The Burst Dymaxion

An Arbitrary Scalable, Energy Efficient and Anonymous Transaction Network Based on Colored Tangles

Seán Gauld[1]*, Franz von Ancoina, Robert Stadler[2]

Abstract
We describe the concept and implementation of a proposed layer to the Burstcoin cryptocurrency called *"The Burst Dymaxion"*. This layer implements an arbitrary scalable, energy efficient and anonymous transaction network based on colored tangles. Tangles are DAGs that can be seen as generalization of the blockchain. Coloring is a simple tagging technique used in cryptocurrencies to allow for coexistence of various instances of a class in a common data context. The Burstcoin network has several unique properties that make it our premier choice for implementation: its Proof-of-Capacity (PoC) consensus algorithm is energy efficient, fair and anti-centralistic, the blockchain is very compact and implements data scrubbing to reduce blockchain bloat and it has smart contracts. The coin has over 3 years of heritage and a community to bootstrap with, yet it is small enough to allow for significant improvements without a long and paralyzing scaling debate. The original Burstcoin blockchain is used as underlying layer to open and close an arbitrary number of general purpose transaction channels, similar to the Lightning Network proposal of Bitcoin, but using IOTA-like tangles for propagation and verification. Each of these channels is not limited in terms of capacity and number of transactions. This concept takes the best traits of the original Burstcoin, IOTA, Monero, ZCash and the newest Bitcoin proposals to create a currency suitable for truly global use. It allows e.g. banks, clearing houses, remittance processors and virtually all other market participants to use quasi-private, yet decentral and trustless transaction channels with desired properties.

Keywords
Burstcoin – DAG – Dymaxion – Lightning-Network – PoC – Ring-Signature – Scaling – Tangle – zk-SNARK

[1] *PoC Consortium - a CryptoGuru SIG*
[2] *Stronzo Bestiale Institute of Technology*
***Corresponding author**: dymaxion@cryptoguru.org

Contents

Introduction 1
1 Building Blocks 3
1.1 Burstcoin Blockchain . . . 3
1.2 Smart Contracts . . . 3
1.3 DAGs and The Tangle . . . 4
1.4 zk-SNARK vs. Ring Signature . . . 4
1.5 Blockchain Binding with ACCTs . . . 5
1.6 Lightning Network . . . 6
1.7 Coloring . . . 6
2 Putting it All Together 7
2.1 Dymaxion Tangle vs. IOTA Tangle . . . 7
2.2 Blockchain-Dymaxion Interaction . . . 7
2.3 Using Nodes P2P for ad-hoc DLs . . . 8
2.4 Dymaxion Anonymity . . . 9
3 Implementation Details 10
3.1 Opening a DL . . . 10
3.2 A Transaction within a DL . . . 11
3.3 Closing a DL . . . 12
3.4 Blockchain Enforcing . . . 12
4 Security Considerations 12
4.1 Collusive Nodes Attack . . . 12
4.2 Spamming and DoS Scenarios . . . 13
5 Results and Discussion 13
5.1 Prototype Performance . . . 13
5.2 Adoption Process . . . 15
Acknowledgments 15
A SpaceMint Paper Errata . . . 16
B The Burst/Qora ACCT Process . . . 17
C Burstcoin CIPs . . . 17
References 23

Introduction

Cryptographic currencies, or cryptocurrencies for short, have been around for about 8 years now. After an existence on the fringes, they are being perceived increasingly as disruptive technology that may change the way how we will see and use money in the future.

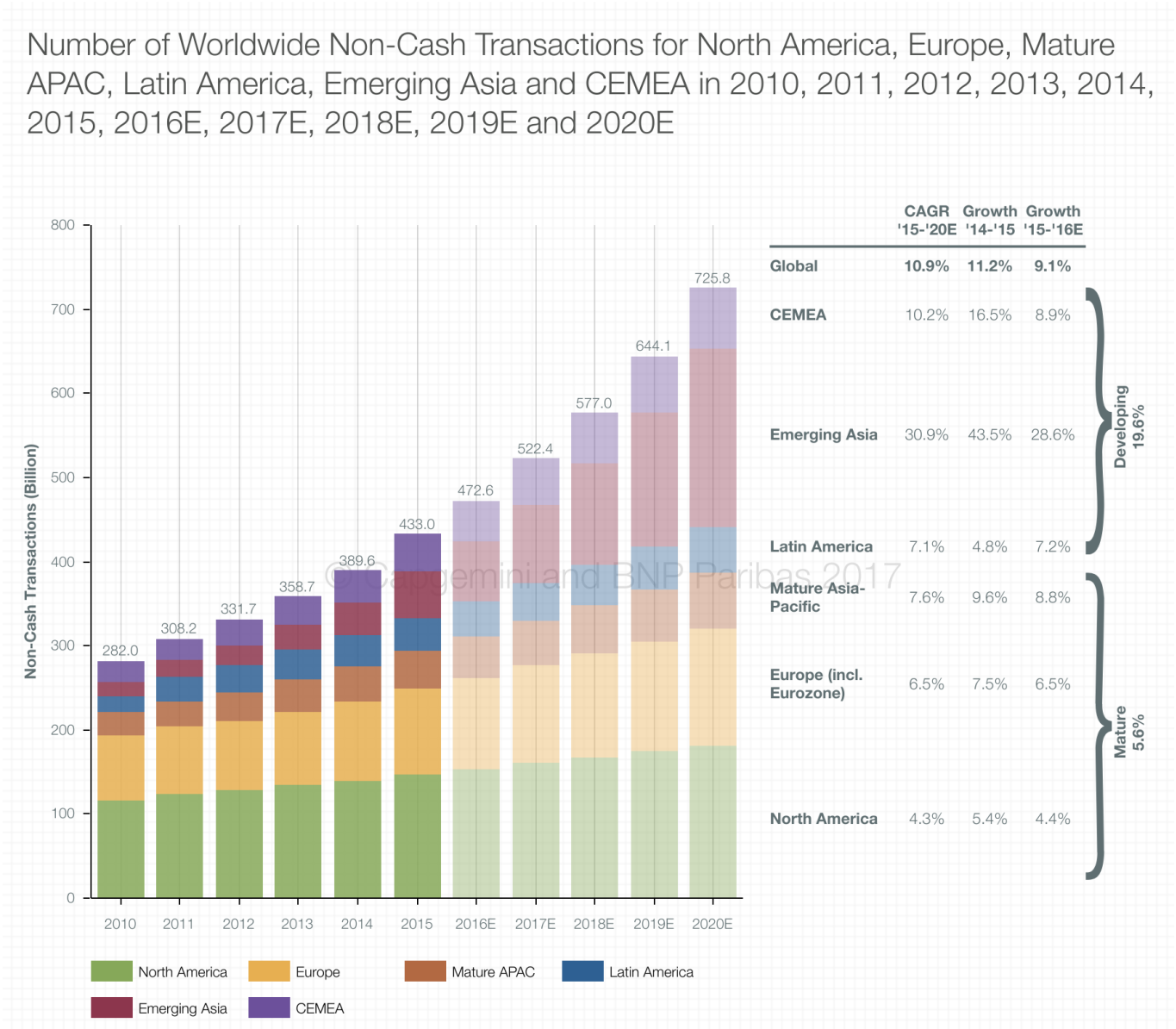

Figure 1. Cap Gemini/BNP Paribas: Number of Worldwide Non-Cash Transactions 2010-2020 (est.)

Despite their technical elegance and possibilities, cryptocurrencies lack mass adoption because of two critical factors: usability and scalability. While usability can be addressed with sufficient engineering and development effort, scalability is often a theoretical problem of its own.

The inherent features of a good cryptocurrency - decentralization and trustless design - often go contrary to traditional methods of upscaling centralistic processes. While Bitcoin's pioneer achievement, the blockchain, solved the problem of a decentralized trust, its inventor certainly left much headroom for scaling that concept for a truly global use.

In fact, speaking of "truly global use", let's fathom the scale of this concept. Assume you would like to support one hundred million people world-wide to be able to use a cryptocurrency like Bitcoin with an average of just 1 transactions per day (about 1.5% of the global non-cash transaction volume - see also figure 1). You would need a 400MiB blocksize to meet that demand. This would imply a blockchain growth of over 56GiB per day.

Aside from the feasibility considerations, it is questionable if a transaction for the acquisition of a sack of rice made in China 2017 should be kept for all eternity on the blockchain and thus in memory or on mass storage of all participating nodes world-wide.

A concept to overcome many of the restrictions a typical blockchain imposes on scalability was presented in the IOTA white paper[1]. In this novel system, transactions are not broadcast over a network of participating nodes, some of which (miners) try to put them into a block. Instead, network nodes can validate transactions and by this validation work actually earn the right to perform transactions themselves. The nodes

have unified roles and form a kind of systolic array[2] for data processing. The data being a transaction propagated along a directed acyclic graph.

Due to it's success and thus increasing usage, Bitcoin started to experience scalability issues, now commonly hitting its theoretical tx/s limit. A long scaling debate about pushing this limit and removing other protocol constraints continues to the present day. One of the suggested extensions to the Bitcoin protocol is called the Lightning Network (LN).[1] It is basically a network of two-party ledger entries for participants who have subscribed to a bidirectional payment channel. While a LN solves many of the problems associated with high transaction volume, micropayments and their cost, it is not a silver bullet. E.g. it does quite little to increase the limits on the number of concurrent users.[3]

Dymaxion is a term that Richard Buckminster Fuller associated with much of his work. It is a portmanteau of the words **dy***namic,* **max***imum, and tens***ion** and sums up the goal of his study, *"maximum gain of advantage from minimal energy input."*

We use this term for the Burst Dymaxion to honor the achievements and influence of this true generalist and also because the geodesic dome - a structure Fuller helped to popularize - has very similar properties to what is proposed in this paper for transaction propagation: Elements of the dome - while each limited in resilience - distribute mechanical stress throughout the whole structure, making geodesic domes able to withstand very heavy loads for their size. In our proposal, the Dymaxion is a set of tangle-based lightning networks, each single one used for propagating transactions. Each single of these networks distributes "transactional stress" among its whole network of participating nodes, allowing for arbitrary scalability. The more nodes participate, the higher the transactional capacity of each of the currently active networks becomes. As any number of these tangle-based LNs can co-exist at the same time, we speak of *Dymaxion layers* (DL).

We therefore present nothing short of an energy-efficient cryptocurrency that is able to sustain the total global load of non-cash transactions.

Below all of this is a small and very moderately growing energy-efficient blockchain with Proof-of-Capacity consensus used for bookkeeping the opening and closing of the Dymaxion layers above it. See also central figure 7.

1. Building Blocks

1.1 Burstcoin Blockchain

Burstcoin is a digital cryptocurrency and payment system derived from the Nxt cryptocurrency. As such it is based on the blockchain technology (Figure 2 shows the general concept of a blockchain with the most essential terms explained), but is using a special class of consensus algorithm called proof-of-capacity (PoC). This algorithm allows for the usage of storage space instead of raw computational power as is being used in Proof-of-Work (PoW) systems and is responsible for making Burstcoin energy efficient. Efficiency - while similar to Proof-of-Stake (PoS) systems, is still achieved by a low barrier to entry, where you do not need to acquire a significant stake of the currency in order to be able to participate in the mining consensus[4].

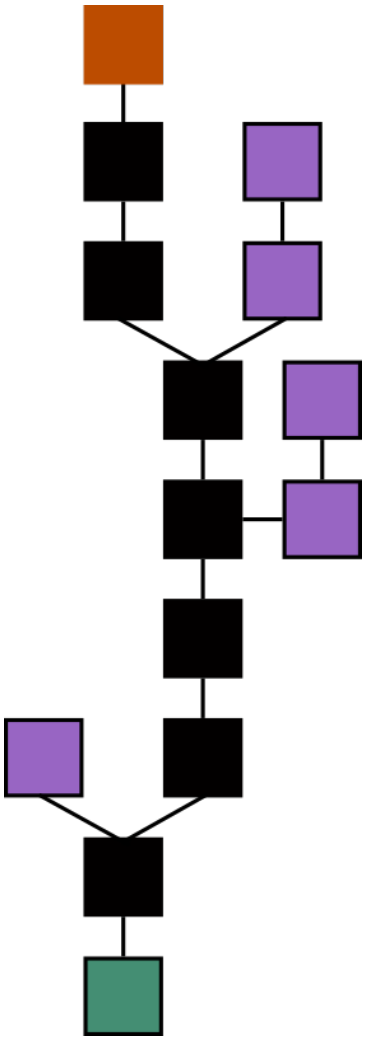

Figure 2. The *main chain* (black) consists of the longest series of blocks from the *genesis block* (green) to the current block (orange). Orphan blocks (purple) exist outside of the main chain on so called *forks*†.

1.2 Smart Contracts

A blockchain can be seen as a distributed database that is ensuring consistency and validity by verification work done by a majority of participating nodes in this blockchains network. While cryptocurrencies such as Bitcoin speak of the blockchain as a public ledger, others such as Ethereum put an emphasis on *smart contracts* - and not only monetary transactions - to store in their blockchain.

A smart contract is a procedural way, usually done by a computer program, to facilitate, verify, or enforce the negotiation

[1] Ethereum has a very similar concept under the name *Raiden Network*

or performance of a contract. Compared to their "inanimate" paper-based ancestors, smart contracts also fulfill the role of otherwise needed lawyers (verification), notaries (validation) and executors (enforcing). For this very reason, smart contracts are seen as a disruptive technology to future digital economies.

