies include highly complex sharding, voting-based and capital-based Proof-of-Stake or workarounds such as the lightning network [3] with the potential for exorbitant fees. Instead, we propose a fully permissionless solution that returns to the well-known Proof of Work approach of the original Bitcoin solution. Mining rewards serve as a network maintenance incentive instead of fees. Since it is impossible to allow every device to participate equally as network nodes while at the same time achieving infinite scalability without employing highly complex sharding technologies, it is intended that network nodes are deployed in sufficiently big computer clusters utilizing state-of-the-art big data technology. Instead, clients simply cooperate with network nodes by providing mining revenue and building blocks as needed.

The network stays permission-less and avoids all centralized constructs such as coordinators. In principle, as long as nodes fulfill the minimum requirements for keeping up with the transaction volume, any full nodes can participate in the network. Regardless, it is intended for very many full nodes to exist regardless of mining rewards. For example, super market chains or banks can deploy their own full nodes to process their transaction volume independent of other full nodes. By utilizing Proof-of-Work, we can avoid the typical pitfalls of Proof-of-Stake-based models such as missing mining accessibility, political apathy and higher capital volatility.

III. Technical Details

In the following, we briefly discuss technical key details and their realization. For visualizations of established concepts, please refer to existing literature. More detailed generic specifications are omitted at this point.

i. Implementation

To achieve high scalability, the node implementation is built upon a Bitcoin implementation and industry standard big data technologies, including but not limited to Apache Phoenix, Kafka and Spark GraphX.

ii. Cryptographic Components

BigTangle currently relies on elliptic curve cryptography for multi-use signatures. The curve used is Bitcoin's Secp256k1. Similarly, the public key hashes are also Base-58 encoded, allowing Bitcoin users to reuse their addresses in BigTangle. The proposed Proof-of-Work hashing function is the currently ASIC-resistant Equihash algorithm. Except for the signature scheme, the base architecture is quantum resistant as shown in [1]. In case of breakthroughs in quantum computing, new signature schemes can be added.

iii. Transactions

The accounting is based on Bitcoin's Unspent Transaction Output (UTXO) model. Users can issue valid transactions as long as they can provide a valid input script for the used UTXOs. The UTXOs use Bitcoins not Turing-complete stack language. thereby allowing for the same set of functionality as found in Bitcoin.

iv. Blocks

A block consists of its header and transactions. In addition to fields existing in Bitcoin, the header contains an additional reference to a previous block, a miner address, a type field for the types as seen below and additional data depending on their type. We briefly list all three basic block types:

Default Blocks contain default transactions. The majority of mining blocks are empty under the assumption that miners have no transactions to perform.

Mining Reward Blocks contain mining reward transactions that are computed locally on demand. They are used in the mining process as detailed later.

Token Issuance Blocks are used to issue custom tokens. Tokens are identified by a public key hash and issuances are legitimized by the corresponding private key signatures. They contain at least one token issuance transactions of the corresponding token.

v. Participants

Participants can take on different roles in the network depending on their available resources and intentions. In the following, possible types of participation are ordered in descending requirements of bandwidth, space and computational power:

Full Nodes keep a copy of the full Tangle. They can fully participate in the network and provide any requested blocks.

Pruning Nodes maintain a pruned version of the Tangle. Only the most recent blocks in terms of confirmation are kept in storage. The node can still fully participate in the mining process unless a highly unlikely reorganization event happens.

Clients do not keep a copy of the Tangle. They rely on the nodes to provide them with necessary information to create transactions and blocks. Clients can solve the proof of work of their issued transactions and blocks and provide incentives for network nodes to assist them.

vi. Protocol Details

Node Maintenance

In the following, we briefly describe assorted technical details required for preparing the Tangle base architecture to achieve scalable operation. As mentioned before, the BigTangle node implementation locally maintains a snapshot called the milestone, a set of blocks it considers as confirmed and thereby in principle finalized. In addition, it maintains additional auxiliary information and block evaluations such as rating, cumulative weight, depth, height etc. as found in [1]. The milestone update process in simplified form consists of the following steps and is performed as often as possible:

1. Ingest new blocks such that an unbroken Tangle is obtained.
2. Update relevant block evaluations.
3. If needed, remove now unconfirmed blocks and dependents from milestone.
4. Find new locally confirmed blocks, resolve conflicts and add to milestone.

