Bobtail: A Proof-of-Work Target that Minimizes Blockchain Mining Variance

George Bissias Brian N. Levine
College of Information and Computer Sciences, UMass Amherst

ABSTRACT

Blockchain systems are designed to produce blocks at a constant average rate. The most popular systems currently employ a Proof of Work (PoW) *algorithm as a means of creating these blocks. Bitcoin produces, on average, one block every 10 minutes. An unfortunate limitation of all deployed PoW blockchain systems is that the time between blocks has high variance. For example, 5% of the time, Bitcoin's inter-block time is at least 40 minutes. This variance impedes the consistent flow of validated transactions through the system. We propose an alternative process for PoW-based block discovery that results in an inter-block time with significantly lower variance. Our algorithm, called Bobtail, generalizes the current algorithm by comparing the mean of the k lowest order statistics to a target. We show that the variance of inter-block times decreases as k increases. If our approach were applied to Bitcoin, about 80% of blocks would be found within 7 to 12 minutes, and nearly every block would be found within 5 to 18 minutes; the average inter-block time would remain at 10 minutes. The cost of our approach is a larger block header.*

1 INTRODUCTION

Blockchain systems are designed to produce blocks at a constant average rate. The most popular systems currently employ a *Proof of Work (PoW)* algorithm as a means of creating these blocks. Bitcoin [25] (and Bitcoin Cash [1]) produce, on average, one block every 10 minutes, and will self-adjust the difficulty of producing a block every two weeks if too many or too few have been produced. Unfortunately, a limitation of all deployed PoW blockchain systems is that the time between blocks has high variance and the distribution of inter-block times has a very long tail. For example, 5% of the time, Bitcoin's inter-block time is at least 40 minutes. This variance impedes the consistent flow of validated transactions through the system.

The high inter-block time variance is a direct consequence of the PoW algorithms that are at the core of blockchains, including Bitcoin [1, 25], Litecoin [2], and Ethereum [17]. In all these systems, generally, the miners repeatedly craft block headers by changing a nonce until the hash of the header is less than a target value t. In other words, the hash of each header is a sample taken randomly from a discrete uniform distribution that ranges between $[0, S]$, where $S = 2^b - 1$ and typically $b = 256$. A block is discovered when the *first order statistic* (i.e., the minimum value) of all sampled values is less than target $0 < t < S$.

In this paper, we propose an alternative process for PoW-based block discovery that results in an inter-block time with significantly lower variance. Our algorithm generalizes the current algorithm by comparing the mean of the k lowest order statistics to a target. We show that the variance of inter-block times decreases as k increases. For example, if our approach were applied to Bitcoin, about 80% of blocks would be found within 7 to 12 minutes, and nearly every block would be found within 5 to 18 minutes; the average inter-block time would remain at 10 minutes. The cost of our approach is a larger block header. We call our approach *Bobtail*[1] mining.

Problem Statement. Consider a fixed interval of time during which the entire network produces θ hashes generating a sequence of hash values $\mathbf{Z} = Z_1, \ldots, Z_\theta$. Let Z be an arbitrary random variable from the sequence $\mathbf{Z}$; note that $Z \sim \texttt{Uniform}(0, S)$. Define V_i to be the ith lowest order statistic of $\mathbf{Z}$, i.e. $V_i = Z_{(i)}$ in standard notation. And let random variable W_k be the mean of the k lowest order statistics:

$$W_k = \frac{1}{k} \sum_{i=1}^{k} V_i. \tag{1}$$

W_k constitutes the collective mining proof (*proof*, for short) for the entire network. Our Bobtail mining criterion says that a new block is discovered when a realized value of W_k meets the target t:

$$w_k \leq t. \tag{2}$$

Notably, this approach is a generalization of current systems, which are the special case of $k = 1$.

Our primary goals are therefore to show, given values of $k > 1$, that: *(i)* there is a significantly reduced inter-block time variance; and *(ii)* the costs are relatively small, which include an increase in block header size and a slight increase in network traffic.

Contributions.

- We derive the statistical characteristics of our approach and validate each empirically. For example,

[1] A *bobtail* refers to an animal's tail that is unusually short or is missing completely [9].

Layer: Layer 2 ideas
Title: Multi-hop payment packetization on Lightning Network channel
Author: Takaya Imai <takaya.imai@unitedbitcoiners.com, takaya.imai@frontier-ptnrs.com>

[Abstract]

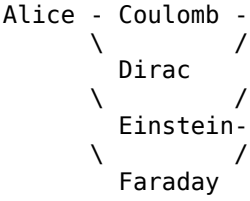
Lighting Network(LN) was invented by Joseph Poon and Thaddeus Dryja in 2015 \cite{ln}.
LN makes Bitcoin very scalable by lower transaction fee and rapid bitcoin transfer.
There are four major LN products \cite{products}, lit, lnd, c-lightning and eclair,
and RFC \cite{bolt}. These has been developped steadily and LN can be extended and
connected (atomic swap) to other blockchains simply.

LN Routing algorisms like flare \cite{flare} and gossip is proposed and developed but
LN has two problems.
One is that it is easy to be centralized. Some whose channel has large money as a
deposit can dominate a whole LN. This depends on routing algorism though.
Flare is one of proposals for a routing algorism but nothing about the centralization.
The other is that nodes in the middle have a possibility to take a long time to get
money if a preimage does not propagate successfully.

I propose one solution. That is a packet-like payment.

[Anti-centralization]

This is a way to reduce centralization of LN and provide higher processing ability of
payment by distributed nodes on whole LN.
Consider to send 10,000 satoshis from Alice to Bob on the following channel network.

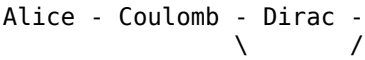


Alice can send at once through Coulomb but this case promote a centralization because
only nodes which have enough deposit on a channel can become one of nodes on a route.
So Alice divides a big payment into small payments such as 10,000 satoshis into 2,500
satoshis * 4 etc.

Even if a channel between Dirac and Bob has 4,000 satoshis as a direction of Dirac to
Bob, this channel can join this payment.
I think smaller amount is better but it might need excess payments so it is important
to keep a balance between process overhead loss and decentralization.

[Payment damage reduction]

This is a way to reduce damage in case that a preimage propagation problem occurs.
Consider to send 10,000 satoshis from Alice to Bob on the following channel network.



Catch-up Mining

Cyril Grunspan
Léonard de Vinci Pôle Universitaire
Research Center
Paris-La Défense Cedex, France
Email: cyril.grunspan@devinci.fr

Ricardo Pérez-Marco
CNRS, IMJ-PRG
Labex Refi , Labex MME-DDII
Paris, France
Email: ricardo.perez.marco@gmail.com

September 24, 2017

Abstract. The rules of the Bitcoin protocol are enforced by economic self-interest. We prove that the protocol is unstable when a miner has over 43% of the total hashrate. In this situation, the economic self-interest of the miner is not always to accept the longest mined chain. Under adequate circumstances, it can be profitable to mine a shorter chain hoping to catch-up the main chain (catch-up mining strategy).

Keywords: Bitcoin, blockchain, mining, proof-of-work, gambling problem, Dyck paths

1. Introduction

Decentralization of the Bitcoin protocol [N] requires that the rules of the protocol cannot be enforced by a regulatory body. Instead, they are enforced by the economic self-interest of the participants in the network: Any deviation from the rules of the protocol must be economically penalized.

One of the principal rules is that miners must adopt newly mined blocks and mine only on top of the longest chain[1]. At first look, and this seems to be a widespread belief, a miner with less than 50% of hashrate[2], seems to not have any interest in mining on top of a shorter chain hoping to surpass the chain accepted by the majority of the network. The reason for this belief is that he will be wasting his hashrate chasing the network longest chain. But this is false.

Theorem 1. *For a miner with more than 43% of the total hashrate it can be profitable to mine a chain that is one or more blocks shorter.*

We assume that the rogue miner has a hashrate less than 50%, otherwise he can mine any shorter chain and take over any longer chain in the long run. The reasons behind the theorem is that although the probability of mining 2 or more blocks before the rest of the network is small, the reward for taking over the main chain is also larger since the miner reaps up the reward (and fees) of the invalidated blocks. Thus this strategy has a positive Expectation Value (EV) when there are enough blocks to invalidate, or the sum of blocks fees and rewards surpass a certain threshold.

Note that this rogue behaviour, that we call "catch up mining", is different from selfish mining when the miner keeps a mined block to himself and starts mining on top of it.

We study the general problem of when it makes economic sense to try to catch up m blocks from the main chain. We study $E_n^m(v)$ which is the EV of the optimal strategy in order to catch up a delay of m blocks in n validation rounds and v is the pay-off.

The function $v \mapsto E_n^m(v)$ is a convex increasing affine by pieces

$$E_m^n(v) = \sum_l \pi(l)\,(v - v(l))_+ \tag{1}$$

where the sum runs over all winning paths shorter than n, $\pi(l)$ is the probability of the path l to occur, and $v(l)$ is the « minimal reward » over the path l.

Let $v_n^m = \inf_l v_n^m$. The sequence v_n^m is non-decreasing on m and decreasing on n. The strategy is profitable if and only if $v > v_n^m$.