Burstcoin has been given a smart contract system in 2014. The formalism used is called AT (Automated Transaction) and has been proposed and implemented by CIYAM[5].

As turing-complete formalism, ATs are both powerful (expressiveness) and dangerous (verifiability) and have been used only sparsely as templates to facilitate simpler smart contracts (SCs), such as lotteries, crowdfunding and an asset exchange. Because of the expressiveness of the formalism, other possible applications are basically limitless, yet have to be designed with great care to avoid situations such as the DAO debacle that led to the hard fork and community split of Ethereum and Ethereum Classic.[2]

1.3 DAGs and The Tangle

A directed acyclic graph (DAG) is a restricted form of a finite directed graph that has no directed cycles. In short, it is a data structure with interesting computational properties and widespread use in computer science. We assume the reader has basic understanding of graph theory and encourage to acquire it otherwise. Figure 3 shows an example of a simple DAG.

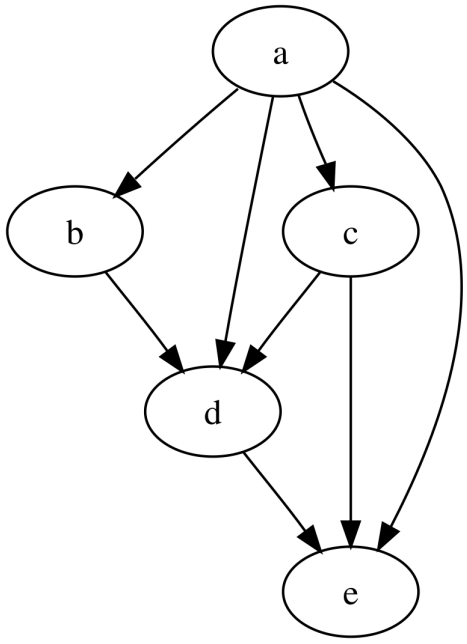

Figure 3. A simple directed acyclic graph consisting of 5 *vertices* (a-e) and 8 *edges* connecting them. The arrows stand for a direction and there are no loops - therefore *acyclic*.

The IOTA white paper [1] describes a blockchain-less approach of ensuring transactions in a cryptocurrency with the use of a so called "Tangle", which is in essence a DAG. The main proposition of this concept, is its feasibility to provide a good infrastructure for microtransactions, mainly targeted at the IoT industry.

Any microtransaction-capable infrastructure seeks to overcome the problem of paying a fee that is larger than the amount of value being transferred. The smaller the value of a microtransaction becomes, the more applications open up to it (tipping, machine-to-machine transactions, microservices, spam protection etc.) but the harder it gets to lower the transaction cost below the value being transferred. Ideally the transaction cost would be zero, but then again - what would the incentive for a transaction processor be?

IOTA answers this dilemma by unifying the roles of network participants. There is no more discrimination between a transaction (tx) issuer and a tx verifier - a *node* has both these roles. The cost for issuing a tx is to provide the verification to at least two other txs. Doing this, the network security is kept up by issuing transactions. It is assumed that the nodes check if the approved transactions are not conflicting. If a node finds that a transaction is in conflict with the tangle history, the node will not approve the conflicting transaction. If a node issues a new transaction that approves conflicting transactions, then it risks that other nodes will not approve its new transaction, which will fall into oblivion.

While not discussing the implementation, the IOTA white paper suggests there being a hidden PoW in the validation process:

> For a node to issue a valid transaction, the node must solve a cryptographic puzzle similar to those in the Bitcoin blockchain. This is achieved by finding a nonce such that the hash of that nonce concatenated with some data from the approved transaction has a particular form.

The "incentive to participate" for a node (assume a node has no need to perform transactions, so why should it validate other transactions) is achieved by nodes keeping statistics on their peers, e.g. how many new transactions are received from a neighbor. If one particular node is "too lazy", it will be dropped by its neighbors. Therefore, even if a node does not issue transactions, and hence has no direct incentive to share new transactions that approve its own transaction, it still has incentive to participate.

1.4 zk-SNARK vs. Ring Signature

The acronym **zk-SNARK** stands for "Zero-Knowledge Succinct Non-Interactive Argument of Knowledge," and refers to a proof construction where one can prove possession of certain information, e.g. a secret key, without revealing that information, and without any interaction between the prover and verifier.

A **ring signature** is a type of digital signature that can be performed by any member of a group of users that each have keys (to sign messages). Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people. One of the security properties of a ring signature is that

[2] https://www.coindesk.com/ethereum-classic-explained-blockchain/

it should be computationally infeasible to determine which of the group members' keys was used to produce the signature.

Both concepts are central elements of cryptocurrencies with a high focus on anonymity, namely ZCash[6] (zk-SNARKs) and Monero[7] (ring signatures).

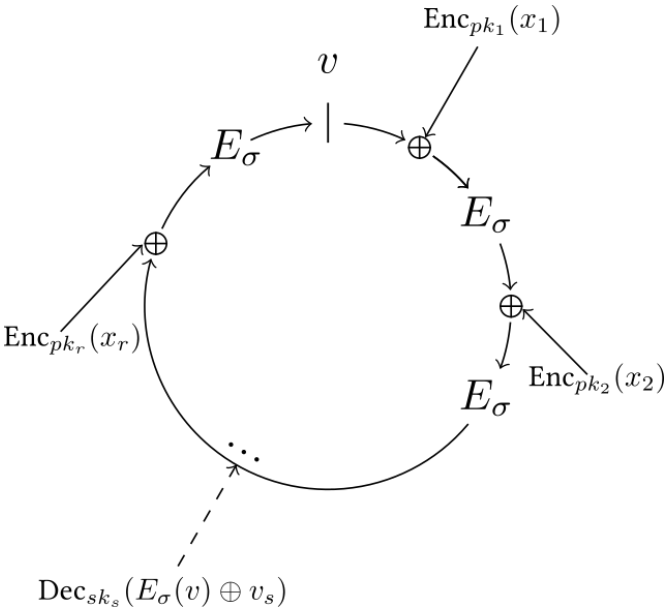

Figure 4. Rivest, Shamir, Tauman ring signature scheme

Ring signatures as referenced by figure 4 are a pretty plain and straightforward way of a multi-sig application: Given n entities each with pub/priv keys, $(P_1,S_1),(P_2,S_2),...,(P_n,S_n)$. Party i can compute a ring signature σ on a message m, on input $(m,S_i,P_1...P_n)$. Anyone can check the validity of a ring signature given σ, m, and the public keys involved, $P_1...P_n$. If a ring signature is properly computed, it should pass the check.

Compared to this, zk-SNARKs seem outright "magical" at first glance. You can verify the correctness of computations without having to execute them and you will not even learn what was executed – just that it was done correctly. In reality, zk-SNARKs can be dissected into four relatively simple ingredients:

1. represent as a polynomial problem
2. random sample for succinct check
3. homomorphic encoding / encryption
4. zero knowledge by checking *structure*

We will not explain these in full here, but encourage the reader to visit good introductions and explaining blogs for zk-SNARKs[8], [9].

We will not discuss the concrete implementation of both concepts here, as there are already off-the-shelf implementations on github[3] [4]

[3] `https://github.com/apuljain/Linkable-Ring-Signature`

[4] `https://github.com/scipr-lab/libsnark`

Both ring signatures as well as zk-SNARKs do have their advantages and drawbacks in anonymizing transactions, which leaves either of these functionalies as optional feature in the setup parameter space for created tangles.

1.5 Blockchain Binding with ACCTs

Atomic Cross-Chain Trading (ACCT) is a mechanism, where two (or more) parties own coins in separate cryptocurrencies, and want to exchange them without having to trust a third party. The term "atomic" here means indivisible, and refers to the fact that sending coins on one chain and sending other coins on the other chain cannot be performed independent of each other.

If these transactions could be performed independent of each other, then while one party could fulfil their side of the bargain and send some coins on one chain, the other party would have the option of going back on his end of the bargain and simply not following through with the protocol, ending up with both coins.

A decentral exchange where two parties, A and B, want to exchange tokens can be based on the following process:[5]

1. A generates some random data (called the secret) x.

2. A generates Tx_1 (the payment) containing an output with the chain-trade smart contract in it. It allows coin release either by signing with the two keys (A_k and B_k) or with (secret x, B_k). This transaction is not broadcast. The chain release script contains hashes, not the actual secrets themselves.

3. A generates Tx_2 (the contract), which spends Tx_1 and has an output going back to A_k. It has a lock time in the future and the input has a sequence number of zero, so it can be replaced. A signs Tx_2 and sends it to B, who also signs it and sends it back.

4. A broadcasts Tx_1 and Tx_2. B can now see the coins but cannot spend them because it does not have an output going to him, and the tx is not finalized anyway.

5. B performs the same scheme in reverse on the alternative chain. The lock time for B should be much larger than the lock time for A. Both sides of the trade are now pending but incomplete.

6. Since A knows the secret, A can claim his coins immediately. However, A, in the process of claiming his coin, reveals the secret x to B, who then uses it to finish the other side of the trade with (x, B_k).

ACCTs have the disadvantage that both chains need to implement the ACCT (of course in addition to being capable to provide smart contracts) in order for this to work.

If Burst was to provide some ACCT with - say - Bitcoin, the ACCT handling would have to be implemented not only on

[5] see Appendix B for a visualization

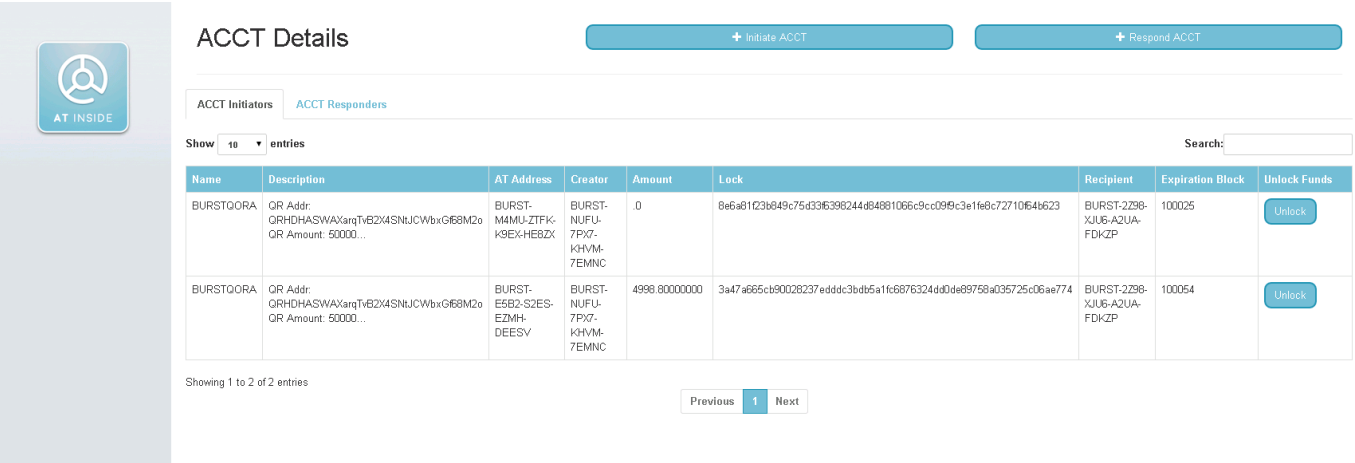

Figure 5. Burst/Qora ACCT prototype

the Burstcoin side, but also in Bitcoin client(s). While not entirely impossible, it is unlikely to get arbitrary cryptocurrencies development teams to provide that kind of interoperability. If, on the other hand, the implementation of ACCTs happened in a situation where the same development team exercises control over both "chains" in question, then this approach certainly seems viable.

Burst already does have a prototype ACCT implementation[6] with the cryptocurrency Qora - see figure 5.

1.6 Lightning Network

The term "Lightning Network" became widely known in 2016 from a publication by Poon and Dryja[10] in the context of the Bitcoin scaling debate. In essence, it is an off-chain protocol running on a P2P network of nodes for making micropayments of digital currencies. Said nodes form a scale-free network of bidirectional payment channels without delegating custody of funds or trust to third parties. Yet, these payment channels are bound by smart contracts to the underlying blockchain to ensure enforcability in case of uncooperative participants.

At the basis of this mechanism are so called *Hash Time-Locked Contracts* (HTLC) implemented as smart contracts on the blockchain. When opening a payment channel, participants must commit an amount in a transaction which is registered on the blockchain. This can be seen a collateral and guarantees enforcability of transactions (so-called commitments) done off-chain in the payment channel. Given their similarity, ACCTs are probably the origin of the technique now called HTLCs.

As introduced for Bitcoin, this system is conceptually not an independent overlay network; it is more a deferral of state on the standard blockchain, as the enforcement is still occurring on the blockchain itself (albeit deferred to future dates and transactions).

Because the payment channels are bi-directional between two parties only, forming a network where n parties can participate in transactions between each other depends on finding "paths" between two parties that can go through intermediaries. In a way, the formation of a lightning network relies on transitive properties of separate P2P communications.

The problem with this approach is similar to the problem of premonetary trade. Achieving transitive pathways for m out of n network participants becomes increasingly difficult if n is not much larger than m. The old explanation why money was invented in the first place holds true for a LN itself: If A had twenty hens and wanted to exchange them for one pig, he either had to find someone willing to do exactly that trade, or two other people - say B and C - where C might have had a pig wanting two goats and finally B wanting twenty hens and having two goats. The categories in that example translate to incompatible monetary volume and time in a LN.

