BigTangle Whitepaper

Kai Cui, Jianjun Cui,
Maximilian Hensel, Maximilian Lowin
kai.cui@bigtangle.net

June 11, 2018. Version 1.1

Abstract

BigTangle is a protocol for the internet of value. It is a cryptocurrency network extending existing directed acyclic graph architectures with MCMC as consensus algorithm. Through the use of industry-grade big data technology in conjunction with its parallelizable architecture, BigTangle sees itself as a successor to Bitcoin that can fulfill economically important key use-cases. Proof-of-Work mining and custom token self-issuances are supported. Key Features: Ease of Use, Completely Feeless, Real-Time Confirmation, Infinite Scalability, Permissionless, Trustless, Decentralized, Distributed Proof of Work and Quantum Security.

I. Introduction

Since the inception of Bitcoin in 2009 [1], new cryptocurrencies have been rising in popularity 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 from revolutionizing highly complex and opaque traditional transaction systems of 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, a scalable Proof-of-Work-based cryptocurrency, as a protocol for the internet of value. The proposed solution's architecture is a directed acyclic graph (DAG) generalization with MCMC as consensus algorithm [10] of blockchains that is capable of scaling to infinite transactions per second while remaining feeless. Beyond the decentralization of payment processing, the network can be used as a base service layer for the decentralization of markets in general, transfer and ownership management, authenticity proofs for assets of any kind etc. It supports all features supported by Bitcoin, including but not limited to escrow transactions, bonded contracts, third-party arbitration, multiparty signatures and so on. The Bitcoin blockchain is simply a special case in BigTangle's DAG architecture. Using the mining reward process detailed later, it is possible to change the BigTangle parameters to return to a conventional blockchain. The protocol includes different block types with important use cases in mind, take for example the storage block and virtual OS block. These are a general extension of the DAG architecture that make other smart contract blockchains a special case in the context of BigTangle as well.

In contrast to most 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 mining contribution.

BigTangle sees itself as a successor to Bitcoin, focusing on economically important key use-cases. 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, Spark and its GraphX API. 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 that 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 [1], [10] and [12].

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 [10] can be effectively computed by utilizing approximations. The network nodes create their local view of the Tangle by maintaining helper constructs such as local confirmed block snapshots called milestones. Validation is then performed based on this snapshot. The consequence is that additional validity constraints are added over the standard Tangle validity constraints, resulting in a stricter validity evaluation than without this approximation. Additional information can be found in chapter 3.

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

Popular solutions to scalable cryptocurrencies include highly complex sharding, voting-based and capital-based Proof-of-Stake or workarounds such as the lightning network [2] 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 of end-users hosting full network nodes and requiring significant computational resources, their clients simply cooperate with network nodes by e.g. providing mining revenue and building blocks as needed.

The network stays permissionless and avoids all centralized constructs. As long as nodes fulfill the minimum requirements for keeping up with the transaction volume, any node can effectively participate in network validation and mining. Independent from mining, it is intended for 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 big transaction volume themselves. By utilizing Proof-of-Work, we can avoid drawbacks of Proof-of-Stake-based models such as missing mining accessibility, political apathy, regulations and increased volatility.

III. Technical Details

In the following, we briefly discuss technical key details and their realization. For further information on established concepts, please refer to existing literature.

i. Implementation

To achieve high scalability, the node implementation is built upon Bitcoin and industry standard big data technologies, including but not limited to Apache Kafka [35] and Spark GraphX [34][33].

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 [3]. The base architecture is quantum resistant as shown in [10]. In case of breakthroughs in quantum computing, new signature schemes and hashing functions can be added.

iii. Transactions and Accounts

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 list all base block types and their use:

Transfer Blocks contain transactions intended to transfer value from one owner to another.

Mining Reward Blocks contain mining reward transactions that are computed in a deterministic fashion. More on this in the mining process detailed later.

Token Issuance Blocks are used to issue custom tokens. Tokens are identified by address plus sequence number and issuances are legitimized by the corresponding private key signatures. They contain at least one issuance transaction of the corresponding token.

Cross Chain Blocks are used for interconnectivity between other blockchains. This enables the connectivity of other blockchain systems with BigTangle.

Storage Blocks are used to store user data. User data is identified by address and usage is legitimized by the corresponding private key signatures and the user data can be encrypted. User data is treated as a value object and can be transferred and traded. The BigTangle Mainnet limits the size of the storage. We thereby create an application layer storage network based on pay for use by using the same technology.

Virtual OS Blocks are used to create a virtual distributed Operation System for decentralized autonomous corporations, smart contracts and any other distributed applications. The distributed applications can be implemented in any modern language and are not limited to using specialized languages. The relevant code and state data is saved in the block as VM containers using technology such as Docker Composer [30] or Kubernetes Containers [32]. The execution of VOS blocks simply changes the state data and creates new blocks. As an example, the Mining Reward Process in BigTangle is implemented using this VOS and all nodes will execute the same computation for mining rewards based on the current data in BigTangle. As another example, the market exchange application is implemented in such a form that it requires only a single node for the execution. As the system provides the option to set n-m execution, there is a trade-off between trust, security and performance.

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.

Pruned 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 [10]. 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 [12] 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 validation constraints 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.

Finally, 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 with inflation.

Token Issuance Process
We legitimize a new token issuance block by signing their transactions with the token's corresponding multiple private keys. A token is identified by its ID in form of a public key hash and sequence number. The issuances can be configured to e.g. disallow further issuances and enforce other custom token rules. Any issuances following another issuance must adhere to the previous issuance's defined rules, e.g. multi-signature checks: the created tokens must be signed by the given number of keys to spent the tokens. It is designed to allow for safe usage.