1. "longest chain" meaning the chain with most work.
2. With more than 50% of hashrate the protocol is well known to be unstable

Bitcoin **Logic: The mathematical path to an universal financial logic**

Yiannis Kiouvrekis
Athens, Galatsi, Kiknon 67, 11146
(+30) 6944347515 | yiannis.kiouvrekis@gmail.com | john.kioubrekis

Scaling Bitcoin

September 25, 2017

Stanford

Abstract

Many experts in the field of Digital Currency and Computer Science in general have highlighted the need for convergence with the field of Formal Methods.
The article of Herihy and Moir, *Blockchains and the Logic of Accountability: Keynote Address*, highlights the necessity of creating a logical system that can express propositions such as “Because A (an agent) endorsed false statement, A can no longer be trust with nuclear codes” and properties like authorization, fairness, incentives as well as behaviors of miners.
In his presentation *How Formal Analysis and Verification Add Security to Blockchain-based Systems* during the last Blockchain Protocol Analysis and Security Engineering Conference 2017 Shinichiro Matsuo (MIT), reports the need for a logical system that is sound and complete, and within which we can describe the notion of security, privacy and thus prove the desirable security specifications.
A first step was made by Brunnler et al with blockchain epistemic logic, but as described by the writers, it is at a rather early stage,. In addition Joseph Y. Halpern and Rafael Pass in *A Knowledge-Based Analysis of the Blockchain Protocol* provide a complete characterization of agent’s knowledge. With this work our goal is:
First, to examine whether it is possible to create a logic that can describe properties such as privacy, security or Common Prefix Property, Chain Quality Property e.t.c. Can there be only one logical system, a universal logic? What knowledge do we need from model theory and mathematical logic in order to be sure that we can express properties in the logical system.? Furthermore, we should question the implications of other properties of a logical system like Craig’s interpolation property and compactness property.
Moreover, what knowledge can we get from the community of formal methods and the work done on the expressiveness of properties such as privacy in logical systems? The second thing we intend to explore is the relationship between Bitcoin, Game Theory and Mathematical Logic. Can we translate the fundamentals of Bitcoin Game in Formal Languages? It is known that the tools of modal logic have enriched the game-theoretic language by making it possible to express concepts that were previously either informally claimed to be captured by a solution concept. A famous theorem of this theory is the notion of com-

Proposal: Weak-Signal Radio Communications for Bitcoin Network Resilience

Nick Szabo, Elaine Ou
{nick, elaine}@globalfinancialaccess.com

Censorship-resistant bandwidth is crucial to maintaining a functional Bitcoin network. Prior work by Apostolaki, et al.has demonstrated Bitcoin's vulnerability to routing attacks, where a BGP hijack or traffic interception can isolate parts of the network [1]. More recently, the Chinese Network Bureau has threatened to block Bitcoin network access by employing deep packet inspection through internet service providers [2].

The Blockstream Satellite was recently deployed to broadcast the Bitcoin blockchain [3], enabling users to receive blocks without incurring the bandwidth cost of running a full node. However, the Blockstream Satellite represents a single fallible source of information, and currently serves as only a one-way relay.

In this work, we explore the feasibility of using high-frequency (HF) radio transmissions to complement satellite receiver nodes for full-node network access. HF radio broadcasts have a range of hundreds of kilometers. For SPV clients, HF radio transmissions can provide sufficient bandwidth to broadcast merkle blocks and signed transactions, giving users the ability to participate in the network without any internet connection at all. Furthermore, the radio transmission of block headers by full nodes can provide early detection of network partitions, preventing the execution of potential exploits.

1 Bitcoin Routing Attacks

BGP routing vulnerabilities have been explored by Apostolaki, et al. [1]. Internet traffic associated with different IP prefixes is exchanged between neighboring networks, or Autonomous Systems (AS). Examples of ASes include broadband service providers such as Comcast or China Telecom, or cloud providers like AWS. In China, there are basically only two state-owned providers, China Telecom and China Unicom. Parts of the Bitcoin network are heavily centralized from a routing perspective.

Routing centralization can be exploited through the isolation of some subset of the network, leading to the creation of two different versions of the blockchain. Individual nodes are also vulnerable to an eclipse attack, which could delay the overall propagation of blocks towards the victim node [5]. For example, a

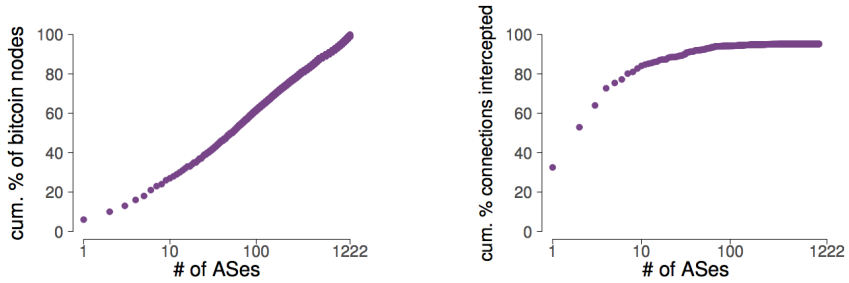


Figure 1: Left: Only 50 ASes host 50% of the Bitcoin network. Right: 3 ASes intercept 60% of all possible Bitcoin connections [1].

Bolt: Anonymous Payment Channels for Decentralized Currencies

Matthew Green
Johns Hopkins University
Baltimore, Maryland
mgreen@cs.jhu.edu

Ian Miers
Johns Hopkins University
Baltimore, Maryland
imiers@cs.jhu.edu

ABSTRACT

Bitcoin owes its success to the fact that transactions are transparently recorded in the blockchain, a global public ledger that removes the need for trusted parties. Unfortunately, recording every transaction in the blockchain causes privacy, latency, and scalability issues. Building on recent proposals for "micropayment channels" — two party associations that use the ledger only for dispute resolution — we introduce techniques for constructing *anonymous* payment channels. Our proposals allow for secure, instantaneous and private payments that substantially reduce the storage burden on the payment network. Specifically, we introduce three channel proposals, including a technique that allows payments via untrusted intermediaries. We build a concrete implementation of our scheme and show that it can be deployed via a soft fork to existing anonymous currencies such as ZCash.

ACM Reference format:
Matthew Green and Ian Miers. 2017. Bolt: Anonymous Payment Channels for Decentralized Currencies. In *Proceedings of CCS'17, Oct. 30–Nov. 3, 2017, Dallas, TX, USA.,* , 17 pages.
https://doi.org/10.1145/3133956.3134093

1 INTRODUCTION

Bitcoin has become increasingly popular as a decentralized electronic currency. In Bitcoin, each transaction is recorded in the *blockchain*, a public transaction ledger maintained by a set of decentralized peers. While this design has proven successful at low transaction volumes, the reliance on a globally-shared ledger has caused serious scaling issues. Since in Bitcoin 1MB blocks are added to the blockchain every ten minutes on average, the Bitcoin transaction rate is limited to fewer than ten new transactions per second across the entire Bitcoin user base [1].[1] Several proposals to increase blockchain bandwidth are being debated in the Bitcoin community today, but none are likely to produce a transaction rate that competes with centralized services such as payment card networks.

A promising approach to addressing the scaling problem is to move the bulk of Bitcoin transactions *off chain*, while preserving the system's decentralized structure and strong integrity guarantees. The leading proposal for off-chain payments is to use *payment channels*, exemplified by the Lightning Network [45] and Duplex Micropayment Channels [30]. Rather than posting individual payment transactions to the blockchain, channels employ the blockchain to first establish a shared deposit between two parties. The parties interact directly to make payments — adjusting the respective ownership shares of the deposit — and communicate with the blockchain only to agree on the final split of escrowed funds. In cases where no direct payment channel exists between two parties, these proposals also allow participants to route transactions via intermediate peers [45]. The main benefit of the payment channel paradigm is that it dramatically reduces the transaction volume arriving at the blockchain, without adding new trusted and centralized parties.

While payment channels offer a solution to the scaling problem, they inherit many of the well-known privacy weaknesses of Bitcoin [40, 46]. Although payments are conducted off chain, any party may learn the pseudonymous identities and initial (resp. final) channel balances of the participants. More critically, payment channels provide few privacy protections against transaction counterparties. By establishing a channel to pay for *e.g.,* Tor bandwidth or web content, a user implicitly links each payment on a given channel to all of her other payments on this channel. This is particularly problematic in the likely event that payments are routed via a common intermediate peer — such as a currency exchange — since the intermediary must now be trusted to keep private your full payment history. Some proposals, such as the Lightning Network, have proposed to work around this problem by routing the payment via *multiple* intermediary nodes; however (as we discuss in §6) this approach substantially increases the complexity of establishing payment channels, and reveals payment information in the event that even a subset of the intermediaries collude.

Several academic works have recently proposed solutions that address the privacy problems of Bitcoin-type currencies [29, 41, 42, 47]. Some of the resulting systems been publicly deployed, notably ZCash [3] (an implementation of the Zerocash protocol [47]) and Monero [2]. Unfortunately, the privacy mechanisms contained in these systems apply to the privacy of transactions *on the blockchain*, and do not address the setting of payment channels. Indeed privacy for payment channels seems fundamentally challenging due to channels' pairwise structure. Even when a channel is funded with anonymous currency, repeated payments within the same channel are inherently linkable. This is concerning, given that one of the main proposed applications of channels is for *web micropayments* — which are often described as a more private alternative to tracking and online behavioral advertising.