It is possible that operation of a LN is well feasible with the help of powerful intermediaries/facilitators who command large financial resources. In this case, however, we can expect the LN to exhibit a decentral and not a distributed topology - compare figures 6 and 8.

1.7 Coloring

In an abstract sense, coloring is a simple tagging technique used to allow distinction, thus coexistence, of various instances of a class in a common data context.

In cryptocurrencies, colored coins is a concept that allows attaching metadata to transactions and by this leveraging the coins infrastructure for issuing and trading immutable digital assets that can represent real world value.

Colored coins[11] are a method to track the origin of Burst coins, so that a certain set of coins can be set aside and conserved, allowing a party to acknowledge them in various ways. Such coins can be used to represent arbitrary digital tokens, such as stocks, bonds, smart property and can even represent real-world objects.

When a coin is "colored", it can be traded on the Burst network just like any other coin in the system. This allows BURST to be exchanged for whatever object the colored coin represents. This concept forms the basis for the Burst Asset Exchange.

[6] based on `http://ciyam.org/at/at_atomic.html`

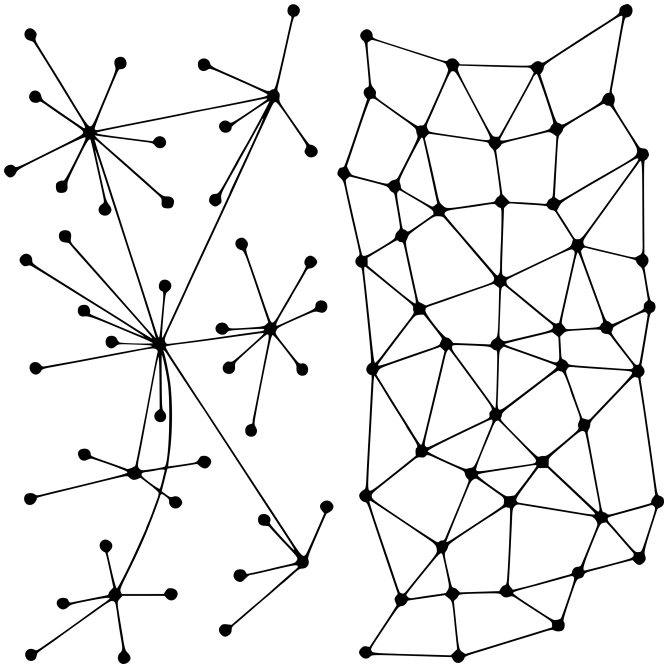

Figure 6. Decentral (left) vs. distributed topology.

Since 2014 it has been suggested, that various coloured coin protocols could be of interest to banks and major financial institutions,[7] because of its inherent applicability to their requirements to use self-issued, quasi-private, yet decentral and trustless transaction channels with desired properties.

Since then various implementations like CoinSpark[8] and MultiChain[9] have evolved from the original concept by Rosenfeld.

While our proposed combination of Colored Coins and tangle-based Lightning Network (which we call *colored tangles*) is a brand new concept, prototype implementations for the Bitcoin LN do already exist.[10]

2. Putting it All Together

As we have seen, all building blocks for our proposal are already in place, not only as theoretical concepts, but well developed and even tested in real-world application use. These are solved problems.

The realization of the Burst Dymaxion therefore seems more like an engineering rather than a research task, but the major contribution of this paper is the seamless integration of said components into a new framework with significant synergistic gains.

[7] https://is.gd/vz3oiN
[8] http://coinspark.org/developers/
[9] https:/www.multichain.com/
[10] https://is.gd/7RcPqf

2.1 Dymaxion Tangle vs. IOTA Tangle

While DAGs have been used as part of graph theory for several decades now and are mathematically well understood today, their use in cryptocurrencies as generalization of the blockchain is a novelty.

We are convinced of the advantages tangles bring to questions of scalability and decentral design, but not every aspect of the current iri (IOTA reference implementation) seems to be the best possible design choice.[12]

Our biggest concern is the cryptographic hash function used in IOTA: Curl. The IOTA vulnerability report[13] has shown already practical signature forgery attacks and while IOTA no longer uses the Curl hash function to hash transactions as part of the IOTA signing process, Curl is still used for other purposes in IOTA.

Similarly disturbing seems the claim of one of the IOTA founders to have placed that vulnerability into Curl on purpose, to be able to shut down copycats. Even leaving the ethical aspect aside - publishing IOTA source under GPL3 license and allegedly placing a cryptographic vulnerability in its core hash function - experience shows, that control over backdoors and vulnerabilities seldom remains in the “right hands”. By extension, this puts IOTA itself at risk.

It seems more apropriate, to use the iri only as design guidance in some aspects, using standard libraries and standard hashing algorithms complemented by own peer-reviewed implementation.

Moreover, the biggest difference being the underlying Burst blockchain allowing Dymaxion tangles to be opened and closed against a point of reference, as well as the existence of an arbitray number of logically separated (colored) Dymaxion tangles compared to just one IOTA tangle. A future IOTA could very well live within the Dymaxion, maybe even in its native form.

2.2 Blockchain-Dymaxion Interaction

Figure 7 is the top-level view on the presented concept. The Burst blockchain acts as a fundamental bookkeeping chain where opening and closing of the Dymaxion layer is recorded. Payment processors, banks, exchanges etc. may use ACCTs (ACTTs really) to open and close payment channels with desired properties (network size, validation reqirements, etc.) and keep them open for a certain period (TTL) - e.g. one day for daytrading on Wallstreet.

The tangles work asynchronously with no inherent clock signal contrary to the 4-minute target time for block generation in the Burst blockchain. Transaction latency in a Dymaxion layer is effectively network latency plus validation latency

$$t_{tx} = t_{nl} + t_{vl} \tag{1}$$

with network latency effectively being the packet time between two peers as sum of network-hop latencies, often measured in milliseconds, and validation time being PoW, PoC or PoS runtime on the node - measured in seconds.

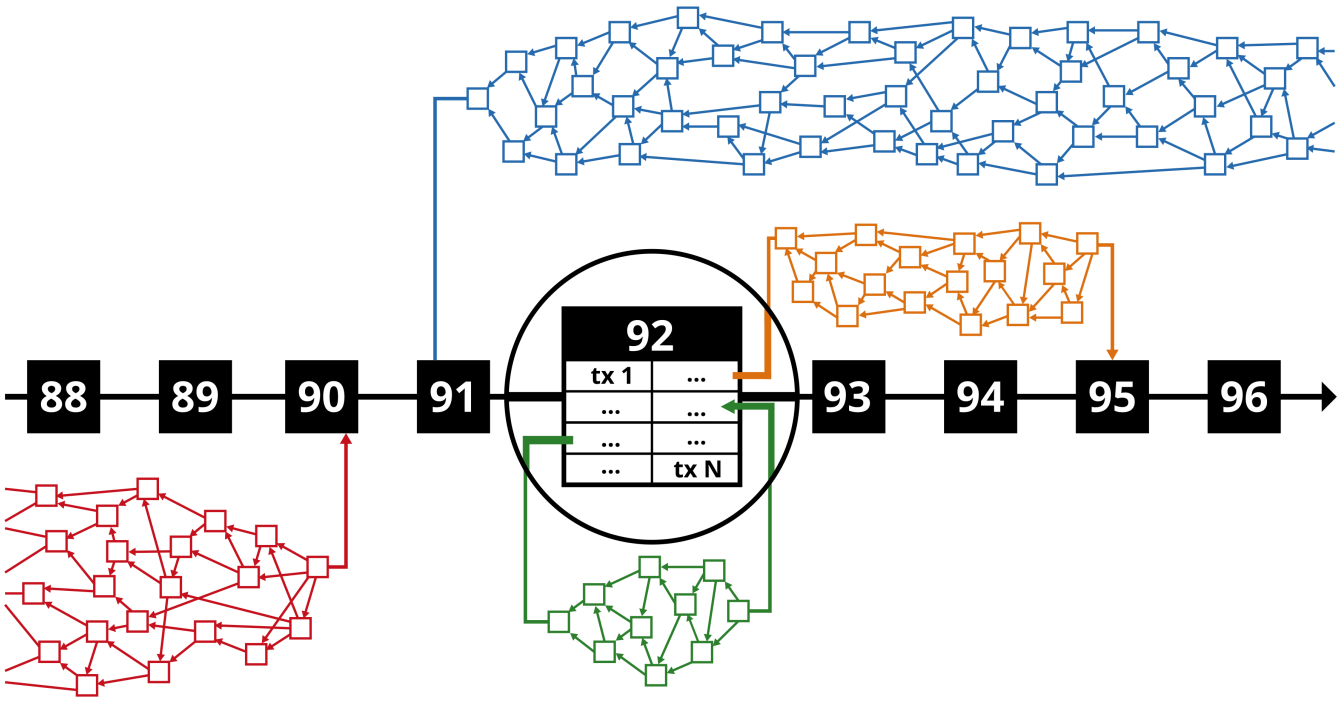

Figure 7. Dymaxion Top-Level view: set of colored tangles over the Burst blockchain. Each tangle representing an independent Lightning Network for P2P transaction channels. Block 92 is magnified to show inner structure containing transactions some of which are opening and closing tangle LNs.

With the average block time on the Burst blockchain being said 240 seconds, this can lead to a situation of potentially short-lived tangles that can be opened and closed intra-block i.e. within one and the same block on the Burst blockchain (see green tangle). The application for this could be a voting where some hundred or a thousand globally distributed voters would have to give their vote within a short period.

Some tangles might live for a day or longer and bear the transactional load of a financial institute or an exchange and only at the end of the day or the respective period close and record that state in the blockchain.

As mentioned in section 1.5, ACCT stands for Atomic Cross Chain Transaction. We will use a slightly modified ACTT - for Atomic Chain-Tangle Transaction from now on. Of course, figure 7 is a very simplified view if you consider that basically every block pictured potentially could - without any Burst blockchain scaling CIP - have up to 255 incoming or outgoing ACTTs. We see only three ACTTs for block 92, one incoming ACTT for blocks 90 and 95 and one outgoing for block 91 and even this simplification already suggests a multiple of the traditional blockchain-only transaction capacity.

We will discuss the implementation details (opening, closing, enforcing) in section 3.

2.3 Using Nodes P2P for ad-hoc DLs

Each cryptocurrency network consists of *nodes* and each of these nodes has contact to other nodes which from its perspective are called *peers*. This infrastructure is already in place for Burst and there is a communication protocol between these nodes to ensure transactions are propagated, blocks are downloaded for new or outdated nodes (blockchain syncing) etc. - see `https://burstwiki.org/wiki/The_Burst_API`

Similar to the Lightning Network Daemon (lnd[11]), we implement a Burst Dymaxion Daemon (bdd) that can - but doesn't need to be - an integral part of the Burst wallet[12].

In any case, the bdd would be active on nodes that do participate in specific colored tangle networks. The decision whether to participate or not, would be defined on dynamic node capabilities as defined in the respective CIP - see appendix C.

One particular detail of the DL network infrastructure acquisition - node discovery - is presented in the next section (see 3.1). A tangle-based network is made for a high degree of decentralization and transaction performance, but has a problem upon startup if there is not a network of sufficient size in place.

The Burst Dymaxion solves this problem by effectively providing the equivalent of a cloud service for nodes. Figure 8 shows the Burstcoin core network - running the PoC blockchain protocol. This base network has been running for over 3 years now and will always form a source of nodes potentially available upon DL initiation.

[11] `https://github.com/lightningnetwork/lnd`

[12] BRS - Burst Reference Software: `https://github.com/PoC-Consortium/burstcoin`

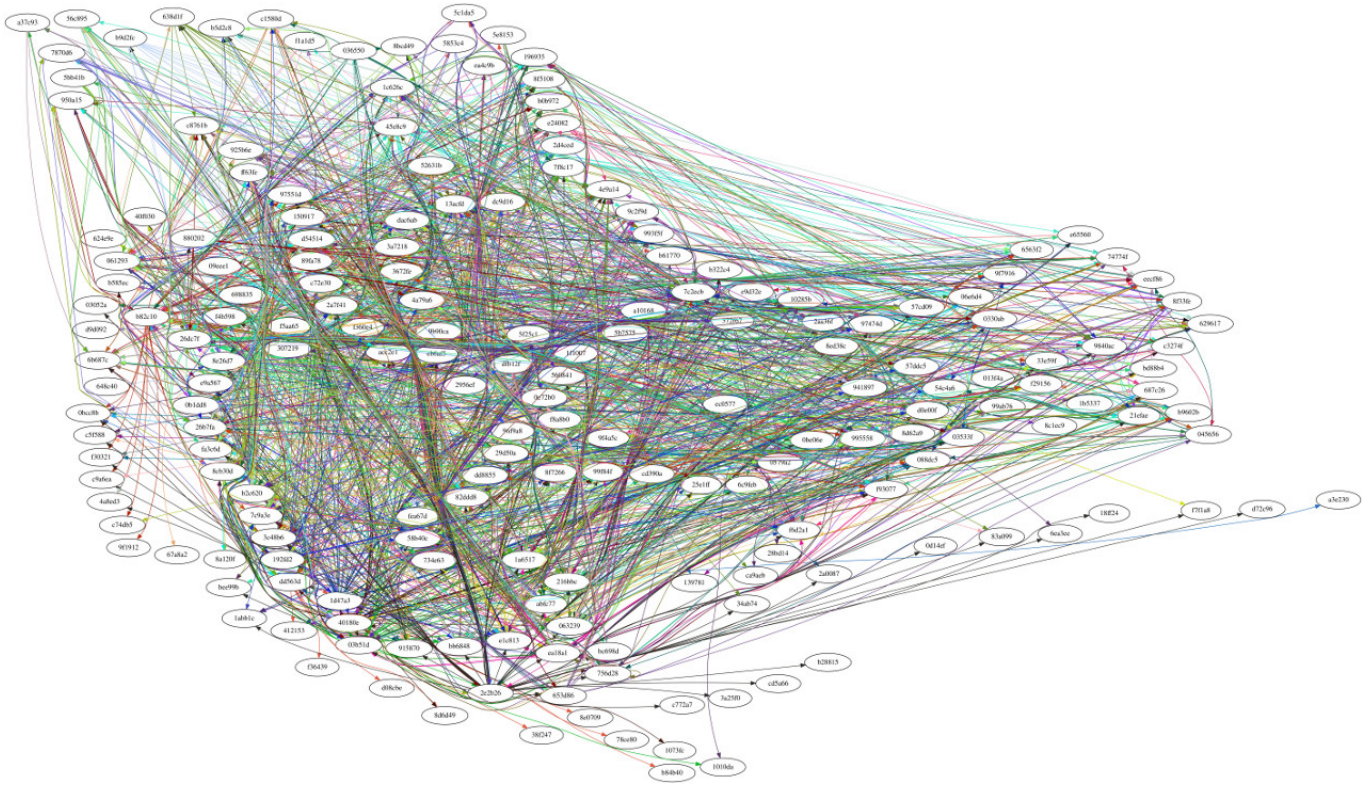

Figure 8. Burst cryptocurrency core network of nodes as of December 22nd. Node discovery by recursive descent.