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 hijacked rating tips, only up to one-third of the rating is hijacked by the attackers for most of the relevant blocks. Further mitigation can be provided by using a low-pass filter with older snapshots for rating calculation.

- In case percentage x of rating tips maliciously approve double spends, we must ensure that no reorganization occurs, meaning that the lower confirmation 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 the network from having conflicting blocks between their milestones. (network split)

The maximum percentage for which such a

lower confirmation threshold exists is one-third. The corresponding lower confirmation threshold is therefore set to two-thirds to ensure that the honest miners still find a consensus most of the time.

Other attacks include parallelizing Proof-of-Work to accumulate the highest cumulative weight and in turn gaining more rewards, 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 economic risk of building invalid blocks, 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, while 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 generic attack vectors such as double spends and Tangle game-theory, please refer to the literature.

V. Practical Use Cases

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

i. Payment

A simple main use case is payment processing. By providing 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 for constant economic risk. Adequate confirmation times scaling logarithmically with the transactions per second can be achieved [10].

ii. Fiat Money

For banks, the token issuance protocol can be used to issue bank-backed tokens denoting conventional fiat money. Since the issuance and usage requires almost no participation in 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. Regardless, private transactions are an interesting extension to Tangle and will be investigated after release.

References

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

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

[3] Biryukov, A., & Khovratovich, D. (2017). Equihash: Asymmetric proof-of-work based on the generalized birthday problem. Ledger, 2, 1-30.

[4] People on nxtforum.org (2014). DAG: a generalized blockchain. `https://nxtforum.org/proof-of-stake-algorithm/dag-a-generalized-blockchain/`

[5] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, Aviv Zohar (2012). On Bitcoin and red balloons. Proc. 13th ACM Conf. Electronic Commerce, 5673.

[6] Sergio Demian Lerner (2015) DagCoin: a cryptocurrency without blocks. `https://bitslog.wordpress.com/2015/09/11/dagcoin/`

[7] Yonatan Sompolinsky, Aviv Zohar (2013) Accelerating Bitcoins Transaction Processing. Fast Money Grows on Trees, Not Chains. `https://eprint.iacr.org/2013/881.pdf`

[8] Yoad Lewenberg, Yonatan Sompolinsky, Aviv Zohar (2015) Inclusive Block Chain Protocols. `http://www.cs.huji.ac.il/~avivz/pubs/15/inclusivebtc.pdf`

[9] Sheldon M. Ross (2012) Introduction to Probability Models. 10th ed.

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

[11] Gilles Brassard, Peter Hyer, Alain Tapp (1998) Quantum cryptanalysis of hash and claw-free functions. Lecture Notes in Computer Science 1380, 163169.

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

[13] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. arXiv preprint arXiv:1612.02916, 2016.

[14] Miguel Correia, Nuno Ferreira Neves, and Paulo Verssimo. From consensus to atomic broadcast: Time-free byzantine-resistant protocols without signatures. The Computer Journal, 49(1):8296, 2006.

[15] Christian Decker, Jochen Seidel, and Roger Wattenhofer. Bitcoin meets strong consistency. In Proceedings of the 17th International Conference on Distributed Computing and Networking, page 13. ACM, 2016.

[16] Christian Decker and Roger Wattenhofer. Information propagation in the bitcoin network. In 13th IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy, September 2013.

[17] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Nicolas Gailly, Ismail Khoffi, Linus Gasser, and Bryan Ford. Enhancing

bitcoin security and performance with strong consistency via collective signing. In 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016., pages 279296, 2016.

[18] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. The honey badger of bft protocols. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pages 3142. ACM, 2016.

[19] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In International Conference on Financial Cryptography and Data Security, pages 507527. Springer, 2015.

[20] Vitalik Buterin. 2014. Slasher: A Punitive Proof-of-Stake Algorithm. (January 2014).

[21] Jinyuan Li and David Mazières. 2007. Beyond One-third Faulty Replicas in Byzantine Fault Tolerant Systems. In Proceedings of the 4th Symposium on Networked Systems Design and Implementation.131144.

[22] Leslie Lamport. 2011b. Byzantizing Paxos by Refinement. In Proceedings of the 25th International Conference on Distributed Computing. 211224.

[23] David Schwartz, Noah Youngs, and Arthur Britto. 2014. The Ripple Protocol Consensus Algorithm. (2014). https://ripple.com/files/ripple_consensus_whitepaper.pdf

[24] Protocol Labs, 2018. Filecoin: A Decentralized Storage Network. https://filecoin.io/filecoin.pdf

[25] block.one, 2018. EOS.IO Technical White Paper v2. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md

[26] ethereum, 2017. A Next-Generation Smart Contract and Decentralized Application Platform- https://github.com/ethereum/wiki/wiki/White-Paper

[27] Colored coins whitepaper, https://docs.google.com/a/buterin.com/document/d/1AnkP_cVZTCMLIzw4DvsW6M8Q2JC0lIzrTLuoWu2z1BE/edit

[28] Secure property titles with owner authority: http://szabo.best.vwh.net/securetitle.html

[29] Mastercoin whitepaper: https://github.com/mastercoin-MSC/spec

[30] Docker: https://www.docker.com

[31] BigTangle User Guide: https://bigtangle.net

[32] Kubernetes Container Orchestration. https://kubernetes.io/

[33] Apache Spark. Unified analytics engine for large-scale data processing. https://spark.apache.org/

[34] Apache Hadoop. Reliable, scalable, distributed computing. http://hadoop.apache.org/

[35] Apache Kafka, distributed streaming platform. https://kafka.apache.org