We stress that concerns about privacy are not theoretical. Several commercial ventures [11, 23, 32] have been founded around the task of analyzing and tracing blockchain transactions. It is reasonable to expect that surveillance will be applied to payment channel systems if they become widely deployed.

Our Contribution. In this paper we propose Blind Off-chain Lightweight Transactions, or *Bolt*. Bolt consists of a set of techniques for constructing *privacy-preserving* payment channels for a decentralized currency. These techniques ensure that multiple payments on

[1] As of early May 2017, this has resulted in a backlog of nearly 165,000 transactions [15].

Hybrid architecture model to increase bootstrapping capability on Bitcoin

Richard Dennis
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Richard.dennis@port.ac.uk

Gareth Owenson
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
Gareth.owenson@port.ac.uk

Abstract— **How to scale Bitcoin is still an open research question, while most of the research currently focuses on increasing the number of transaction Bitcoin can process, this paper takes a different view, and looks at the bootstrapping method. We demonstrate an effective and proven attack on the current DNS protocol which enables a low resourced attacker to partition new nodes joining the network. We then conduct analysis on how well the current DNS model can scale, before suggesting a hybrid P2P architecture model comparing this model with the current protocols, in terms of resistance to the DNS attack and scalability.**

Blockchain, scalability, cryptographic protocols, distrubted networks, peer-to-peer, Bittorent

I. INTRODUCTION

Bitcoin is the first worldwide, mass adopted cryptocurrency and digital payment system to be implemented and deployed without a requirement of a centralized repository system or administrator. It was invented by an unknown programmer, or a group of programmers, under the name Satoshi Nakamoto, published in a white paper in 2008, before being released as open-source software in 2009.

The key invention made by Nakamoto was the blockchain is a novel peer-to-peer approach which links a sequence of transactions or events together in a way that makes them immutable.

The blockchain is a public ledger of all transactions that have ever been completed since the first "genesis" block. Each transaction from the Bitcoin protocol is broadcast to all nodes in the network which are maintaining the blockchain.

A blockchain-node and a miner are two types of nodes on the network, which while can conducted on the same node, is usually separated. A blockchain-node can be classed as node which maintains and updates the blockchain, with valid blocks received from miners on the network.

Each blockchain-node confirms if each transaction is valid and can be added to a block. Each blockchain-node confirms every transaction made on the network, to do so, each blockchain-node will search through the blockchain they store and maintain to see if the user requesting the transaction has got enough funds to process the transaction, and this transaction has not previously been conducted. Only once this process has happened will each node compile a block (a group of transactions) and send this to the miners. There are not incentives to run a blockchain-node.

A miner participates in the process by which transactions are verified and added to the public ledger, known as the block chain, and also the means through which new bitcoin are released. The mining process involves compiling recent transactions into blocks and trying to solve a computationally difficult puzzle. The participant who first solves the puzzle gets to place the next block on the block chain and claim the rewards. The rewards, which incentivize mining, are both the transaction fees associated with the transactions compiled in the block as well as newly released bitcoin

Bitcoin is the most successful blockchain-based network; it has a market cap of over USD 8.5 billion and sees an average of 214,000 transactions being conducted on its network every day.

Blockchain-based networks have not properly addressed the issue of scalability; this causes the original decentralized nature of the blockchain to become increasingly centralized, as only the highest-resourced users are able participate in the network. This is because each node on the network is required to store the entire blockchain, which stores every transaction since its deployment and consequently low-resourced users; such as mobile users – are excluded from the network.

There has been several other peer-to-peer (P2P) and decentralized networks such as BitTorrent which have face similar scalability issues, overcame some of these issues with the use of a Hybrid architecture model, combining both the client-server model and P2P architecture.

This paper proposes a new approach in the way lower resourced nodes can be included in the network. It will examine how trusted "super nodes" can be utilized enabling a global view of the network, while providing monitoring facilities of all nodes on the network. We will conduct a thorough analysis on how the introduction of super nodes can aid in the scalability of the bootstrap process, by first demonstrating a unique attack against the currently implementation of the bootstrap protocol, and then analysing how the introduction of super nodes not only prevents such an attack from occurring, but also allows for a greater number of simultaneous nodes to bootstrap at the same time.

How to Charge Lightning*

The Economics of Bitcoin Transaction Channels

Simina Brânzei† Erel Segal-Halevi‡ Aviv Zohar§

September 25, 2017

Abstract

Off-chain transaction channels represent one of the leading techniques to scale the transaction throughput in cryptocurrencies. However, the economic effect of transaction channels on the system has not been fully explored up until now. We present a framework for economic analysis of the lightning network and its effect on transaction fees on the blockchain. Our framework allows us to reason about different patterns of demand for transactions, different topologies of the lightning network and to derive the resulting fees for transacting both on and off the blockchain. Our initial results (that should be considered carefully) indicate that while the lightning network does allow for a substantially greater number of transactions to pass through the system, it does not necessarily provide higher fees to miners, and as a result may in fact lead to lower participation in mining within the system.

1 Introduction

A main approach to solve the scalability problem in Bitcoin is to use off-chain transaction channels that allow parties to communicate and transfer funds while communicating directly, and only occasionally to settle on the blockchain. The recent deployment of SegWit, a solution to transaction maleability (among other benefits) opens the path for better constructions of off-chain transaction channels. While transaction channels themselves are limited to exchanges between pairs of individuals, further developments like the lightning network allow to route payments over longer paths and thus can allow the construction of a well connected network of payment channels that can be used to transfer money quickly and with relatively little interaction with the blockchain.

One of the key unknowns regarding fast payment networks is the economic effect that they will have on the Bitcoin fee market. If the blockchain is used less often, fees to miners are paid less frequently and competition for space in blocks declines. Bitcoin's security depends heavily on having a large amount of computational power invested in solving proof-of-work puzzles, making it hard for attackers to double spend or censor transactions in the currency. As the

*The authors are in alphabetical order. This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 740282), from the Israel Science Foundation (grant 616/13 and grant 1083/13) and from the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau (grant 039-9230). Simina was also supported by the ISF grant 1435/14 administered by the Israeli Academy of Sciences and Israel-USA Bi-national Science Foundation (BSF) grant 2014389 and the I-CORE Program of the Planning and Budgeting Committee and The Israel Science Foundation. Erel was supported by the ISF grant 1083/13.

†The Hebrew University of Jerusalem, Israel. E-mail: simina.branzei@gmail.com

‡Ariel University, Israel. E-mail: erelsgl@gmail.com

§The Hebrew University of Jerusalem, Israel. E-mail: avivz@cs.huji.ac.il

Redesigning Bitcoin's fee market

Ron Lavi[1], Or Sattath[2,3], and Aviv Zohar[2]

[1]Technion
[2]The Hebrew University
[3]MIT

September 25, 2017

Abstract

The security of the Bitcoin system is based on having a large amount of computational power in the hands of honest miners. Such miners are incentivized to join the system and validate transactions by the payments issued by the protocol to anyone who creates blocks. As new bitcoins creation rate decreases (halving approximately every 4 years), the revenue derived from transaction fees start to have an increasingly important role. We argue that Bitcoin's current fee market does not extract revenue well when blocks are not congested. This effect has implications for the scalability debate: revenue from transaction fees may decrease if block size is increased.

The current mechanism is a "pay your bid" auction in which included transactions pay the amount they suggested. We propose two alternative auction mechanisms: The Monopolistic Price Mechanism, and the Random Sampling Optimal Price (RSOP) Mechanism (due to Goldberg*et al.*). In the monopolistic price mechanism, the miner chooses the number of accepted transactions in the block, and all transactions pay exactly the smallest bid included in the block. The mechanism thus sets the block size dynamically (up to a bound required for fast block propagation and other security concerns). We show, using analysis and simulations, that this mechanism extracts revenue better from users, and that it is nearly incentive compatible: the profit due to strategic bidding relative to honest biding decreases as the number of bidders grows. Users can then simply set their bids truthfully to exactly the amount they are willing to pay to transact, and do not need to utilize fee estimate mechanisms, do not resort to bid shading and do not need to adjust transaction fees (via replace-by-fee mechanisms) if the mempool grows.

We discuss these and other properties of our mechanisms, and explore various desired properties of fee market mechanisms for crypto-currencies.

Using the Chain for what Chains are Good For

This proposal is for a high-level introductory talk. The list below can be compressed into a paragraph if that fits the abstract format better.

The blockchain is simultaneously Bitcoin's core innovation, letting it succeed where no other system had before, and its greatest weakness, requiring miner discretion in choosing transactions, while no other part of the system has any third-party dependence. In exchange for this dependence, miners produce an increasingly-immutable proof-of-publication medium, allowing anyone at any time to see what transactions occurred and in what order, and to be assured that their view matches the view of all other validators.