2.4 Dymaxion Anonymity

Both ring signatures as well as zk-SNARKs are part of the DL implementation and upon opening a DL can be chosen - optionally - to conceal transactions that do happen within that specific DL. This has consequences for the respective DL record (see section 3.3 as well as for monetary supply on the DL.

For various reasons, there are no plans to bring RS or zk-SNARK anonymity on-chain. Due to the cutting edge nature of ZK-Snarks, in our opinion there is not enough peer-review of the underlying cryptography to make it mandatory or part of the base chain. Participants of a DL who desire anonymity can limit the value-at-stake by providing zero collateral. An application scenario would be dissidents that had to carry only clearing cost for on-chain operations and use the DL for encrypted communication, voting or similar.

We believe a global payment system should be agnostic of any moral, legal, monetary or social requirements as these are most often not global in nature. That's why groups wanting to use the DL for anonymous value-transfer activities can do so. Enabling the anonymity features of a DL is supposed to be the result of a cost-benefit consideration.

One such potential cost to anonymous DL mode of operation comes from the fact that:

> In Zcash, where there is no "certain scarcity" and if god forbid, Zcash had a bug that allowed for people to generate more Zcash coins than the intended money supply, then it is possible that nobody could tell. If it were a severe bug, potentially somebody could inflate the money supply by hundreds of millions of dollars, making a profit while lowering the price of Zcash for speculators. There are several examples of major cryptocurrency bugs that have led to a massive misallocation of the quantity of cryptocurrency that should have been in circulation.[13]

Contrary to this, even in the worst case of a bug in the anonymity functions, due to the TTL of a DL any possible damage is contained and limited to the collaterals used when forming the DL. Even if a "DL goes wrong", this has no effect on other past, present or future DLs and certainly not on the base chain.

While zk-SNARKs and the Pinocchio protocol[14] do allow for concealment of even the value being transferred, said problem with an uncontrolled supply of tokens may not be worth the risk.

Ring Signatures provide a very nice alternative for complementary scenarios with anonymity requirements. There is no risk of uncontrolled supply and in case the DL has a larger number of members RS can guarantee a sufficient level of anonymity with good transaction performance levels.

While anonymity on-tangle can be covered for a large parameter space of requirements, on-chain anonymity is somewhat weak by itself. Many addresses are re-used and even

[13] http://zcoin.io/zcoin-and-zcash/

named. Their connection to on-tangle transactions (via collaterals) can prevent any anonymization effort on-tangle.

Fortunately there are two mechanisms to improve on-chain anonymity. The somewhat crude approach is a mixing service, that theoretically can provide sufficient concealment of source and target payment streams, but from a cryptographic point of view it is more an obfuscation functionality than real anonymization.

The other mechanism for on-chain anonymity is a little-known feature of CIYAM AT: anonymous ACCTs.

One might think, that anonymous ACCTs are not possible, because the AT itself has no secrets, but with slight changes to the process it can be altered to allow just that.

Normally an ACCT would involve putting the same "hash" into two ATs (each residing on a different blockchain, or in our case chain and tangle) and then sending the secret to unlock both in order to make the transaction take place - see appendix B.

It turns out, there is no need for the secret to be "identical" - so the on-chain AT would store the hash

```
e47823401ae24e6885de22e1c427c9df\
e477ce5e1095f3c2bf4dcbc35f2c7ee0
```

while the AT on-tangle could store

```
0d5a2b016e90b3db4d42d5a4c420f75b\
b131bb4465bdfebb8037daadf0174a54
```

Both hashes seem completely unrelated, and to some observer they are, yet the ATs do have the knowledge that one is a SHA256 of "Burst123", while the other is the SHA256 of "Tangle123". (this is of course a trivial example of how to do this - the in-production version does use a more elaborated callenge-response process than just using fixed strings with some concatenated id).

Assuming we will have numerous ATs operating across numerous blockchain-tangle DLs all doing transfers at around the same time it is pretty clear that any observer aiming for "total awareness" would have to record the total global sum of transactions the Dymaxion is performing. It seems highly unlikely for this to succeed when observing a "globally decentralized transaction machine".

3. Implementation Details

The life cycle of a Dymaxion Layer consists of the following steps

1. Initiation
   (a) Initiator set up
   (b) Creating the DL (Node Discovery, AT)
   (c) Subscribe Phase
2. Operation
3. Closing
   (a) Node DL shutdown broadcast
   (b) Clearing

3.1 Opening a DL

Opening a Dymaxion Layer (DL), is the main operation for a tangle to spring into existence. While more complex, the operation is conceptually very similar to an ACCT, creating an asset or the BTC Lightning Network initiation process.

DL Initiation

A *Tangle Initiator* creates an on-chain ACTT (which is effectively a HTLC) with several tangle parameters including, but not limited to:

Consensus Type PoC, PoW and PoS will be supported. PoC is the preferred way for full nodes with attached storage ("Plots"), PoW and PoS available for various application scenarios, where small tangle nodes (IoT) possibly cannot perform PoC or some even PoW.

Time-To-Live (TTL) Global Time-Lock for the tangle. A point in the future where the tangle will fold. Maximum TTL could be decades, but we propose in a first release a max. TTL of 2^{17} blocks on the Burst blockchain (ca. 1 year). Must be bigger than the Subscribe-Time.

Subscribe-Time Deadline for subscribers to sign their collaterals on-tangle. Maximum 360 blocks (ca. 24h).

Collateral Collateral in BURST by the tangle initiator. This collateral is used as backing for whatever Units (see below) the initiator issues. The collateral can be as low as 0, but then any currency, asset, bonds, futures, shares or whatever the issued Units represent must be backed by other means (promises - similar to current Burst Asset exchange) and might not have as much trust as a collateral-backed issuance. The collateral is blocked and unspendable for the issuer until the tangle folds. Technically, the collateral is defined by the BURST address and it's UTXo which the initiator names and signs.

Features A DL can be created with more parameters defining its behavior. The most significant probably being anonymous transactions and their type, where the participants (initiator and subscribers) can perform transactions using ring signatures or zk-SNARKs (see 1.4). In this case, records of the tangle transactions (see 3.3) can be kept, but do not contain any information to make the participants of the tangle identifiable.

Units Number and types of units issued, can be applied e.g. for different types of stocks like common, preferred for an IPO issuance. Other than that each type of Unit can be a 64bit number of *Quants*.

Subscribers Optional list of addresses who are subscribers to the tangle - the participants. This is a pull operation, that can happen without interaction of the subscriber himself. It has no effect on the availability of the funds on these particular addresses. All are spendable or can receive more BURST. Until the subscriber himself does not sign

his participation on-tangle, the particular address is still *detached* from the tangle. If this parameter is given, we speak of a *private DL* else it is a *public DL* - open to everybody, in which case tx fees for clearing of balances upon folding are paid by the subscribers themself.

Cost Creating a tangle implies a minimum transaction fee, which is independent of the collateral. It consists of a fixed and a variable part. The fixed part being a constant fee going to the miners in the block where the tangle opening was recorded .The variable part depends on the number of subscribers - see equation 2 - and is locked to a time until the tangle folds. This fee is intended to cover the tx cost for clearing operations for the participants.

$$F_{var} = N_{subscribers} * F_{ordinary-payment} \tag{2}$$

Although the on-chain accounts are protected from any form of hijacking - simply being referenced by the tangle initiator has no effect - an initiator has also no motivation to reference an arbitrary number of on-chain accounts, as this will raise the tangle opening fee. Unspent tx fee (reserved for a subscriber who never signed on-tangle) is forfeited for the tangle initiator and goes to the miners of the Burst blockchain block where the folding of the tangle is recorded.

The tangle initiator is technically just the first subscriber to the tangle, therefore his collateral is defined the same way as for all subsequent subscribers: by providing a signature for the respective Burst address with X funds at the time of signing.

By successfully recording the tangle to the Burst blockchain, it will get a unique ID - its "color" - and by definition all units on this colored tangle are limited to this context. As of now, there is no inter-tangle transaction possible, all value transfer desired to happen between two different colored tangles has to go over the Burst blockchain as intermediary.

Node discovery

Figure 9 shows the step immediately following the DL initiation: node discovery. In this step, a DL initiation requirement is broadcast to the peers of the node that got the tangle initiation requirement (not necessarily the node of the tangle initiator).

The peers record this requirement, broadcast it to their peers and check the DL parameters against their own capability configuration (see C) and in case the node's configuration does allow for handling the requested DL, the node becomes part of the DL network.

In section 5.1 we discuss the latency and throughput benchmarks for tangle initiation under various connectivity scenarios and geographic situation.

DL Subscription

Within the *Subscribe Time* as defined by the tangle initiator, each subscriber must sign his address on-tangle to become participant of the newly formed DL. If a list of subscribers has been given, the subscriber must be part of that list (private DL) and the respective tangle can be seen as an invite-only. If no such list has been given, the DL is open to everyone with an address on the Burst blockchain.

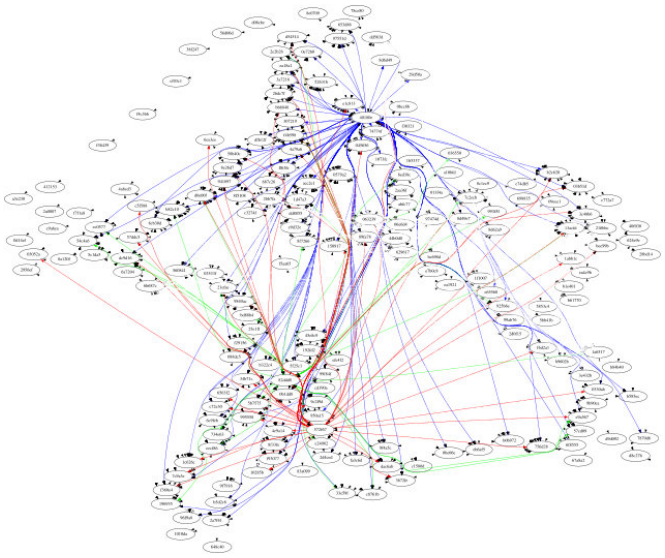

Figure 9. Parallel node discovery for 3 colored tangles (DL network infrastructure acquisition): Peer lookup, check against node capabilities.

While there are defined maximum values for TTL and subscribe time, there is no minimum time given, except the constraint

$$t_{ttl} > t_{st} \tag{3}$$

must be met. This allows for short lived dymaxion layers that can be opened and closed within one block and its approx. 240 seconds life time. Naturally these kind of short-lived DLs are probably more suited for machine2machine IoT transactions instead.

3.2 A Transaction within a DL

For consistency reasons, to describe a transaction process on the DL, we would like to use parts of the IOTA terminology: *sites* are transactions represented on the tangle graph. The network is composed of *nodes* which issue and validate transactions.

As in IOTA, the main idea of the tangle is the following: to issue a transaction, nodes must work to approve other transactions. Therefore, nodes that issue a transaction are contributing to the network's security.

So the price for a transaction on-tangle is not named in fees, but in validation work done for other transactions. IOTA uses some kind of "low-effort PoW" to validate a transaction. This design choice makes sense, if you target IoT devices and lots of them. The problem is, it works only with sufficient security if such a large scale tangle network is already in place.

The Burst Dymaxion implementation will support all three types of consensus algorithms, but we will start off with just PoC until the network has reached sufficient size and more experience from network operation is gained.

The on-tangle PoC validation will be quite similar to BURST mining and also using regular PoC2 plot files, although the re-

quirements for the plot size of a node will be much lower - in the order of 1-5 GB plot size. The node will validate transactions also by finding deadlines to the tx hash of the transaction to validate.

In case it can't find a deadline below a certain threshold, the validation has failed and the node can't validate that particular transaction.