We consider as locally confirmed any block that has reached the upper confirmation threshold in terms of rating and is sufficiently deep. We also add a hysteresis to removing blocks to prevent unnecessary reorganizations.

To ensure that the honest miners always find a consensus, we set the lower confirmation threshold to two-thirds as shown in chapter 4. For stability purposes, we introduce a hysteresis and set the upper confirmation threshold to 70%

Of particular note is the conflict resolution procedure. It should only find application in step 4 if malicious nodes approve conflicting block combinations such that conflicting blocks are considered locally confirmed. Nonetheless, this case must be handled correctly and is also used during the tip selection algorithm. In short, we process conflicts in descending order of maximum rating occurring in the conflicts, eliminating all losing milestone candidates by removing them and all their approvers or dependents from the milestone or candidate list.

Lastly, we may prune no longer relevant blocks and their evaluations to prevent the Tangle from growing indefinitely in terms of storage space. For example, blocks in the milestone and their conflicting counterparts could be pruned after reaching a combination of sufficient depth, age of confirmation etc.

Validation and Approval Selection

When generating a new block, we require two previous blocks to reference such that no conflict exists in the union of referenced blocks. To find such conflict-free block pairs, we first apply an MCMC algorithm similar to the approach shown in [2] to find single tips. We then resolve conflicts according to the conflict resolution procedure detailed before and reverse on the path taken until a block consistent with the milestone is found. Finally, we repeat an analogous conflict resolution procedure for a pair of obtained blocks, resulting in two blocks that are conflict free and consistent with the current local milestone.

A full list of generic validation constraints (e. g. no conflicting token issuances, no conflicting mining rewards) is omitted at this point.

Mining Process

To incentivise node operation and network maintenance, we introduce a mining process quite similar to the Bitcoin mining process. Since the Tangle is an asynchronous network and every node sees a different version of the Tangle, we cannot simply reward what is seen locally since we wish for an approximately fixed inflation rate. Instead, we must introduce a fix point such that every node can calculate the same rewards in a normalized manner.

We divide blocks by height intervals and issue mining reward blocks to reward intervals. All blocks referenced by the mining reward blocks in the respective interval are considered for compensation and the mining reward block must therefore be in conflict with other such blocks of the same reward height interval.

Using only the blocks approved by the mining reward block, we can compute consistent rewards since we know the referenced subgraph to be unbroken in order for the mining reward block to be considered for confirmation. The calculation of rewards is then done locally and on confirmation only to prevent network spam.

One key problem is how to approve mining reward blocks that are consistent and fair according to the nodes local Tangle state. The solution is to use the following validity constraints for mining reward blocks: All of the approved blocks of the specified height interval must be in the milestone and most of the milestone blocks of the specified height must be referenced at the time of reward block reception. To prevent deadlocks due to local inconsistencies, we must also allow this constraint to be overridden by e. g. sufficient rating and depth, as in that case the rest of the network has accepted the block as valid.

Additionally, we must include countermeasures against non-validating miners' attacks as shown further below. For example, we can punish the blocks with the lowest cumulative weight in their specific height as seen from the fix point view, since attackers will pursue a suboptimal tip selection algorithm to avoid validating new blocks and therefore in general gain less cumulative weight.

Now it becomes apparent why we use the fix point approach: Since we need to punish non-validating miners' blocks, it is easier and more consistent to detect local consistency of referenced blocks at reception from a fix point view than it is to evaluate the varying inconsistency levels of cumulative weight.

Lastly, a per-transaction reward for the next interval is calculated analogously to Bitcoin's difficulty adjustment: By enforcing a monotonous increase of block timestamps and limiting the validity of timestamps from the future, the per-transaction reward is adjusted according to the current transaction rate to enforce an approximately constant coin emission rate of e. g. 2% of total supply p.a.

Token Issuance Process

The token issuance process is trivial: We can simply legitimize a new token issuance block by signing their transactions with the token's corresponding private key. A token is identified by its ID in form of a public key hash. The first time issuance can be configured to e. g. disallow repeated issuances and establishes any other custom token rules.