Bitcoin's essential use of the blockchain is to prevent *double-spending*: to publish transaction outputs as they are created and later consumed as inputs. This provides an unambiguous beginning and end of each output, which an otherwise unattainable goal in a relativistic world. However Bitcoin uses the blockchain for much more than this: it has a script system which allows users to set arbitrary spend conditions on their coins; it allows transactions to be time-locked and invalid until some time has passed; it ensures transactions are executed atomically and not peeled apart on the network. All of these conditions are published on the chain and verified by all validators. It is the thesis of this talk that these "non-essential" uses of the blockchain can often be done with significantly reduced (or eliminated) use of the blockchain, and that this has tremendous benefits for the transactors themselves as well as the network as a whole.

First, we describe the costs of blockchain usage.

- Blocks appear only every ten minutes on average, meaning long and unpredictable latency for users of the chain.
- During this time transactions are published to the network, leaking private timing and source information, plus the transaction data itself, to anyone who cares to analyze it. This exposure undermines users' privacy, businesses' confidentiality, and the fungibility of the currency itself.
- This public data can be seen by miners before they include transactions, which poses a censorship risk for users as well as an incentive for adversaries to pressure miners into censorship.
- Blockchain space is limited, forcing users to pay for this data even as its existence harms them.
- This data must be validated by all participants in the system who want to verify that their view of its state is untampered with. Since supporting these users is a core component of the Bitcoin ethos, the result is a limitation on the scalability of the entire system.
- Finally, the rules for validating data on the blockchain are system-wide rules that cannot be changed without agreement from all users. This is not even possible for conflicting rules, but when it is possible it is extremely hard to measure and hard to achieve.

BITCOIN MINING AS A CONTEST

1. Introduction

Since its introduction Bitcoin received much attention as a peer-to-peer cryptocurrency based on the blockchain technology (Bonneau et al 2015, Narayanan et al 2016). Adoption of Bitcoin may exhibit advantanges as well as critical aspects (Böhme et al, 2015; Athey et al, 2016). From an economic perspective its use may facilitate exchange and possibly save on transaction costs. Because of its exchangeability with fiat currencies such as the dollar, advantages could also come from a speculative activity based on oscillations of the exchange rate.

However, one of its most distinguishing features is that registration of transactions is done through the so called *mining* activity undertaken by some subjects. Such activity consists in solving a puzzle requiring high computational power, since registration of a block of transactions can only take place once the puzzle has been solved. Providing the right economic incentives to solve the puzzle is very important for the transactions to be registered on the underlying ledger. This is why for this activity *miners* are compensated with two types of rewards: first, for any solved puzzle the miner will receive a fixed sum of bitcoins by the protocol and, moreover, individuals behind a transaction may offer a fee to the miner for its registration. The larger such fee the higher the incentive for the miner to enclose a transaction in the next registered block. The fixed sum received by the protocol for each block of registrations will tend to decline over the years until its disappearance, after which only fees paid for transactions registration will reward the miners.

In this paper we focus on the mining activity as a source of economic profitability (Narayan et al, 2015), where the main strategic decision taken by miners is how much to invest in computational power to solve the puzzle. Since mining costs are increasing, the chosen level of power is becoming a critical issue for the Bitcoin community, which is what motivates our analysis. Within a very simple static game theoretic framework, our model provides some interesting insights. Due to the assumption of exponential waiting time for the solution to a puzzle, the mining activity can be characterised as an all-pay *contest* (Konrad, 2009, Vojnovic, 2015) a conceptual framework widely adopted in social sciences. All-pay contests are competitions where winners are awarded a prize specified in advance by the organizer. They require investments to participate, and this is what makes them all-pay, and victory by a participant typically occurs probabilistically. Therefore, those who obtain no prize lose their investments, unless these could be re-used in other contests. Winning probabilities are often called contest functions.

Indeed, mining activity can be seen as a contest where participants are trying to come first in the competition for the solution of the puzzle, receiving as prize a given amount of Bitcoins as well as some fees from the other participants.

At the Nash Equilibrium of the mining game with perfect information, while the level of computational power chosen by an active miner depends also on how many bitcoins could be obtained solving the puzzle, the decision to become an active miner depends only on his own marginal costs as compared to the opponents' cost structure. That is, the decision to be an active miner would only depend upon how efficient are the competitors but not on how many bitcoins will be obtained as reward.

Topic (tentative): Bitcoin script 2.0 and strengthened payment channels

Johnson Lau, Olaoluwa Osuntokun

Bitcoin uses a scripting system to imbue users with the power to designate the conditions under which UTXO's they create can be spent. However, the capabilities of Script to date are limited in many ways. Amongst these limitations include:

- Several useful opcodes were disabled in a series of emergency softforks in 2010
- Users are unable to commit to additional scripts or conditions after a UTXO is created. Scripts in the scriptSig (prior to segwit) were malleable by third parties and cannot be utilized in any meaningful way
- Script operations have very restricted access to the rest of the transaction. Only the 6 SIGHASH types allow limited introspection into the execution environment (inputs, scripts, transaction spending, etc)
- Numerical operations accept up to 32-bit signed integers, while 51-bit is needed to cover the full range of bitcoin supply (2.1E15)

The activation of segwit has paved the way for the evolution of contracts on top of Bitcoin due to it's long overdue malleability fix and added Script versioning capabilities. The malleability fix allows contract designers to safely nest pre-signed contract execution pathways typically heavily utilized in the Bitcoin model of contract design. Additionally, the Script versioning features allows for rapid evolution of Script, as it's possible to entirely re-design portions of Script with a single version bump. Within the development community several new features have been proposed for the next generation of Bitcoin Script including:

- OP_PUSHTXDATA (Covenant)
- Signature-time commitment to additional scripts
- OP_CHECKSIGFROMSTACK
- Re-enabling all the disabled operations
- More SIGHASH types
- Addition of op-codes for primitive EC operations (group op, scalar mult)

The set of proposed additions to Script outlined above have the potential to significantly broaden the expressivity, and power of Bitcoin's Script in the domain of smart contract design. In this talk, we'll begin by briefly overview the past history of the evolution of Script. With the necessary historical context explored, will then detail the new anticipated proposed additions to Script which include the augmented power of Script introspection. Finally as a case study, we'll utilize the new Bitcoin contract functionality to design a new version of payment channels for Lightning which improve upon the scalability, safety, and privacy of prior iteration of channel design.

References:

BIP draft: `Merklized Script`
https://github.com/jl2012/bips/blob/vault/bip-0114.mediawiki
BIP draft: `Pay-to-witness-public-key`
https://github.com/jl2012/bips/blob/vault/bip-0VVV.mediawiki

David Vorick Follow
Draft · 14 min read

Microchains—massive blockchain scalability, among other significant advantages

Proposal Note: I would boil this down to 30 mintues by selecting a few of the biggest ideas, most importantly the scorched-earth defense idea, and then presenting on those.

I am about to propose a system which I claim solves many of blockchain's problems at the same time. We are able to achieve this by turning some of our core assumptions about blockchain upside down, and by leveraging new advances in game theory and crypto-economics, without which we would not be able to achieve any remote semblance of security.

There are four major problems in blockchain today which I believe a microchains ecosystem can solve:

- Scalability. This is the biggest one, and the real reason that everyone should take a close look at microchains. I claim that we can get somewhere between 1,000x and 10,000,000x scalability over traditional blockchains, primarily because we end up with a secure system where people no longer need to validate every transaction to achieve full security and full trustlessness—and can do so without finality, proof of stake, or any of the other flaky mechanisms used by other scalable system
- Upgradeability. Today, upgrading a blockchain is a headache. Either you have to do a fancy soft-fork which gains widespread adoption, or you have to perform a hardfork, which either needs absolute adoption or creates a network split. In the microchains system, you can easily perform incremental hardfork upgrades, where only a small fraction of the network needs to upgrade at a time.
- Blockchain Factions. Bitcoin today requires everyone to run the exact same consensus code. Multiple implementations are highly frowned upon due to the risk of tiny hardfork bugs that cause

The Future of Proof of Work
Min Chen, CTO, Canaan Creative China (Avalon)
30 Minute Presentation

In this presentation, Min Chen, CTO for Canaan Creative China - the inventors of the AvalonMiner, the first Bitcoin mining ASIC microprocessor - describes where Proof of Work is heading, along with problems approaching this solution, and how to best solve them. Particular emphasis will be placed on the evolution of ASIC microprocessor development from the perspective of the Canaan Avalon series of chips towards where Bitcoin mining and development is going, and how to do this in silicon.

In her first presentation on the subject publicly, Min Chen will cover some unwritten history of the evolution she led from FPGA based mining Icarus and Lancelot, to the first Avalon Bitcoin mining ASIC microprocessors up to the current scaling issues facing Bitcoin, power requirements, and paths for future currency development using hardware.

PRESENTATION PROPOSAL FOR SCALING BITCOIN STANFORD

Measuring maximum sustained transaction throughput on a global network of Bitcoin nodes

Andrea Suisani,[1] Andrew Clifford,[1] Andrew Stone,[1] Erik Beijnoff,[1] Peter Rizun,[1] Peter Tschipper,[1] Alexandra Fedorova,[2] Chen Feng,[2] Victoria Lemieux,[2] Stefan Matthews[3]

[1] Bitcoin Unlimited, [2] University of British Columbia, [3] nChain

Overview

Although it is well understood that increasing Bitcoin's block size limit (currently 1 MB) would immediately reduce transaction fees and improve confirmation reliability, concern exists regarding the network's ability to safely and reliably handle the associated increase in transaction throughput.