If a node manages to validate two transactions this way, it can use its solution to send a transaction itself and the cycle repeats (with another set of on-tangle nodes).

Because this type of validation is verifiable (nodes can verify the PoC validation solution found by other nodes), there is no need for a central entity to issue any specific transactions as this is currently the case with the IOTA coordinator.

Moreover, there is no gain for nodes to amass more PoC space for validation, as it will not give them more "validation power". The PoC validation is a minimum-threshold satisfiability problem, which is either met or not. Should a miner with several hundred TB of PoC2 plots decide to use these for on-tangle tx validation (which we assume will be more the rule than the exception), it will not give him any more power than a regular 1-5 GB node has, as the tx validation PoC2 search is a 1st fit.

3.3 Closing a DL

When a Dymaxion layer is closed, we speak of the tangle *folding*. This is because upon initiation of closing the layer, clearing of all participants' (initiator and subscribers) balances happens and after the final balances are in the Burst blockchain - as ordinary payment transactions with the respective fee from the tangle setup cost - the whole structure of the tangle is scrapped. As of now, there are two possible conditions when a tangle will fold:

- the tangle TTL has expired
- there are no subscribers (except initiator) to the tangle after the subscribe-time

The first condition is defined by the tangle intiator when creating the dymaxion layer, whereas the second condition is defined by the subscribers who either do not show up (subscribe) on the tangle at all, or who unsubscribe.

Unsubscribing from a tangle does not free up the subscribers funds on-chain or on-tangle. These still remain locked until the tangle folds.

Tangle records

If a tangle participant chooses to keep the records, he is free to do so, but these are of no relevance to the funds that are now on the Burst blockchain.

Some participants, most often the initiator if it's a financial institution, are required by law to keep these records and they can do so, but these records have no impact on blockchain size.

On the other hand, the Burst blockchain can serve as warrant of validity of these privately kept records, as the tangle entry and exit points are stored. The final tangle state is being recorded with its hash value so forging the records, although they may have been archived by a single entity only, is considered impossible. This attribute of immutability is an important legal requirement in archiving financial documentation.

Even if all tangle participants choose or are required to keep the tangle records (similar to bank statements), the storage is in their responsibility and thus only of their local interest. The blockchain is free to just record openings and closings of the tangles working above it and sustaining the high-volume transactional load. Tangle storage is left completely to the private domain only to those who have interest keeping the records, yet these records' immutability is ensured by the decentral Burst blockchain.

3.4 Blockchain Enforcing

Security of off-chain transactions is enforced by blockchain smart-contracts without creating an on-blockchain transaction for individual payments.

The blockchain serves the function of an arbiter, so it is possible to conduct transactions off-blockchain without limitations. Transactions can be made off-chain with confidence of on-blockchain enforceability and deterministic results.

Enforcing the transfer of collaterals after folding of the tangle does not require cooperation from any counterparty. Clearing operations are solely on the ACTT running both on-chain as well as on-tangle.

While subscribed parties can choose to not participate in any on-tangle transactions, their on-chain collaterals remain frozen until the tangle is folded.

4. Security Considerations

4.1 Collusive Nodes Attack

In a tangle network a possible attack scenario is a set of collusive nodes causing either double spends ot modifying transactions along the graph while validating them for each other and causing a so called *parasite chain* - see [1], pg. 20.

The IOTA paper introduces a family of Markov Chain Monte Carlo (MCMC) algorithms to counter the problems of bad tip selection from collusive attackers.

The Dymaxion implementation uses instead a *Byzantine Consensus Algorithm*. From a mathematical model point of view, attackers in the network can be considered "faults" in a distributed computing protocol. Byzantine fault tolerance (BFT) is achieved if the non-malicious nodes have a majority agreement on their strategy like handling or default values for missing or corrupted messages.

We use the BFT-SMaRt[14] - a Byzantine fault-tolerant state machine replication library to build dependable protocols. Given f as the number of faulty nodes, the total number of nodes needs to be $3f-1$ for a correct operation of the transaction propagation on the tangle ($2f-1$ honest nodes).

This puts an upper boundary for the maximum tolerable portion of malevolent nodes at almost 33%. If there are more

[14] https://bft-smart.github.io/library/

than 1/3rd malevolent nodes, trustable transaction validation on-tangle is not possible without further security precautions.

While we believe that an attack with such a high number of colluding nodes is infeasible for large global networks, this theoretical limit also exists for the IOTA network as a whole, no matter the tip selection algorithm used.

Contrary to this, a Dymaxion layer has additional security measures in place. As described in section 3.1, the node discovery run picks potentially from the total set of nodes in the Burst network a subset suitable and available for the initiated colored tangle. Given N as the total number of nodes in the Burstcoin network, and T as the number of available and suitable nodes for the Dymaxion layer tangle, where $T \subset N$ (T is proper subset of N), collusive nodes would have to take up to 33% of the whole Burst network N and not only a subset T.

We also would like to point out, that all members of set N are constantly vetted as part of the Burst network and once misbehaving are blacklisted by other nodes in the network, which effectively bars them from being chosen into set T.

Even if a tangle initiator would work in collusion with such a network, the tangle initiation parameters cannot be set in a way that would favor the preferred choice of malicious nodes and by this lure the tangle subscribers into a trap to steal their collaterals.

4.2 Spamming and DoS Scenarios

Transactions on-tangle are free from fees. If they were truly zero-cost transactions, there would be nothing at stake for some attacker who would like to issue a huge amount of transactions for some own benefit (spamming) or to simply congest the network and hinder other transactions to be validated (DoS). In the worst case, the attacker could amass a processing power that would allow him to execute a whole class of majority attacks with dire consequences to the network as such.

IOTA currently counters this problem by the use of a so-called Coordinator - a specific node run by the IOTA foundation to make specific transactions called *Milestones* - to stabilize the network until it has reached a size that would make majority attacks infeasible.

The Burst Dymaxion design starts at a much better initial situation. Instead of a centralized entity to rely on, the underlying Burst network and the blockchain form already points of reference upon entry and exit of a tangle. Moreover, nodes propagating on-tangle transactions are required to perform a PoC for tx validation.

PoS and PoW consensus algorithms for on-tangle transactions will be added as DLs will run on devices not capable to perform a PoC consensus (or are defined for a low resource requirement operation see C), but the initial mode of operation will allow for the network to grow in a decentralized manner without any entity to exercise central control over any aspect of it.

As mentioned in section 4.1, nodes in the Burst network are vetted and blacklisted when misbehaving (e.g. spamming) already. This is actually a functionality that has been added and battle-tested in the wake of the spamming attacks that occured in July 2017.

5. Results and Discussion

5.1 Prototype Performance

We performed various benchmarks with prototypes for DL initiation consisting of node discovery latency timings, inter-node communication throughput and various cost metrics.

The benchmarks are done with a scripted prototype and are preliminary. As such they can be expected to improve, but as you can clearly see from the numbers, perfomance is good even in the worst case scenario with many adverse effects coming into play.

Latency

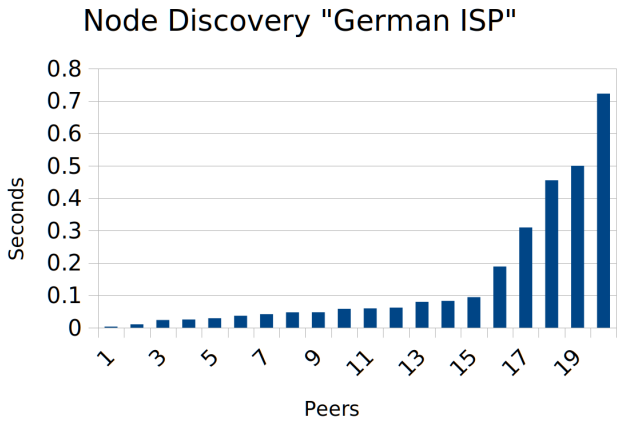

Figure 10. Node discovery at reference point (wallet.burst.cryptoguru.org)

Figure 10 shows node discovery latency from our point of reference, the PoCC online wallet, hosted as some german ISP and representing currently the best case scenario for the prototype. As you can see, the maximum response time from a peer answering (a potential tangle node) is below 0.8s with the majority of peers having answered within 0.1s. This can be considered the best case scenario, where a broadcasting node with a good connectivity is also in a region of dense Burst nodes presence.

For comparison, figure 10 shows discovery latencies from a node with regular internet connectivity (cable) for a home network, situated near a region with good Burst nodes coverage, but no nodes in the own country yet. This can be considered the standard situation in areas where the Burst network starts to proliferate. The PoCC network observer gives a good overview of areas with dense Burst nodes presence and their neighboring regions.

Even in this case, almost half of the answering peers remain under 0.1s with the slowest peer answering with a latency of

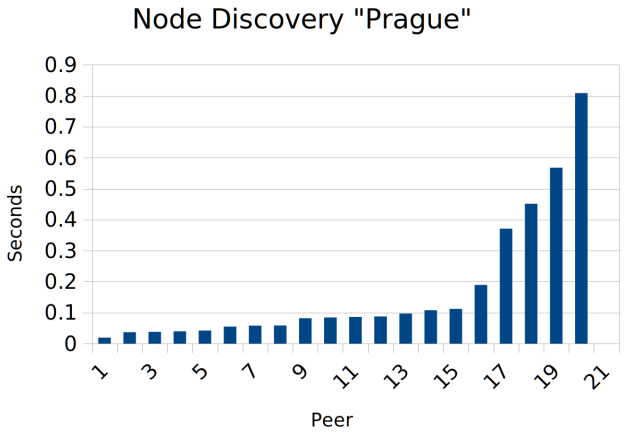

Figure 11. Node discovery from a Burst node situated in Prague - home network

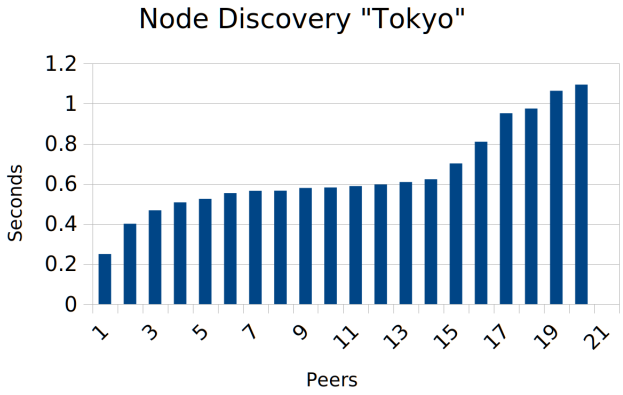

Figure 12. Node discovery from a Burst node situated in Tokyo - ISP network

around 800ms. Even for the very specific case of a *flash tangle* to be formed and folded within one Burst block (ca. 240 seconds) this would leave over

$$n_v = (240 - t_i - t_f)/t_{avg} \tag{4}$$

validation steps within the tangle for transactions. t_i being the tangle initation time, t_f the tangle folding time, t_{avg} being the median response time per node, 0.085228s in the "Prague" case. Depending on the width of the tangle $w = n_n/2$, this allows for at least

$$n_v * w \tag{5}$$

on-tangle transactions until folding the tangle potentially in the same block. $(238/0.085228) * 10 = 27925$ transactions in this case.

So even some pathologically small tangle consisting of only 20 nodes, could perform intra-block almost 28k transactions in a 240s time window (over 10 million tx per day). This is the green tangle depicted in figure 7.

While this number isn't even the best possible case, let's look at some worst case scenarios. Tangle initiation, folding and operation from a region far away from dense Burst nodes presence. Once Burst nodes proliferate to form a truly global and omnipresent network, such a situation may not even exist, but it is always good to have a $\underline{\lim}_{n\to\infty} x_n$ estimate.

Figure 12 shows the situation for a node far away from other Burst nodes. We can see a maximum of 1094ms latency with the median being at 582ms. In this case our formulas 4 and 5 resolve to roughly $(237/0.582) * 10 = 4072$ transactions during a regular Burst block time (1.46 million tx per day).[15]

The most exiled node the PoCC could benchmark is situated in Sydney, Australia. Measured latencies from this node

[15] theoretically 237.812 seconds, but we are rounding to account for further processing delays

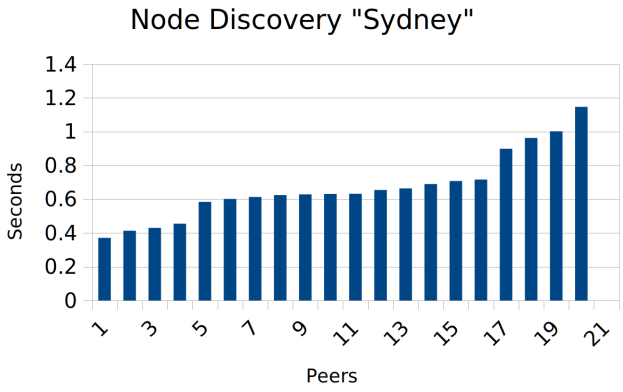

Figure 13. Node discovery from a Burst node situated in Sydney - ISP network

with a tangle formed in Europe are shown in figure 13. The maximum latency is 1147ms, the median is 631ms which results to $(237/0.631) * 10 = 3756$ transactions during a single Burst block time (1.35 million tx per day).

Throughput

Besides latency, we measured throughput for the peer initiated to form a DL. Table 1 shows the results for the prototype implementation. The results are representative, but measuring throughput for long-distance node-node communication is naturally highly volatile, as time of day, global network load or regional ISP activity can have an influence.