IV. Attack Mitigation

We assume that at least two-thirds of the blocks are honest and validate correctly. For up to one-third of malicious hashing power, we assume that while hashing power does not directly correlate with rating influencing, only up to one-third of the rating is hijacked by the attackers for most of the relevant blocks. It is likely that the percentage of hijacked rating tips would be significantly lower, and further mitigation can be provided by including older snapshots into the rating calculation algorithm. As such, we must ensure that the honest miners still find a consensus:

- In case percentage x of rating tips maliciously approve double spends, we must ensure that no reorganization occurs, meaning that the lower threshold must be below $(1-x)$.

- In case percentage x of rating tips maliciously approve a conflict, we must ensure that no network split occurs, meaning that the lower confirmation threshold must be above $(\frac{1-x}{2}+x)$ to prevent one half of the honest hash power seeing a different version than the other half.

It is obvious that the maximum possible x for which such a lower confirmation threshold exists is one-third. The corresponding lower confirmation threshold therefore is two-thirds.

Other attacks include parallelizing Proof-of-Work to accumulate the highest cumulative weight, which is mitigated by the probabilistic nature of MCMC as well as network latency and milestone update rate in general being slower than Proof-of-Work computations.

Not validating any transactions runs the risk of wasting hashing power, while building your own subgraph by approving your own blocks only will lead to high orphaning risk due to introducing more than the optimal amount of transitions on your approved blocks.

Trying to relink a pre-built subgraph of higher height to circumvent gaining less cumulative weight is mitigated by penalizing high height difference transition probabilities.

For other attack vectors such as double spends and Tangle game-theory, please refer to [2] and [1].

V. Practical Use Cases

As mentioned before, projected practical use cases the token derives its value from mostly include substitution of various currently costly and trust-based technical processes. In the following, use cases are briefly explored.

i. Payment

An obvious main use case is payment processing. By providing a scalable infrastructure, BigTangle enables the global transaction volume to be processed in one network. Most importantly, this offers infrastructural cost advantages by eliminating complex and costly processes of traditional payment processing for banks, companies and the general populace. Note that the network hashing power is approximately proportional to the BigTangle internal token market cap and is therefore decoupled from actual transaction volumes, theoretically resulting in downwards unbounded energy upkeep at the cost of increased confirmation times. At the same time, adequate confirmation times scaling logarithmically with the transactions per second can be achieved [1].

ii. Fiat Money

For banks, the token issuance protocol can be used to issue bank-backed tokens denoting conventional fiat money. Since the issuance in principle requires little to no participation in keeping up the network, BigTangle is a low cost solution for all parties. Fiat money transactions can then feasibly be processed within seconds on a global scale.

iii. Stock Markets

Markets for stocks, bonds etc. can easily be realized by creating new token equivalents. Companies can publish stocks and use the BigTangle network, essentially substituting costly stock exchange processes by the feeless BigTangle processing network.

Examples for the largest segments that will be affected: Bonds, Swaps, Derivatives, Commodities, Unregistered/Registered securities, Over-the-counter markets, Collateral management, Syndicated loans, Warehouse receipts, Repurchase markets etc.

iv. Micro Transactions

Service fees can now be charged in microdollar range or alternatively via seconds of hashing power due to the departure from winner-takes-it-all, allowing for new business models, e. g. online newspapers with alternatives to commercial advertisement.

v. Supply Chain

Under the assumption of trustworthy suppliers issuing authenticity tokens, it is trivial to track product authenticity via token transfers. This use case extends into classic supply chain management as well, allowing the trustless tracking of inventories in supply chains.

VI. Future Investigations

i. Privacy

There are a multitude of arguments for and against private transactions. Nevermind the arguments, intransparency is an interesting extension to Tangle and will be investigated after release.

ii. Smart Contracts

Although in the authors' eyes, Turing-complete smart contracts are no requirement for achieving the main goal of infrastructure cost savings, more generalized smart contracts for Tangle will be subject of future investigations.

References

[1] Popov, S. (2016). The tangle. https://iota.org/IOTA_Whitepaper.pdf

[2] Popov, S., Saa, O., & Finardi, P. (2017). Equilibria in the Tangle. arXiv preprint arXiv:1712.05385.

[3] Poon, J., & Dryja, T. (2016). The bitcoin lightning network: Scalable off-chain instant payments.

[4] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.