To investigate this concern, we set up a global network of Bitcoin mining nodes[1] configured to accept blocks up to one thousand times larger (1 GB) than the current limit. To those nodes we connected transaction generators, each capable of generating and broadcasting 200 transactions per second (tx/sec) sustained.[2] We performed (and are continuing to perform) a series of "ramps," where the transaction generators were programmed to increase their generation rate following an exponential curve starting at 1 tx/sec and concluding at 1000 tx/sec—as illustrated in Fig. 1—to identify bottlenecks and measure performance statistics.

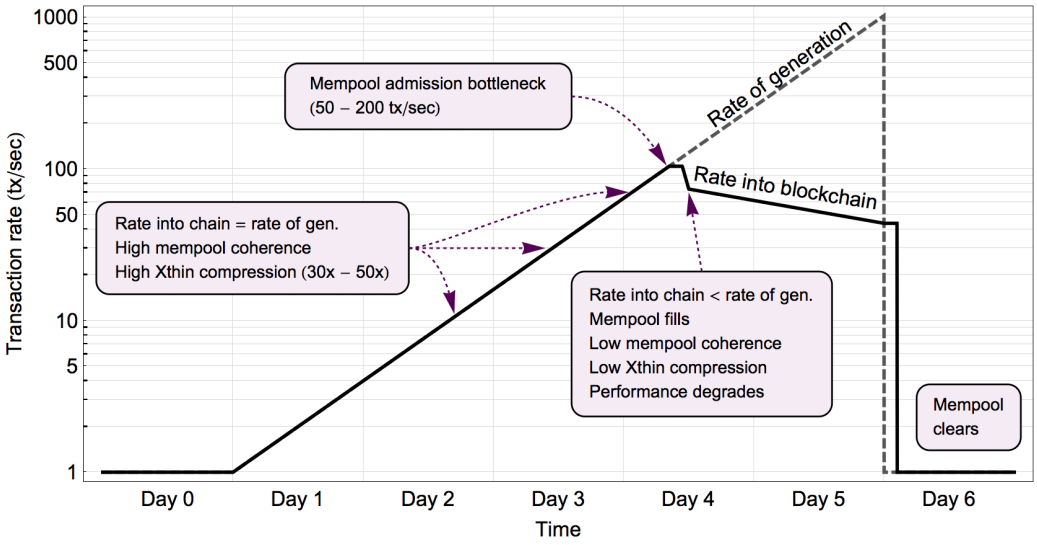


Fig. 1. Ramp input and typical node response.

[1] At the time of writing, there were mining nodes in Toronto (64 GB, 20 core VPS), Frankfurt (16 GB, 8 core VPS), Munich (64 GB, 10-core rack-mounted server with 1 TB SSD), Stockholm (64 GB, 4 core desktop with 500 GB SSD), and central Washington State (16 GB, 4 core desktop). With the passing of BUIP065 and the associated $300,000 per year funding for the Gigablock Testnet Initiative, additional mining nodes will be deployed in Beijing, Bangalore, Sao Paulo, Sydney and Vancouver. The results we present at Stanford will include data from this larger test network as well.

[2] At the time of writing, there were generators in San Francisco, New York, London, Amsterdam, Singapore and Bangalore (all 8 GB, 4 core VPS). Generators are Python applications interacting with a local instance of `bitcoind`.

Graphene: A New Protocol for Block Propagation Using Set Reconciliation

A. Pinar Ozisik† Gavin Andresen George Bissias† Amir Houmansadr† Brian N. Levine†
†College of Information and Computer Sciences, UMass Amherst

1 INTRODUCTION

We propose an efficient method of announcing new blocks called *Graphene*. This document summarizes a more detailed description of Graphene that has been published previously [7].

Graphene blocks are a fraction of the size of related methods, such as Compact Blocks [3] and Xtreme Thinblocks [9]. For example, while a 17.5 KB Xtreme Thinblock can be encoded in 10 KB with Compact Blocks, the same information can be encoded in 2.6 KB with Graphene. We use a novel interactive combination of Bloom filters [2] and Invertible bloom lookup tables (IBLTs) [5], providing an efficient solution to the problem of set reconciliation in Bitcoin's p2p network.

Block announcements are validated using the transaction content comprising the block. However, it is likely that the majority of peers have already received these transactions, and they only need to discern them from those in their mempool. In principle, a block announcement needs to include only the IDs of those transactions. For example, Corallo's *Compact Block* design [3] significantly reduces block size by including a transaction ID list, though the cost is increased coordination between peers to 5 messages. *Xtreme Thinblocks* [9] works similarly to Compact Blocks but has greater data overhead. Specifically, if an `inv` is sent for a block that is not in the receiver's mempool, the receiver sends a Bloom filter of her IDpool along with the request for the missing block. As a result, Xtreme Thinblocks are larger than Compact Blocks but require just 3 messages. The community has discussed in forums the use of IBLTs (without Bloom Filters) for reducing block announcements [1, 8], but these schemes have not been formally evaluated and are less efficient than our approach. Our method is novel; we have proved and demonstrated that it is smaller than all of these recent works, and still requires 3 messages between sender and receiver for coordination.

2 THE GRAPHENE PROTOCOL

Unlike other approaches, Graphene never sends an explicit list of transaction IDs. Instead it sends a small Bloom filter and a very small IBLT. The intuition behind Graphene is as follows; a summary appears in Figure 1.

PROTOCOL 1: Graphene

1: Sender: Sends *inv* for a block.
2: Receiver: Requests unknown block; includes count of txns in her IDpool, m.
3: Sender: Sends Bloom filter $\mathcal{S}$ and IBLT $\mathcal{I}$ (each created from the set of n txn IDs in the block) and essential Bitcoin header fields. The FPR of the filter is $f = \frac{a}{m-n}$, where $a = n/(c\tau)$.
4: Receiver: Creates IBLT $\mathcal{I}'$ from the txn IDs that pass through $\mathcal{S}$. She decodes the *subtraction* [4] of the two blocks, $\mathcal{I} \triangle \mathcal{I}'$.

Figure 1: A summary of the Graphene protocol.

The sender creates an IBLT $\mathcal{I}$ from the set of transaction (txn) IDs in the block. To help the receiver create the same IBLT (or similar), he also creates a Bloom filter $\mathcal{S}$ of the transaction IDs in the block. The receiver uses $\mathcal{S}$ to filter out transaction IDs from her pool of received transaction IDs (which we call the IDpool) and creates her own IBLT $\mathcal{I}'$. She then attempts to use $\mathcal{I}'$ to *decode* $\mathcal{I}$, which, if successful, will yield the transaction IDs comprising the block. The number of transactions that falsely appear to be in $\mathcal{S}$, and therefore are wrongly added to $\mathcal{I}'$, is determined by a parameter controlled by the sender. Using this parameter, he can create $\mathcal{I}$ such that it will decode with very high probability.

In sum, the Bloom filter from the sender allows the receiver to determine which transactions from its mempool are in the block. Other approaches require a much larger Bloom filter to keep the false positive rate small; in Graphene, the Bloom Filter FPR is high because the IBLT recovers any mistakes made. Similarly, if only the IBLT was used, it would be much larger than our use of the two mechanisms.

A Bloom filter is an array of x bits representing y items. Initially, the x bits are cleared. Whenever an item is added to the filter, k bits, selected using k hash functions, in the bit-array are set. The number of bits required by the filter is $x = y\frac{-\ln(f)}{\ln^2(2)}$, where f is the intended false positive rate (FPR). For Graphene, we set $f = \frac{a}{m-n}$, where a is the expected difference between $\mathcal{I}$ and $\mathcal{I}'$. Since the Bloom filter

Bobtail: A Proof-of-Work Target that Reduces Blockchain Mining Variance

George Bissias Brian N. Levine
College of Information and Computer Sciences
University of Massachusetts Amherst

Introduction. Blockchain systems are designed to produce blocks at a constant average rate. Bitcoin [1] produces, on average, one block every 10 minutes. Unfortunately, the time between blocks has high variance and the distribution of inter-block times has a very long tail. For example, 5% of the time, Bitcoin's inter-block time is at least 40 minutes. This variance impedes the consistent flow of validated transactions through the system.

The high inter-block time variance is a direct consequence of Bitcoin's Proof-of-Work (PoW) algorithm. Generally, the miners repeatedly craft block headers by changing a nonce until the hash of the header is less than a target value t. In other words, the hash of each header is a sample taken randomly from a discrete uniform distribution. A block is discovered when the *first order statistic* (i.e., the minimum value) of all sampled values is less than target t.

Proposal. We propose an alternative process for PoW-based block discovery that results in an inter-block time with significantly lower variance. Our algorithm generalizes the current algorithm by comparing the mean of the k lowest order statistics to a target. As a result, the variance of inter-block times decreases as k increases. For example, if our approach were applied to Bitcoin, about 80% of blocks would be found within 7 to 12 minutes, and nearly every block would be found within 5 to 18 minutes; the average inter-block time would remain at 10 minutes. The cost of our approach is a larger block header. We call our approach *Bobtail*[1] mining. Figure 1 shows results from experiments with this method.