"One-off" measurements handle one tx validation (connect, validate) then exit, so we can rule out any network caching effects, although we can assume for a long-running tangle better network throughput performance as peer connections will be in router caches.

The high performance between "Sydney" and "Tokyo" nodes is probably due to the fact that these - while geographically

min \ max	Germany	Prague	Tokyo	Sydney
Germany	–	22.30	98.70	68.60
Prague	11.70	–	3.82	3.24
Tokyo	81.80	3.26	–	152.00
Sydney	52.40	2.63	150.00	–

Table 1. one-off node-node connection throughput in Mbit/s

distributed - are operated by the same ISP and evidently have a dedicated connectivity.

The poor connectivity between the "Prague" node and the two "Sydney" and "Tokyo" nodes suggests that it is probably not a good idea to initiate a tangle from your home network. In fact, we assume tangle initiation will be done by large corporations or institutions with dedicated hardware and excellent connectivity.

Still, let's examine what throughput-limited performance we could expect if a tangle was to be formed between geographically distributed nodes such as in this case. The payload for an ordinary non-anonymous transaction is roughly 200byte. An anonymous ring signature takes up to 3 kB, while zk-SNARK tx is roughly 2kB.

Even in our worst-case scenario (Prague-Sydney), we can see that throughput is not a limiting factor. Theoretically, even a 2.63 Mbit/s throughput (336.64 kB/s) could allow for

- 1683 tx/s ordinary payments
- 168 tx/s RS-anonymous transactions
- 112 tx/s zk-SNARK-anonymous transactions

However, compared to the latency-limited 3756 tx per Burst block time - giving us "only" 15 tx/s for a pathologically small and excessively relocated tangle, throughput limits are not the issue.

5.2 Adoption Process

Burst has been around since 2014, yet it is still small enough to allow for significant improvements without a long and paralyzing scaling debate. It is natural if cryptocurrencies with large market capitalization have to be very conservative with their changes to the code base given the amounts at stake.

While Burst is more flexible in this aspect, as a cryptocurrency there are certain principles development must adhere to. Changes to the code base must have the approval of the majority of users, which in this case are wallet/node operators (not miners). Blindly forking the coin can result in community split and is undesireable in general.

Establishing The Burst Dymaxion will be an incremental process consisting of the implementation of several Burst CIPs (see section C - partially C). Each of these steps will carefully adhere to a well defined and transparent process with evident benefits resulting from its adoption.

Because the Burst Dymaxion is a layer on top of the Burst blockchain, virtually all essential components of Burst will remain unchanged and if there is a change, backward-compatibility is always considered the premier option (e.g. PoC2 CIP in section C)

If more rigorous changes are required with significant consequences (hard-forking, re-plotting), the cost-benefit evaluation must weigh significantly in favor to the benefit aspect.

It is evident, the Dymaxion as whole represents the most significant update to the Burst cryptocurrency. The claim is to provide a cryptocurrency capable of sustaining the *total* global load of non-cash transactions. To the best of our knowledge any Burst stakeholder section should root for this to happen.

In general, each change should undergo the CIP workflow (see also C). Each feature would have two thresholds that need to be met in order for it to become active: 1) an activation block height, which should be set far enough in the future, so every node operator has enough time to prepare for it, and 2) a defined percentage of signalling nodes to support this feature, which would by definition be Burst nodes capable of the feature in question.

Acknowledgments

The authors would like to thank many members of the cryptographic community for their support and valuable input. Anton Yip provided great insight into Burst CIYAM ATs and their applicability to ACCTs. Burstcoin community member Quibus took the time for a forensic dilligence of the Burst plotting and mining process. Also, his proposal of a backward-compatible and un-gameable PoC2 can be considered a milestone for Burst. Tom Créance (@Gadrah) helped with the visual representation of some figures. Sergey Blagodarenko for providing insight and code on the PoC1 gameability problem. We also would like to express our gratitude to the numerous authors of Bitcoin, IOTA, Monero and ZCash. Without the giants who kindly let us stand on their shoulders, we would not have had the building blocks necessary for the Burst Dymaxion.

Appendices

A SpaceMint Paper Errata

SpaceMint: A Cryptocurrency Based on Proofs of Space[15] is a paper about a proposed cryptocurrency named SpaceMint which is based on a ”proof of space” concept - some of the authors presented in another paper.[16]

In section 2 "Related Work", the authors make some claims about Burstcoin as it is the only coin implementing a Proof-of-Capacity consensus - somewhat related to the Proof-of-Space described in the paper. In particular they make 3 claims about Burst weaknesses:

- By requiring the examination of a constant fraction (0.24%) of reserved disk space, Burst is - according to the authors - inefficient compared to SpaceMint which requires only a logarithmically proportional amount of examination.
- Verification is problematic in Burstcoin, because “a miner has to verify 8 million blocks to verify another miners claim”.
- Burstcoin being susceptible to *time-memory tradeoffs*, thus allowing miners to mine using PoW and using just a small fraction of space to be at the same rate as “honest miners”.

We would like to clarify and refute some of these claims, because they seem to have originated in the SpaceMint authors' incomplete understanding of the Burstcoin PoC and even some simple arithmetic mistakes.

Ad “inefficient examination” It is correct, that Burstcoin requires per round - each block every 240 seconds - 1/4096th of the space reserved on disk to examine[16]. The authors also point out correctly this being 0.024%. Unfortunately in their model comparison with SpaceMint, using a 1TB mining space, they transform this 0.024% into 24 gigabyte of data to be examined for Burst, while SpaceMint allegedly requires only 24 megabyte to be examined.

Due to lack of availability we were not able to verify the claimed requirements for SpaceMint, but 0.024% of 1TB correctly translates into 240 megabyte, thus a factor of 100 lower than the authors' wrong comparison value. We therefore believe, that the claim of Burst inefficiency merely exists due to this error in the SpaceMint paper.

Ad “problematic verification” The authors avow that their assessment of the Burst plotting and mining process is only their best guess, based on the - admittedly - sparse and informal specification at the time of writing the paper. They base their claims mainly on the old, and even at the time of its publication not exact Mining/Plotting diagram.[17]

The biggest deviation from the actual situation results from the fact, that in order to compute one scoop (64 bytes), two SHABAL256 operations have to be performed, as each SHABAL256 delivers a 32byte (256bit) hash value only.

Because the amount of input data that is being given to the SHABAL256 is capped at 4096 bytes, it's also not " 8 million 256bit blocks" that are being hashed in total to get all scoops in a nonce, but exactly 33292288 bytes, therefore a total 1040384 of 256bit blocks. So roughly $\frac{1}{8}$th of the claimed value in the SpaceMint paper.

Ad “time-memory tradeoffs” As mentioned in the previous paragraph, many estimates of the authors are based on the insufficient information about how Burstcoin works at the time available to them. This results in certain follow-up errors that skew the computations. E.g. there are 8192 hashes computed and not 4097. So for the attack as described by the authors, there is actually only 1/8192th of the total plot space needed (the final 32 bytes to perform the XOR)[4].

Now with the current PoC used with Burstcoin, there indeed are time-memory tradeoffs possible, roughly as the authors describe. The Burstcoin developers have been aware of this possibility to use less *Capacity* and invest more *Work* for a specific range of Scoops (at this moment the highest 64 Scoops 4096 - 4032 are prone to this attack in a somewhat viable way).

This attack on PoC mining fairness was possible in theory, but not feasible economically, because the PoW required for this mode of operation consumed far more energy than a PoC mining style. However recent advancements in hardware and GPU performance do show, that economical feasibility is just a matter of time.

Therefore this situation must be addressed. A PoC2 CIP (see C) is underway to ensure the Burstcoin blockchain is not gameable in any way - not even with the advent of any theoretical SHABAL256 ASIC devices postulated by the SpaceMint authors.

Summary Burst is neither inefficient in plot examination during the mining process, nor is the load put on a verifier “too high”. We agree that there is a small theoretical possibility for “time-memory tradeoffs”, but as of this moment its security impact is low. Burst will undergo a series of upgrades and this issue will be addressed by a PoC2 CIP (see section C).

We would also like to add that reducing mining and verification effort is not necessarily the ultimate goal. The SpaceMint authors are aware of so called *Nothing-at-stake problems*:

> When replacing PoW with a different type of proof that is computationally easy to generate (such as PoSpace), a series of problems arise which are collectively known as nothing-at-stake problems. Intuitively, because mining is cheap, miners can (1) mine on multiple chains, and (2) try multiple blocks per chain, at very little additional cost. (3) These two problems potentially allow for double-spending attacks and slow down consensus.

We believe the Burst mining process is a premier example

[16] a so called *Scoop*

[17] See also `https://is.gd/bwPjCb`

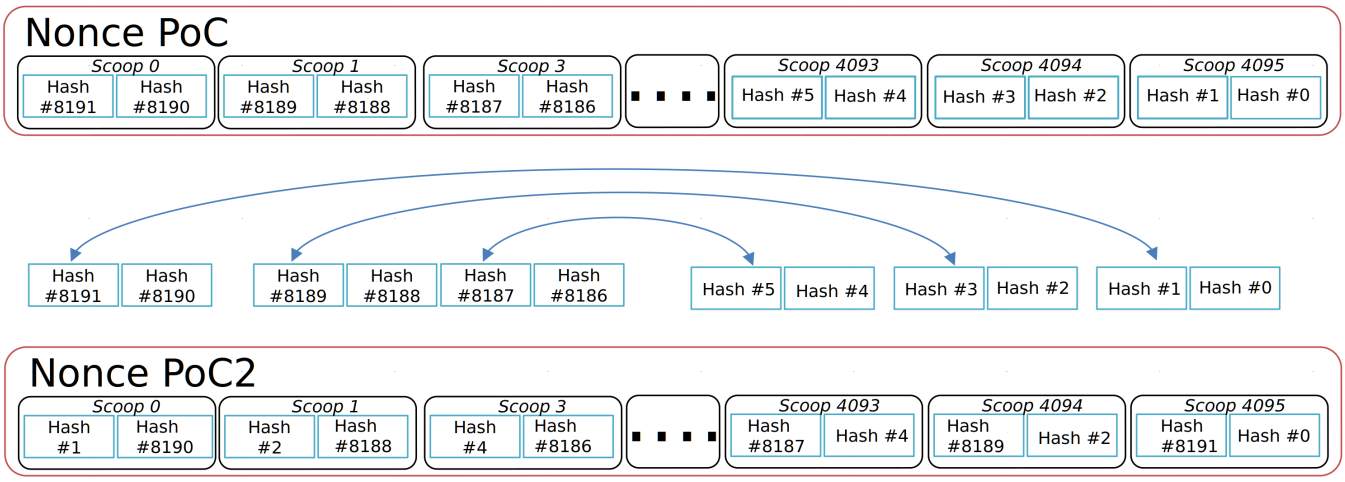

Figure 14. Proof-of-Capacity 2.0: backward-compatible plots with scoops consisting of interleaved SHABAL256 hashes. Preventing time-memory tradeoff for high-range scoops.

of an equilibrium between energy-efficiency and nothing-at-stake problem prevention.

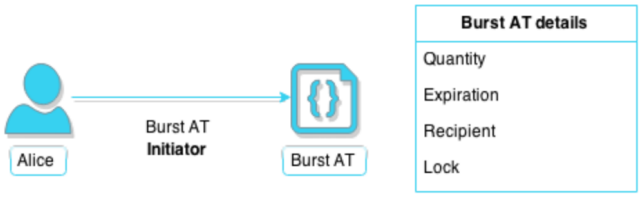

Figure 15. Burst/Qora ACCT step 1: Alice deploys an ACCT AT on Burst that will contain the responder's address, quantity, password and expiration time. Two hashes are made of the password: key and lock

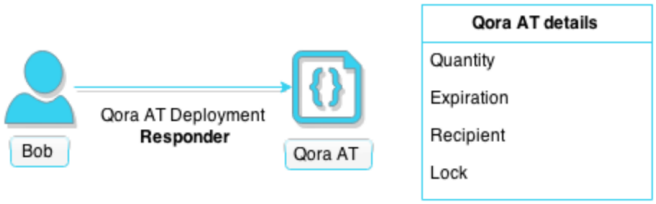

Figure 16. Burst/Qora ACCT step 2: Upon examining Alices AT, Bob deploys an ACCT AT on Qora with the initiator's address, quantity, lock sent from Alice and expiration time (less than than what Alice set).

B The Burst/Qora ACCT Process

The general process of an ACCT is described in section 1.5. For the concrete Burst/Qora realization, which to the best of our knowledge was the 1st ACCT ever realized, please see the depicted reference below.

Figure 17. Burst/Qora ACCT step 3: Alice sends after examining the AT from Bob, the key to the Qora AT.

- Alice (the Initiator) wants to trade Burst for Qora.
- Bob (the Responder) wants to trade Qora for Burst.

Now there are only 4 steps necessary to be provided by the AT mechanism running on both the Burst and the Qora chains - see figure 15 to 18. The communication between Initiator and Responder, like sending the details of the generated ATs and INitiator to Resipient key in case everything went well is of course done off-chain and the mode of operation depends on what communication channels are established (website, phone, face to face,...).

C Burstcoin CIPs

Burstcoin Capability Improvement Proposals[18] establish a process, defined by the Burst community, similar to BIPs[19] (Bitcoin Improvement Proposals) and EIPs[20] (Ethereum).

CIPs (short for "Capability Improvement Proposal" or even "Coin Improvement Proposal") are meant to advance further development of Burstcoin and describe proposed standards for the Burstcoin platform, including core protocol specifications, client APIs, nomenclature and contract standards.

[18] https://burstwiki.org/wiki/CIP
[19] https://en.bitcoin.it/wiki/Bitcoin_Improvement_Proposals
[20] https://github.com/ethereum/EIPs

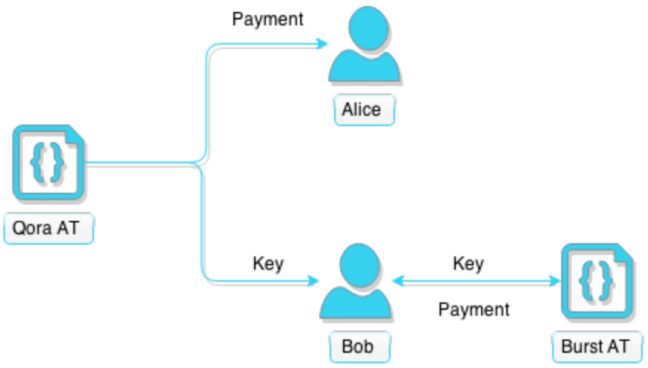

Figure 18. Burst/Qora ACCT step 4: The Qora AT sends the Amount to Alice's address and reveals the key to Bob for the Burst AT. Bob sends the key to the Burst AT and receives the payment.

What's not covered by CIPs are changes or improvements to the coin that can be done without any change to the protocol or API, such as UI or usability improvements. Improved wallet-UI or initializing a Burst address with a public key without an outgoing transaction fall into that category.

PoC2

The "time-memory tradeoffs" mentioned in the SpaceMint paper are real and show a weakness of the PoC (henceforth PoC1) consensus used in Burst. A proof-of-concept implementation exists (see figure 19) - improving its speed is only an engineering task. While they currently do not represent a fatal threat to Burstcoin (one can assume around 2% of all mining capacity going to dishonest miners), it is good practice for a cryptocurrency, as well as a sign of responsible development, to address such issues in a timely manner. Moreover a Burst blockchain being the backbone of many high-volume payment channels must address security issues even if they haven't yet crossed the threshold to practical applicability.

Therefore several Burst core developers discussed ways to address this problem in a way that would not only fix the issue, but also do so with minimum impact to the current stakeholders, in this specific case miners who are vested with a large capacity of plots.

The PoC2 proposal is a minimally invasive way to achieve time-memory tradeoff resistance, while keeping the currently used plots functional. Figure 14 shows the concept of hash interleaving to re-shuffle SHABAL256 hashes in scoops in a way so that each scoop represents an equal amount of hashing effort.

Software used for the mining process can operate on both PoC1 as well as PoC2 format, where PoC1 requires twice the reads compared to PoC2 and works on both optimized as well as unoptimized PoC1 plots.

For better PoC2 performance a PoC1 $\rightarrow$ PoC2 converter will be offered.

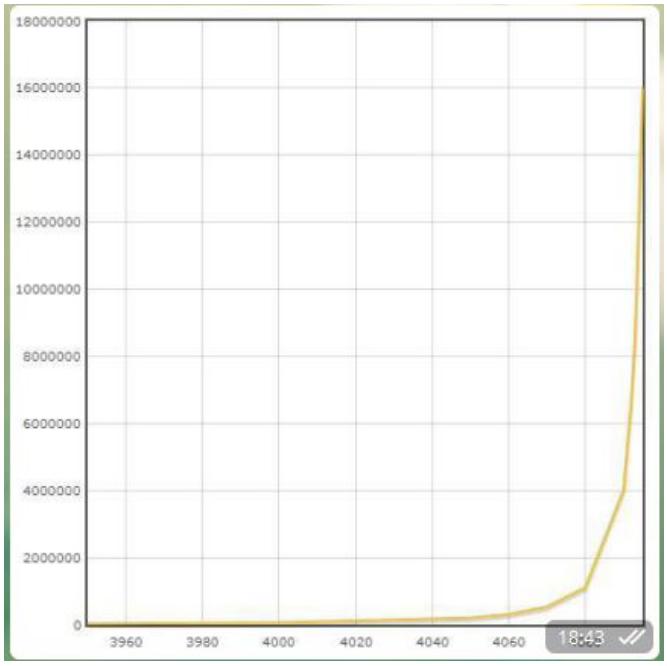

Figure 19. PoC1 time-memory tradeoff PoW miner speed. Y-axis shows nonces/minute on 1 CPU core, X-axis the nonce number. source: Sergey Blagodarenko

As can be seen in figure 20, PoC1 optimized plots provide significant advantages compared to unoptimized plots. The number of seeks of a unoptimized PoC1 plot is basically n-times the number of nonces in that plot, which can be in the millions.

Compared to this, the difference between seek times of a PoC1 plot interpreted as a PoC2 (unoptimized PoC2) and "native" PoC2 is merely a factor of 2, as for each PoC2 scoop, two PoC1 scoops have to be read (consisting of the two 32-byte SHABAL256 hashes).

This factor 2 applies to both unoptimized as well as optimized PoC1 plots.

Burst Units Nomenclature

Burst is divisible, similar to many other cryptocurrencies, into 100 million parts. Up to now, BURST had an equivalent value in the low cent range, so nomenclature of fractions of BURST seemed not important.

Similar to the nomenclature of Bitcoin units[21] we propose a simple naming scheme for the fractions of BURST - see table 2.

While the canonical SI-names milliBURST (mBURST) and microBURST (μBURST, or uBURST) are probably best used in technical documentation and protocol specifications, the more intuitive names like Burst-cent (BC $\rightarrow$ "Bessie") instead of centi-BURST or $\frac{1}{1000}$th of this (MilliBessie $\rightarrow$ "Maybel") can be used for human2human communication.

We denote the smallest unit of BURST as "Planck".

[21] https://en.bitcoin.it/wiki/Units

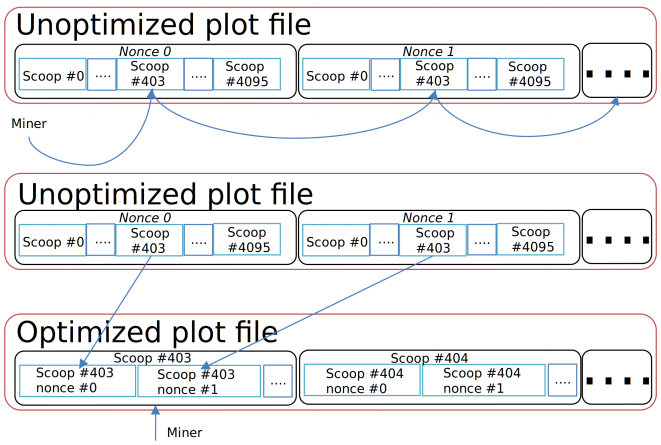

Figure 20. PoC1 unoptimized and optimized plots. Optimized plots reduce HDD seek time significantly, by an order of magnitude equal to the number of nonces present.

Decimal (BURST)	Canonical Name	Alternate Name
1.00000000	BURST	Burst
0.0**1**000000	cBURST	Bessie
0.00**1**00000	mBURST	–
0.00001000	–	Maybel
0.00000100	uBURST	–
0.00000001	–	Planck
0.12345678	*digit reference*	–

Table 2. Units of BURST and its fractions

Dynamic Block Size and Tx Cost

Currently, the minimum tx fee is 1 Burst and the maximum number of transactions that can go in a block is 255. Both these hard-coded values do work well only for a narrow parameter space of Burst value and transactional load on the network.

If we imagine a scenario where Burst rises to a value of 1, 10, 100 USD - or even BTC levels, an immutable transaction fee of 1 Burst would certainly be inhibitory to network usage. While that might not be a big problem in a future scenario where Burst is used as (valuable) collateral in opening and closing Dymaxion layers, there are other arguments against keeping these values as-is.

Hard-capping the lower bound of tx cost to 1 Burst effectively dismisses possible Burst microtransactions, because a transaction of 0.0012 Burst with transaction cost of 1 Burst is illogical.

On the other hand, simply lowering transactional cost could lead to spam attacks on the network. The low value of Burst together with fixed cost per transaction, independent of transaction size in bytes, already did enable a spamming attack in July 2017 disturbing network operation significantly.

At this moment, the transaction fee possible for one block are in the range from 0 Burst (for empty blocks) to 255 000 Burst (for a block filled with 255 generated assets). There is no

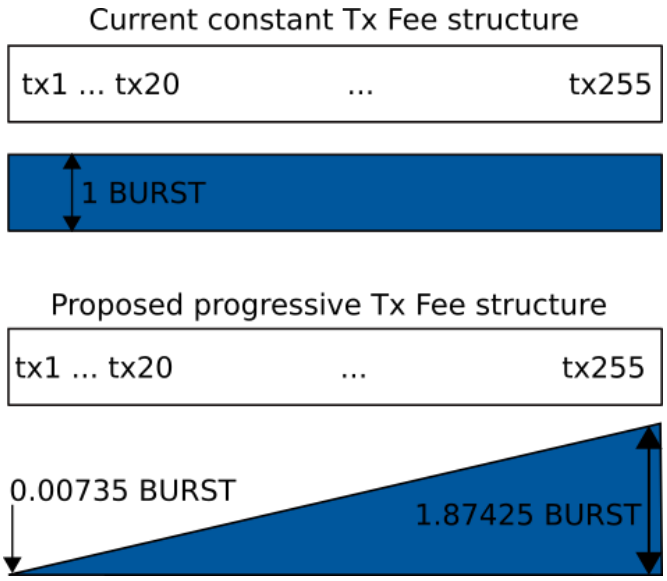

Figure 21. Comparison constant and linear-progressive fee structure inclusion guideline. A linear-progressive fee structure does allow for microtransactions, while preventing spamming and ensuring miner rewards in same height as in the constant-fee system under high-load conditions.

way tx fees could be above this upper limit and there is no way - even if users would like to spend more than the minimum of 1 BURST per transaction - to get more than 255 transactions into a block.

We propose to make transaction cost and block size dynamic values to be better able to cope with varying transactional load on the Burst blockchain.

Given the Burst blockchain is stored in a relational database, varying block size is not much of an issue. Yet, space on the blockchain is to be considered a scarce, thus precious resource. Cost of a transaction should therefore depend on how much space it will occupy within a block and thus on the blockchain.

Furthermore, the cost of a transaction should depend on network load, which by extension means fill state of forged blocks. Currently values for the number of transactions and the maximum block payload are hard coded in the burst wallet source:

```
MAX_NUMBER_OF_TRANSACTIONS = 255;
MAX_PAYLOAD_LENGTH =
   MAX_NUMBER_OF_TRANSACTIONS * 176;
```

If we assume a regular block filled with the current max. capacity of 255 ordinary payments, the total tx fee would be 255 Burst and the payload would be $255 * 176 = 44800$ bytes.

This means, the current protocol expects - roughly, without headers - to be paid 1 BURST per 176 byte of tx payload. Compared to this, the bitcoin network[22] offers a fee structure of

[22] https://bitcoinfees.earn.com/

1 to 430 Satoshi per byte, with best-case 0-block delay starting at 270 Satoshi/byte.

As of December 2017, the transaction cost structure comparison between Burstcoin and Bitcoin (1 BURST = ca. 150 Sat), shows 0.85 Satoshi/Byte in the Burstcoin network and therefore a roughly 317x higher cost for Bitcoin block space and ca. 900x higher cost for network transactions, as a Bitcoin transaction is roughly 500 bytes in size[23].

In Fiat, this means that at a price of around 3 US-cent per BURST, one byte of payload costs 0.017 cent in the Burstcoin network and around 5.4 cent in the Bitcoin network.

If the hardcoded tx cost for the Burst network was to remain at 1 BURST minimum, Burst would reach the same level of tx cost that Bitcoin has today at a price of $176 * 5.4 = 950.4$ US-cent, therefore around $9.50.

Tx no.	Tx fee	Total fees
1	0.00735	0.00735
100	0.73500	37.11750
255	1.87425	239.90400
510	3.74850	957.74175
765	5.62275	2153.51325
1020	7.49700	3827.21850

Table 3. Progressive Tx fee reference: per-tx fee and total block tx-fee for various number of transactions in a maximum 1020 tx/block model.

If we look at todays price levels, Burstcoin miners were ensuring the network for a maximum payment of 91800 Burst daily. On average, the network has been doing around 5000 tx/day. So in addition to the block reward, the transaction fee reward per day, for the whole Burstcoin network is around $150 in December 2017.

In order to cope with future development of BURST price and transaction volume, we propose a progressive tx fee structure guideline, where wallets can decide and priorize what transaction to include in the current block depending on the fill-state of the current block and memory pool backlog - see figure 21. The area of the blue rectangle and triangle is roughly the same (255 vs. 240). In order to be included in the next block a transaction currently in mempool must provide a higher tx fee than the currently free slot in the block requires. If the transaction can fulfill this requirement, it is included in the block, if it can't it will wait in mempool for the next block, where this process repeats.

The progressive approach opens the door for more on-chain scalability. Instead of limiting the max. number of transactions to 255, we could now theoretically have an open limit and solve much of the scalability issues blockchains have *per se*. Instead, we propose a conservative extension of the max. number of tx.transactions to 1020 (4-fold). With the same linear progressive rule, the reference values for a 1020-tx "triangle" are shown in table 3.

[23] https://en.bitcoin.it/wiki/Transaction_fees

Together, these two modifications (linear-progressive fee and max 1020 tx/block), would ensure that nominally, miner earnings will remain the same. "Nominally" meaning same network load, same price.