This proposal summarizes a full paper available from https://forensics.umass.edu/bobtail.

Bobtail Mining. Consider a fixed interval of time during which the entire network produces θ hashes generating a sequence of hash values $\mathbf{Z} = Z_1, \ldots, Z_\theta$. Let Z be an arbitrary random variable from the sequence $\mathbf{Z}$; note that $Z \sim \texttt{Uniform}(0, S)$. Define V_i to be the ith lowest order statistic of $\mathbf{Z}$, i.e. $V_i = Z_{(i)}$ in standard notation. And let random variable W_k be the mean of the k lowest order statistics:

$$W_k = \frac{1}{k} \sum_{i=1}^{k} V_i. \tag{1}$$

W_k constitutes the collective mining proof (*proof*, for short) for the entire network. Our Bobtail mining criterion says that a new block is discovered when a realized value of W_k meets the target t:

$$w_k \leq t. \tag{2}$$

Notably, this approach is a generalization of current systems, which are the special case of $k = 1$.

[1] A *bobtail* refers to an animal's tail that is unusually short or is missing completely (https://en.wikipedia.org/wiki/Natural_bobtail).

BlockSci: a Platform for Blockchain Science and Exploration

Harry Kalodner, Malte Möser, Steven Goldfeder, Alishah Chator, and Arvind Narayanan

Abstract

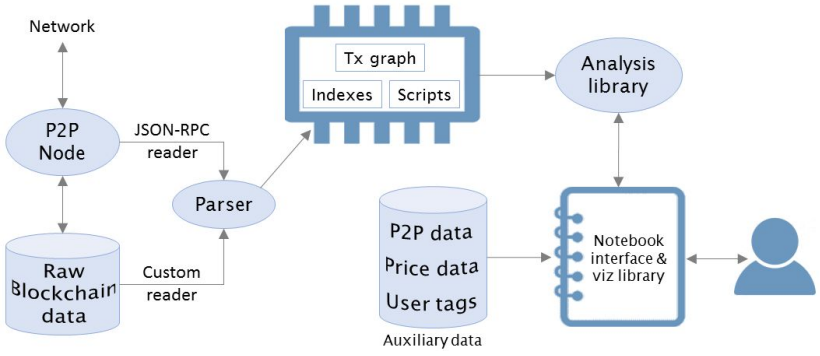
In order to understand the scaling demands of Bitcoin, we need to understand the usage demands on Bitcoin. Using BlockSci, a high-performance tool for blockchain science and exploration, we can answer questions about the nature of transactions that are being included in the blockchain. This knowledge, in turn, can help build better scaling solutions for Bitcoin.

Talk Proposal

The Bitcoin blockchain — currently 140GB and growing — contains a massive amount of data that can give us insights into the Bitcoin ecosystem, including how users, businesses, and miners operate. In this talk we present BlockSci, an open-source software tool that enables fast and expressive analyses of Bitcoin's and many other blockchains. BlockSci has already been used in multiple academic papers, and we have released it as open-source software to encourage open and reproducible blockchain science.

An open source cryptocurrency analytics framework can provide valuable data to evaluate changes to Bitcoin. Often ideas and heuristics are discussed in the abstract, without concrete numbers attached. If numbers are cited, the details of the analysis are often private and unverifiable. Ideally, decisions impacting Bitcoin users should be made based on reliable data that was interpreted in a reproducible way.

BlockSci enables the science of blockchains. It addresses three pain points of existing tools: poor performance, limited capabilities, and a cumbersome programming interface. BlockSci is 15x–600x faster than existing tools, comes bundled with analytic modules such as address clustering, exposes different blockchains through a common interface, imports exchange rate data and "mempool" data, and gives the programmer a choice of interfaces: a Jupyter notebook for intuitive exploration and C++ for performance-critical tasks. We show an overview of BlockSci's architecture below.



Overview of BlockSci's architecture

BlockSci's design starts with the observation that blockchains are append-only databases; further, the snapshots used for research are static. This makes an in-memory analytical database the natural choice. On top of the obvious speed gains of memory, we apply a number of tricks such as converting hash

1. Changes without unanimous consent

"By expecting a few developers to make controversial decisions you are breaking the expectations, as well as making life dangerous for those developers. I'll jump ship before being forced to merge an even remotely controversial hardfork."
-- https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-June/009137.html

If proposed consensus changes have unanimous consent, then life is easy: developers are happy to accept the patches, miners are happy to deploy the updated software and can quickly indicate that they are enforcing the changes making the risk of chain splits arbitrarily low, validating nodes are happy to enforce the new rule set, the bitcoin economy is happy to continue respecting the resulting ledger, and regulators are happy to refrain from making anyone unhappy.

However there are many reasons why any of those people may not be happy for any given proposed change:

- Different people in the bitcoin ecosystem may have different goals; as a trivial example, regulators might propose a change due to a strong desire to prevent bitcoin from being usable for criminal purposes, while others may oppose that change due to its impact on anonymity, fungibility, or efficiency. as bitcoin grows, it is likely to attract more people with conflicting goals, increasing the frequency of this form of conflict arising.

- There may be a lack of understanding of the impact of a change; perhaps if everyone understood a change's actual effects, everyone would support (or oppose) it, but it may be that many people are unable to reach this conclusion with confidence, whether due to ignorance, or lack of time to fully study the issue are unable. If bitcoin adoption increases amongst people with less technical background or with different attitudes to monetary policy, this class of controversy is more likely to occur.

- Bitcoin upgrades are generally designed as Pareto-improvements, that is either everyone in the ecosystem is made better off, or at least are not made any worse off. It may be that some changes will not be able to be made in this manner, and some changes will cause a net benefit to the ecosystem, despite a net loss to some individuals.

Incentives and Trade-offs in Transaction Selection in DAG-based Protocols

Yoad Lewenberg and Yonatan Sompolinsky

Introduction

In Bitcoin, and in the many cryptocurrencies that followed its path, the public ledger consists of a single chain of blocks. Every block embeds a pointer to a single parent block, and only blocks on a single chain -- the longest, usually -- are considered; blocks off this chain (aka orphans) are omitted from the ledger and discarded. In contrast, Lewenberg *et al.* [1] proposed a new setup where blocks may reference several parent blocks as their predecessors. The resulting data structure is a Directed Acyclic Graph of blocks (a block DAG). DAG-based protocols are increasingly gaining traction (see, e.g., the work on the SPECTRE protocol [2], which utilizes all blocks avoiding the notion of chain altogether; or Ethereum, which uses a DAG in order to reward off chain blocks and mitigate centralization thus), which justifies reexamining the ways in which these protocols can be deployed in reality and deliver on their promise.

One primary advantage of a DAG ledger is that it improves the scalability of the system in terms of transaction throughput, by allowing contribution of content from more blocks (not only those contained in a single chain). The high level idea is to incentivize miners to avoid embedding the same transaction in multiple blocks in the DAG, since that would waste potential throughput and at the same time harm the miners themselves; indeed, the fee from a transaction may only be distributed once, either to one block or divided somehow.

Concretely, assume that two blocks were created by honest miners at about the same time, and therefore they do not reference one another (directly or even indirectly). It is possible that some conflicting transactions appear in these blocks, in which case one of the conflicting transactions must be omitted, according to some rule. Either way, the remaining sets of transactions, which are compatible, can be co-accepted and considered part of the UTxO. Of course, if these blocks merely duplicate the same transaction-set then nothing will be gained by accepting both and we might as well discard one of them. However, if these blocks contain mutually unique transactions, integrating both blocks into the ledger and accepting their non-conflicting content improves the overall throughput.

The Inclusive paper [1] analyzes the dynamics that unfold from a game theoretic point of view. Overall, we observe that miners will indeed be incentivized to avoid collisions and include unique contents in their blocks. They can do so by randomizing over their local mempools and selecting transactions for their next blocks according to some probability distribution (the equilibrium distribution). The distribution takes into account the fee that each transaction entails, as well as the probability that the miner will indeed earn it in case of a collision.

A New Blockchain-as-a-Service Paradigm

Yonatan Sompolinsky and Yoad Lewenberg

I. Introduction

Since the inception of Bitcoin, many blockchain based systems have been proposed and deployed. Typically, the underlying protocol specifies which transactions should be considered valid, and how to create and add new blocks of transactions to the ledger. Importantly, most if not all of current blockchain systems follow Satoshi's design in that the validity of blocks and that of transactions is coupled: a valid block must not contain invalid transactions.

We propose revisiting this design choice. Our objective is to increase the scalability of the system as well as to introduce a more soft and decentralized governance model for innovation on the application-layer. These two objectives are achieved by treating separately the different layers of blockchain, and in particular decoupling the validity of blocks from that of transactions. The result is a system where miners can contribute their mining resources to secure several application-layer protocols simultaneously. Our solution is closely connected to the concept of merged mining, yet is different in some core aspects.

II. Blockchain layers

In our design, blockchain consists of two primary layers, the mining layer and the application layer. The mining layer is where the formation of the ledger takes place: users publish data (or: transactions) with an associated fee, and miners collect these data and embed them in blocks. The rules of block creation are dictated and enforced by the mining protocol. However, this mining protocol is agnostic to the data itself---the validity of a block is determined solely by its header and structure, and does not depend on the transaction data embedded in it.
The result of the mining layer is a public ledger of transactions, or simply *the ledger*.