Moreover micro-transactions should be possible and under high-load conditions miner profit would be a multiple of what it is now. By enabling limited micro-transaction capability on-chain now, Burst would open to new applications and markets. Still, with the progressive fee structure spamming would be out of the question and financially strong market participants could basically "throw money at the problem", when e.g. a financial institute would need to open a Dymaxion layer in the current block, it most probably could do so at a premium.

Dynamic Node Capabilities

The network consisting of P2P nodes will never be homogenous as hardware capabilities of the nodes will always differ. In our opinion it is not desireable to make artificial (1-dimensional) distinction between so-called "super-nodes" and "regular-nodes" as some cryptocurrencies do. Neither does it seem to be the right design decision if some minimum capability requirements on nodes are imposed to all nodes (such as available memory) thus leaving potential capacities unused.

We believe a fine-grained configurability of nodes, extending on current principles, has to be implemented. While today the BRS configuration options allow to allocate number of CPUs - or enabling GPU support to the processing capacity and also define how much network traffic and peers a node is willing to cope with, more fundamental settings are not possible.

In the wke of the spam attacks, limits on mempool have been hard-coded in the wallet. While this proved to be a very effective measure against memory DoS attacks - what the spam attack effectively was - it prevents nodes with higher resources to make use of their full potential to support the network. We propose the mempool size being configurable to better adjust to the node capabilities.

Other parameters, similar to packet introspection of routing protocols but in this case applying to parameters of transaction datagrams, could define policies for nodes, e.g. which transactions to relay and which not. This is common practice e.g. in the Bitcoin network, where a node can decide to support low-fee transactions (by forwarding them) or not.

We believe a structure in the network will evolve from these parameters, better adapted to the underlying node capabilities - or what node operators are willing to provide - forming naturally a hierarchy of nodes with backbones and super-backbones emerging from that.

On the other end of the scale it will also allow nodes to participate which are not capable of doing so today. Very small embedded devices from the ubiquitous IoT, that can support the network serving even merely as repeaters with very little resource requirements.

In section C we talked about transactions that have to remain in mempool in case they could not be included in the current block. Because nodes have limits for mempool size, transactions with the lowest fees are being discarded from mem-

pool first if maximum mempool size is reached. Acting as a spam prevention, this also ensures a priorization of high-fee priority transactions.

Nodes with more resources and higher mempool storage parameters will still have these low-fee transactions in mempool while others may have already discarded them. So while it will not be impossible for a low-fee transaction to "survive" until it gets included in a block (dip in traffic volume), tx-fee is certainly proportional to chance of being included in a timely manner or at all.

Dymaxion

The Burst Dymaxion is not only the most significant update to the Burst cryptocurrency, it is the biggest technological update any cryptocurrency has ever received. Each of the components features a full outstanding protocol feature by itself, so establishing the Dymaxion as a whole needs to be done as an incremental process, split up into several separate CIPs:

- adding ring signatures library
- adding zk-SNARKs library
- adding parts of the IOTA iri
- adding ACCT (ACTT) templates

As mentioned earlier, all of these components are already in place, and some may be at the time of the publication part of the BRS repository. The integration and enabling of these features will be subject to a community-approved roadmap defined by block-height and adoption rate (percentage of supporting wallets in use) for the features as described in section 5.2.

Also the features themself will be establishen in an incremental process. While there is support for all major consensus algorithm types, only PoC - as already present - will be enabled in the first instance of the Dymaxion.

Post-Dymaxion: PoC3

The advantages of PoC in comparison to PoW or PoS have been mentioned and examined under multiple perspectives already. While PoC and its designated successor PoC2 are energy-efficient, which in our opinion is the only sustainable way for a cryptocurrency with global impact, the space used for the PoC consensus is of no other use than to perfom mining and tx validation for the Burst blockchain.

Critics have pointed out that the disk space used for Burst is "lost" or "wasted" otherwise as plots are not really usable for anything else. This is formally true and because of this we propose establishing a Proof-of-Capacity 3.0 consensus some time after the Dymaxion comes to full effect.

This PoC3 will exist in parallel to PoC2. It will be based on dual-use data instead of the Burst mining-only plots of PoC and PoC2. Dual-use means real-world data, like movies, audio, Wikipedia archive files, OpenStreetMap GIS data and more. In general, large immutable files of permanent interest to all - not the private word document, holiday picture or browser cache.

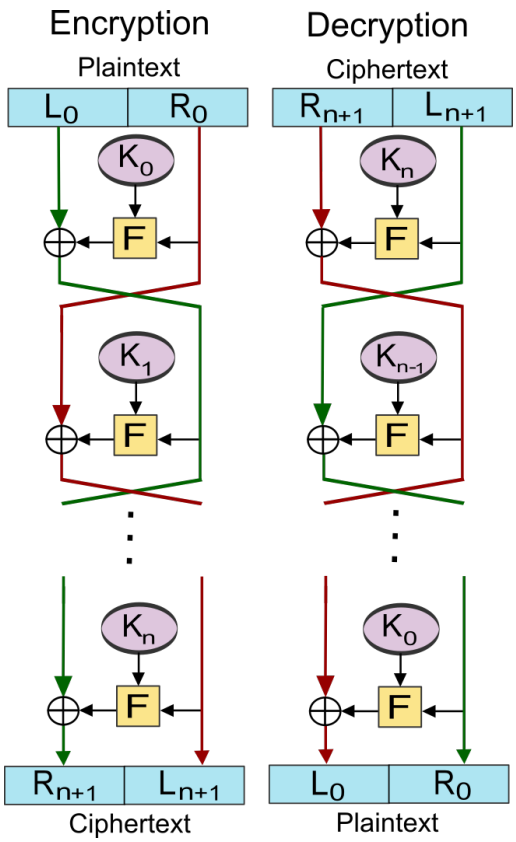

Figure 22. Symmetric Feistel cipher transformation for individualization of a PoC3 plot file.[‡]

The protocol would allow these files to be announced to the network (SHA256, size) and voted on by nodes for validation inclusion. If a very high percentage (say 95% threshold) of the nodes would vote in favor of adding these files to the pool of "dual-use plots" which also implies storing these files themself, each node would find deadlines in these files based on a succinct test of values of a virtual hard-to-pebble[17] **trees** laid over individualised versions of these files. Once in the pool of accepted plots, PoC3 plots would need to remain there, because while for mining their presence is optional, for ab-initio validation of the blockchain when resyncing their presence is mandatory.[24]

First, each PoC3-accepted file would undergo a Feistel cipher transformation (fig. 22) with the numeric Id of an account to individualize it as plot and counter grinding attacks. The mining process would assume the file being mapped on a RB-Tree(figure 23) layout as seen in figure 24.

Let the basic operation of the Feistel encryption with the round function F and sub-keys $K_0, K_1, \ldots, K_n$ for the rounds $0, 1, \ldots, n$ be as follows:

Split the plaintext block into two equal pieces, (L_0, R_0), in our case two 32-byte chunks.

[24] For the future, we can picture a situation with a distinction of "full PoC nodes" containing all PoC3 and being able to make such a resync and "restricted PoC nodes" being able to sync blockchain from the full nodes

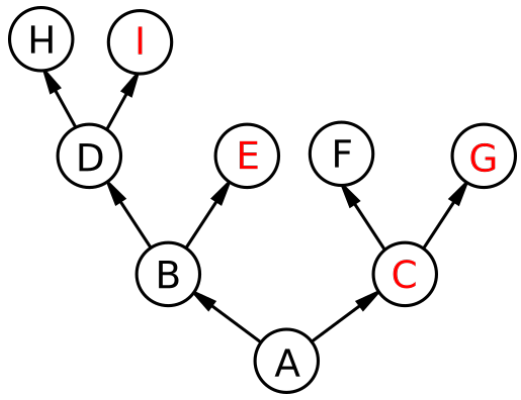

Figure 23. Red-Black binary tree structure for traversing PoC3 plots. Each node being a 64-byte data chunk, each depth represents next level as depicted in the data layout of the next figure.

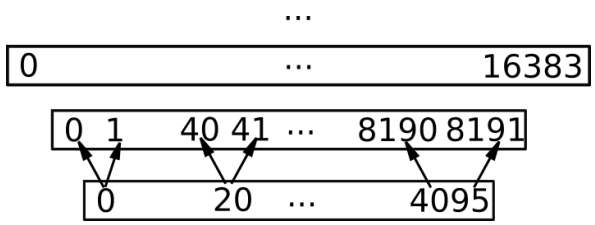

Figure 24. Data layout of a PoC3 "plot". Each level twice the size, maximum 64 levels allow for a plot size of up to 2 YiB (yobibyte).

For each round $i = 0, 1, \ldots, n$, compute

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus F(R_i, K_i).$$

Then the ciphertext is $R_{n+1}, L_{n+1})$. Decryption of a ciphertext (R_{n+1}, L_{n+1}) is accomplished by computing for $i = n, n-1, \ldots, 0$

$$R_i = L_{i+1}$$
$$L_i = R_{i+1} \oplus F(L_{i+1}, K_i).$$

Then (L_0, R_0) is the plaintext again. Of course by "plaintext" in cryptography nomenclature we mean the more generic term *original data*.

Similar to the current Burst mining process, a scoop would define the starting point in the data layout (index in the lowest level). The algorithm would traverse the RB tree of 64-byte scoops, where the path decision (start, then red or black for each level) would simply be defined by the bit sequence of the previous block-id 1=B, 0=R.

A maximum of 64 steps allows for a maximum PoC3 plot file size of 2 YiB (yobibyte). The values along the traversed path would be concatenated and hashed. All values along the path as well as their hash value are then broadcast to the network, where any nodes can take over the validation of the submitted data. As the length of the path defines the size of the underlying plot in a logarithmic way, also the position of a submitted deadline on this path modifies this deadline logarithmically (in the case of this binary tree structure this resolves to simply halving the value for each level $n : v = \frac{1}{2^{n-1}}$, so $\frac{1}{2}$ for level 2, $\frac{1}{4}$ for level 3 etc.

Using Feistel ciphers allows for individualization, yet lossless retrieval of arbitrary data. The symmetry guarantees the cipher being of same length as the original data, paving the way for a dual-use Proof-of-Capacity consensus.

The Burst blockchain will thus not only form the fundamental layer for a truly global transaction network, it can also take over a custodian role in globally distributed redundant storage. This means it can be used for the safe preservation of all information that has been acquired by our civilization and that is of permanent interest.

References

[1] Serguei Popov. The Tangle. https://iota.org/IOTA_Whitepaper.pdf, 2017. [Online; accessed 21-October-2017].

[2] N. Petkov. *Systolic Parallel Processing*. Elsevier Science Inc., New York, NY, USA, 1992.

[3] John Ratcliff. The Lightning Network Glass is Half Full Post. http://codesuppository.blogspot.com/2016/03/the-lightning-network-glass-is-half.html, 2016. [Online; accessed 22-October-2017].

[4] Quibus. Technical information about mining and block forging. https://burstwiki.org/wiki/Technical_information_about_mining_and_block_forging, 2017. [Online, accessed 27-November-2017].

[5] CIYAM Developers. Automated transactions documentation index. http://ciyam.org/at/. [Online, accessed 16-November-2017].

[6] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin (extended version).

[7] Nicolas van Saberhagen. Cryptonote v 2. 0. *HYPERLINK https://cryptonote.org/whitepaper.pdf*, 2013.

[8] https://z.cash/technology/zksnarks.html, 2017. [Online, accessed 27-November-2017].

[9] Christian Reitwiessner. https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/, 2016. [Online, accessed 27-November-2017].

[10] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network:scalable off-chain instant payments. https://lightning.network/lightning-network-paper.pdf.

[11] Meni Rosenfeld. Overview of colored coins. https://bitcoil.co.il/BitcoinX.pdf, 2012.

[12] Nick Johnson. Why i find iota deeply alarming. https://hackernoon.com/why-i-find-iota-deeply-alarming-934f1908194b.

[13] Ethan Heilman, Neha Narula, Thaddeus Dryja, and Madars Virza. Iota vulnerability report: Cryptanalysis of the curl hash function enabling practical signature forgery attacks on the iota cryptocurrency. https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md.

[14] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society, 2013.

[15] Sunoo Park, Krzysztof Pietrzak, Albert Kwon, Joël Alwen, Georg Fuchsbauer, and Peter Gazi. Spacemint: A cryptocurrency based on proofs of space. *IACR Cryptology ePrint Archive*, 2015:528, 2015.

[16] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. Proofs of space. *IACR Cryptology ePrint Archive*, 2013:796, 2013.

[17] Vivek Bhupatiraju, John Kuszmaul, and Vinjai Vale. Exploring proof of space with hard-to-pebble graphs. https://math.mit.edu/research/highschool/primes/materials/2016/conf/10-2%20Bhupatiraju-Kuszmaul-Vale.pdf, 2016.

Licenses

† original file: Theymos from Bitcoin wiki vectorization: Own work (https://commons.wikimedia.org/wiki/File:Blockchain.svg), „Blockchain“, modified, https://creativecommons.org/licenses/by/3.0/legalcode
‡ Feistel_cipher_diagram.svg: Amirki derivative work: (https://commons.wikimedia.org/wiki/File:Feistel_cipher_diagram_en.svg), „Feistel cipher diagram en“, modified, https://creativecommons.org/licenses/by-sa/3.0/legalcode

Changelog/Errata

- initial version