Next, the application layer is where the ledger is interpreted by users, or *clients*. An interpretation protocol *P* specifies how to read the ledger, and specifically how to treat those transactions that were labelled as belonging to *P*; every transaction in the ledger specifies in its own header the name/ID of the application-layer protocol according to which it ought be interpreted. For instance, transactions can be labeled by their creators as <bitcoin txn>, <zcash txn>, <ethereum smart contract>, <ethereum classic smart contract>, etc. The user can then decide which protocol(s) to run on his client, according to his own economic value and preference. For instance, the client can ignore all but transactions labeled as <bitcoin txn>, and interpret the rest of the ledger data as blank data. In this way, we allow multiple application-layer protocols to be co-hosted on the same ledger and mining platform.

We stress the following crucial design-rule: A block that was mined correctly (according to the mining protocol's instructions) remains valid regardless of its data. In particular, it may contain invalid <bitcoin txns> transactions, and while users running the Bitcoin client will ignore these

DOUBLE SPEND RACES

CYRIL GRUNSPAN AND RICARDO PÉREZ-MARCO

Abstract. We correct the double spend race analysis given in Nakamoto's foundational Bitcoin article and give a closed-form formula for the probability of success of a double spend attack using the Regularized Incomplete Beta Function. We give a proof of the exponential decay on the number of confirmations, often cited in the literature, and find an asymptotic formula. Larger number of confirmations are necessary compared to those given by Nakamoto. We also compute the probability conditional to the known validation time of the blocks. This provides a finer risk analysis than the classical one.

To the memory of our beloved teacher André Warusfel who taught us how to have fun with the applications of mathematics.

1. Introduction.

The main breakthrough in [7] is the solution to the *double spend problem*. Before this discovery no one knew how to avoid the double spending of an electronic currency unit without the supervision of a central authority. This made Bitcoin the first form of *peer-to-peer* (P2P) electronic currency.

A double spend attack can only be attempted with a substantial fraction of the hashrate used in the *Proof-of-Work* of the Bitcoin network. The attackers will start a *double spend race* against the rest of the network to replace the last blocks of the blockchain by secretly mining an alternate blockchain. The last section of [7] computes the probability that the attackers catch up. However Nakamoto's analysis is not accurate since he makes the simplifying assumption that honest miners validate blocks at the expected rate. We present a correct analysis and give a closed-form formula for the exact probability.

Date: February 9th 2017.

2010 *Mathematics Subject Classification.* 68M01, 60G40, 91A60, 33B20.

Key words and phrases. Bitcoin, blockchain, double spend, mining, proof-of-work, Regularized Incomplete Beta Function.

Acknowledgements: We are grateful to N. Emerson for his comments and reading over the article.

Discreet Log Contracts

Thaddeus Dryja

MIT Digital Currency Initiative

Abstract

Smart contracts [1] are an often touted feature of cryptographic currency systems such as Bitcoin, but they have yet to see widespread financial use. Two of the biggest hurdles to their implementation and adoption have been scalability of the smart contracts, and the difficulty in getting data external to the curency system into the smart contract. Privacy of the contract has been another issue to date. Discreet Log Contracts are a system which addresses the scalability and privacy concerns and seeks to minimize the trust required in the oracle which provides external data. The contracts are discreet in that external observers cannot detect the presence of the contract in the transaction log. They also hinge on knowledge of a *discrete* logarithm, which is a plus.

Model

There are 3 parties involved in the contract process: Alice, Bob, and Olivia. Alice and Bob are contract counterparties, while Olivia is the oracle. Alice and Bob do not trust each other and do not need to know any legal identifying information about each other, but they must be able to communicate over an authenticated channel, and they must be able to persistently recognize each other. Alice and Bob also must be able to receive signed broadcast messages from Olivia. Olivia does not need to be aware of Alice and Bob, and ideally she has no contact other than broacasting information. The information is compact enough that broadcast could take place over the Bitcoin network itself, though this should not be necessary.

The DLC protocol can be used for a wide variety of contracts, covering most cases where payouts between parties depend on a publicly known number in the future. In this example, Alice and Bob make and execute

Higher-level Scaling in Layer 2: Blockstack Subdomains

Aaron Blankstein
aaron.blankstein.com
Blockstack PBC

September 22, 2017

Blockstack implements a naming system on top of Bitcoin. This is achieved through translating name operations, e.g., registrations, transfers, or name data updates, into Bitcoin transactions, using the `OP_RETURN` field of Bitcoin transactions to hold the operation's data. Rather than storing data associated with a name (domain name in the Blockstack Naming System) directly in transactions, data (zonefiles) are stored in a gossip-replicated network called Atlas. The security of the mapping from domain to zonefile is ensured by storing the hash of that data on the Bitcoin chain.

While this separation of data storage from the blockchain itself enables a large amount of scalability, each domain name registration requires *three* bitcoin transactions: a `PREORDER`, which reserves a hashed name, a `REGISTER`, which reveals that name, and an `UPDATE` which sets the associated zonefile hash for a name. This three-step process poses both usability and scalability concerns. With respect to usability, this means that a new user of Blockstack would not only need to pay three transaction fees, which can be prohibitive, but would also need to wait for all of these transactions to confirm, and other indexers in the naming network to read and integrate that information. Blockstack's indexers are, by default, configured to only process transactions which have at least 6 confirmations, which means a new user would have to wait up to 9 blocks before possessing a usable name. For scalability, this process limits the rate of new user registrations. This system can handle roughly 4000 new users per hour, and would need to account for nearly all Bitcoin transactions in that hour.

To address these concerns, Blockstack developed a system of subdomains which allows for a weaker form of name ownership, but one that dramatically reduces the costs of a registration, both to the network and the individual. A subdomain of the form `foo.bar.id` is a name that is resolved by a network node by first resolving `bar.id` and then finding a valid subdomain entry for `foo`. Subdomains may be owned by bitcoin addresses unrelated to the domain operator's bitcoin address. Because of this, the operations that flow out of name ownership can be performed by the *subdomain owner*. For example, the owner

Atomically Trading with Roger: Gambling on the success of a hardfork

Patrick McCorry[1], Ethan Heilman[2] and Andrew Miller[34]

[1] University College London p.mccorry@ucl.ac.uk
[2] Boston University heilman@bu.edu
[3] University of Illinois at Urbana-Champaign soc1024@illinois.edu
[4] Initiative for Cryptocurrencies and Contracts, initc3.org

Abstract. We present atomic trade protocols for Bitcoin and Ethereum that can bind two parties to swap coins in the event that two blockchains emerge from a single "pre-fork" blockchain. This work is motivated by a bet between two members of the Bitcoin community, Loaded and Roger Ver, to trade 60,000 bitcoins in the event that Bitcoin Unlimited's planned hardfork occurs and the blockchain splits into two distinct forks. Additionally we study several ways to provide replay protection in the event of hardfork alongside a novel mechanism called migration inputs. We provide a detailed survey and history of previous softforks and hardforks in Ethereum and Bitcoin.

1 Introduction

Bitcoin [30] is the world's first successful and most valuable cryptocurrency. In June 2017, it reached a market cap of $43bn USD [11] and processed ≈250,000 transactions per day [5]. However, Bitcoin's future is uncertain; it is reaching its capacity limits, and so far the community has failed to reach consensus on how best to increase its capacity.

One proposed approach for increasing capacity, called Bitcoin Unlimited (BU), involves removing the 1-megabyte-per-block parameter that most directly effects the capacity limit [35]. A competing approach, the Core Roadmap [27], calls for a technical upgrade called SegWit [25], followed by deployment of the overlay payment network, Lightning [31]. Both approaches require changing the network's consensus rules; however there is a critical difference between them, BU is implemented as a hardfork upgrade, whereas Core relies on softforks. These two approaches are mutually incompatible: unlike a hardfork, a softfork is "forward-compatible" in the sense that blocks mined using the new rules can still be processed by non-upgraded clients (for additional details see Section 2.3).

If the community remains divided on which approach to support, then the result may be a schism, where each faction maintains a distinct fork of Bitcoin with mutually incompatible consensus rules.[5] Both blockchains will diverge post-fork,

[5] A schism has previously occurred in the case of Ethereum, whose *TheDAO* hardfork precipitated a split into Ethereum and Ethereum Classic.

ValueShuffle: Mixing Confidential Transactions*

Tim Ruffing
Saarland University

Pedro Moreno-Sanchez
Purdue University

In Bitcoin's initial design, privacy plays only a minor role. The initial perception of Bitcoin providing some anonymity and fungibility has been refuted by a vast set of academic works [1]–[4]. This state of affairs has led to a plethora of privacy-enhancing technologies [5]–[10] aiming at overcoming these shortcomings either by defining entirely new cryptocurrencies [11], [12] or without breaking with the fundamental design of Bitcoin.

A prominent example for the latter approach is Confidential Transactions (CT) [5], a novel transaction format to enforce payment value privacy in Bitcoin by hiding transacted values in homomorphic commitments. Another example is Stealth Addresses (SA) [6], a mechanism for payers to generate unique one-time addresses for improved payee anonymity.

To achieve payer anonymity, the most prevalent approach that retains compatibility with Bitcoin is coin mixing: A group of users exchange their coins with each other, effectively hiding the relations between funds and owners. Users jointly generate a CoinJoin [13] transaction that enables an atomic transfer of funds to fresh output addresses and prevents theft by design. If users exchange their output addresses by means of an anonymous P2P broadcast protocol [10], [14], mixing is possible in a decentralized fashion (*P2P coin mixing*) while making sure inputs cannot be linked to outputs even by malicious users in the mixing, and such malicious users cannot prevent the honest users from successfully completing the protocol.

However, the aforementioned approaches focus typically on just one aspect of privacy (payer anonymity, payee anonymity or payment value privacy), leaving the privacy landscape in Bitcoin orphan of a comprehensive privacy solution.

a) Challenge: To achieve comprehensive privacy, it is necessary to combine CT, SA, and coin mixing into a single solution. SA or other means to generate one-time addresses can be easily combined with coin mixing, but while CT has in fact been designed with CoinJoin mixing in mind, it is not clear that the trust models of CT and anonymous P2P broadcast protocols, which are required for decentralized coin mixing, can be made compatible. The design of CT assumes that a transaction is created by just one user, whereas in coin mixing it is a group of users who jointly need to create a CoinJoin transaction in a decentralized P2P fashion. A naive non-solution is that the users reveal their balances and the corresponding secrets (the blinding factors for the homomorphic commitments used in CT) to each other, and then create a CoinJoin transaction. However, this is not an option for coin mixing because the users typically do not trust each other.

*This is an extended abstract of the work "Mixing Confidential Transactions: Comprehensive Transaction Privacy for Bitcoin" which appeared at the Bitcoin Workshop 2017 and is available at https://eprint.iacr.org/2017/238.pdf.

I. Mixing Confidential Transactions

In this talk, we present ValueShuffle, the first coin mixing protocol compatible with CT. It enables a group of mutually distrusting users to create a CoinJoin confidential transaction, without revealing the relation between inputs and outputs or their payment values to each other. Since ValueShuffle successfully combines coin mixing, SA and the CT proposal, the resulting currency provides comprehensive privacy, i.e., payer anonymity, payee anonymity and value privacy. Since it builds upon CoinJoin, ValueShuffle inherits a variety of features crucial to its practical deployment in the Bitcoin ecosystem, e.g., compatibility with blockchain pruning.

By combining coin mixing with SA and CT, we exploit synergies which make P2P coin mixing both more efficient and more practical, thereby releasing the full potential of coin mixing. We achieve that goal by overcoming the two main limitations of current coin mixing approaches.

First, all forms of coin mixing have been heavily restricted to mixing funds of the same value, because otherwise it is trivial for an observer to link inputs and outputs together just based on their monetary value. Adding value privacy to coin mixing removes this restriction entirely but comes with the challenge of proving to the network that no money is created in the mixing, since payment values are no longer in clear.

Second, current P2P coin mixing protocols suffer from the problem that users are required to mix their funds (in a CoinJoin transaction) by sending them to a fresh address of their own first, which removes the trace to the owner. Only afterwards can users spend the mixed funds to a payee in a second transaction.

This two-step process renders mixing expensive for users, who pay additional fees and need to wait longer, and for the entire Bitcoin network, which has to process essentially twice the amount of transaction data. This is highly undesirable and creates a conflict between privacy and efficiency.

In ValueShuffle, instead, we rely on SA and CT to enable users to send their funds directly to the expected receivers in the CoinJoin transaction, which is arguably the most desirable mode of use of CoinJoin.

A. Overview of ValueShuffle

ValueShuffle is an extension of the P2P coin mixing protocol CoinShuffle++ [10], which is the result of instantiating the efficient message mixing protocol DiceMix [10] in the setting of CoinJoin-based coin mixing. To connect CoinShuffle++ with

ZeroLink: The Bitcoin Fungibility Framework

Authors

nopara73,
Hidden Wallet,
adam.ficsor73@gmail.com

TDevD,
Samourai Wallet,
PGP

Acknowledgements

Special thanks for Adam Gibson and Chris Belcher from JoinMarket, Ethan Heilman from TumbleBit, Dan Gershony from Breeze Wallet and Kristov Atlas from Open Bitcoin Privacy Project for tolerating my constant bugging and bothering to acquire their invaluable reviews, suggestions and feedbacks.

Support

186n7me3QKajQZJnUsVsezVhVrSwyFCCZ

Concurrency and Privacy with Payment-Channel Networks*

Giulio Malavolta†
Friedrich-Alexander University Erlangen-Nürnberg

Pedro Moreno-Sanchez†
Purdue University

Aniket Kate
Purdue University

Matteo Maffei
TU Wien

Srivatsan Ravi
University of Southern California

1. INTRODUCTION

Permissionless blockchains protocols such as Bitcoin are inherently limited in transaction throughput and latency. Therefore, in the forethought of a growing number of Bitcoin users and most importantly payments about them, scalability is considered today an important concern in the Bitcoin community. Among alternative proposals, the use of Bitcoin *payment channels* [1, 8] to realize off-chain payments has flourished as a promising approach to overcome the Bitcoin scalability issue. As a generalization, current efforts leverage a path of opened payment channels from the payer to the payee with enough capacity to settle their payments, effectively creating a *payment-channel network (PCN)* [8].

Many challenges, such as liquidity, routing scalability, privacy or concurrency, must be overcome before a PCN is widely deployed. Here, we study privacy and concurrency in PCNs and show an inherent trade-off between them.

The Privacy Challenge. Although it might seem that payment channels inherently improve the privacy of Bitcoin payments as they are no longer logged in the blockchain, the actual privacy guarantees are not clear among the community. Recent research works [3–5] propose privacy preserving protocols for payment hub networks, where all users perform off-chain payments through an unique intermediary. However, it is not clear how to extend these solutions to multi-hop PCNs. The Lightning Network [8], a multi-hop PCN, does not provide all the privacy guarantees of interest in a PCN. For instance, a payment routed through a path of payment channels includes an identifier that can be used by intermediate users to derive who is paying to whom [3]. In summary, the lack of rigorous definitions for the protocols, threat model and the privacy notions, hinders an in-depth security and privacy analysis of PCNs.

The Concurrency Challenge. The consensus algorithm eases the serialization of concurrent on-chain payments. However, the bulk of off-chain payments in a PCN are no longer required to be added to the blockchain and therefore no user has a view of all concurrent off-chain payments at any time for their serialization. On the other hand, individual users cannot easily avoid concurrency issues either since a payment might involve several intermediate users. As PCNs scale to a large number of payments, concurrent payments are likely to happen and concurrency issues must be thoroughly investigated.

*A full version of this work has been accepted at CCS'17. A preprint version is available at https://eprint.iacr.org/2017/820

†Both authors are considered co-first authors.

2. OUR CONTRIBUTIONS

In this talk, we plan to present the following results from our research on the concurrency and privacy issues of PCNs:

- We formalize for the first time the security and privacy notions of interest for a PCN, namely *correct balance*, *value privacy* and *relationship anonymity*. Intuitively, correct balance ensures that no honest user loses coins as intermediate hop in the payment path. Value privacy ensures that the payment value is not leaked to off-path users. Finally, relationship anonymity guarantees that payer and payee of a payment remain anonymous among a set of possible payers and payees, even in the presence of an on-path adversary.
- We study the concurrency issues in PCNs and present two protocols Fulgor and Rayo that tackle this issue differently. Fulgor is a blocking protocol similar to other payment networks such as credit networks [6] that can lead to deadlocks where none of the concurrent payments go through. Overcoming this challenge, Rayo is the first protocol for PCNs guaranteeing non-blocking progress: At least one of the concurrent payments terminates.
- We characterize an arguably surprising tradeoff between privacy and concurrency in PCNs. In particular, we demonstrate that any PCN that enforces non-blocking progress inevitably reduces the anonymity set for sender and receiver of a payment, thereby weakening the privacy guarantees. Therefore, in contrast to Fulgor, Rayo inevitably provides relationship anonymity only against an off-path adversary.
- We formally describe Multi-Hop HTLC, a smart contract that lies at the core of Fulgor and Rayo and which ensures privacy properties even against users in the payment path from payer to payee. We provide an efficient instantiation based on the recently proposed system ZK-Boo [2], improving on previous proposals [9] by reducing the data required from 650 MB to 17 MB, the running time for the prover from 600 ms to 309 ms and the running time for verifying from 500 ms to 130 ms. Moreover, Multi-Hop HTLC is compatible with the current Bitcoin scripting system.
- We have implemented an evaluated a prototype of Fulgor and Rayo. Our results show that a privacy-preserving payment in a path with 10 intermediate users can be carried out in as few as 5 seconds and incurs on 17 MB of communication overhead. These results are in line with with other privacy preserving payment systems [6, 7]. Additionally, our evaluation shows that Fulgor and Rayo can scale to cater a growing number of users with a small overhead that can be further reduced with an optimized implementation.

Optimizing fee estimation via the mempool state

Karl-Johan Alm <karl@dglab.com>
DG Lab

September 12, 2017