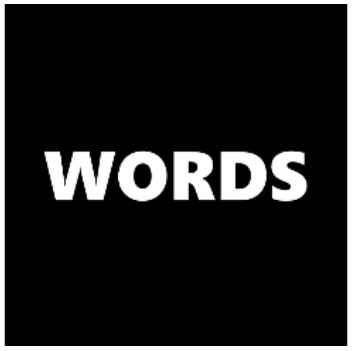
CY18 April

A collection of Bitcoin commentary from the brightest minds in the crypto community.

Contents

Goals and Scope



Crypto Words is a journal of Bitcoin commentary, established February 13, 2019. Its purpose is to document and advance commentary and research in disciplines of particular interest to the Bitcoin community. The journal is broad in scope, publishing content from original research, essays, blog posts, and tweetstorms from a wide variety of fields, especially governance, technology, philosophy, politics, and economics, but also legal theory, history, criticism, and social or cultural analysis. Its broader mission is to capture the conversations and think pieces in the Bitcoin space for current and future researchers. *Crypto Words hopes to* continue and expand the tradition established by publications such as the *Journal of Libertarian Studies* and *Libertarian Papers*.

History

There exists a gap in Bitcoin publishing. For authors with commentary and scholarly papers on topic, the choice of publication outlets is relatively limited. The number of journals that serve as outlets for crypto research is in any event too small, as the number of crypto thinkers continues to grow with every market cycle.

This generation of Bitcoin thinkers have limited places to submit thought pieces for publication. Content is scattered across the web, and in some cases behind paywalls which prevent the free flow of information. With the advent of the Twitter and blogging, authors also now have the option of self-publishing: they post the content to their own site or some private site, link it in a blog post, or post a working paper. But this is obviously not the best way to document and publish. What is needed is a journal that takes full advantage of the possibilities of the digital age as a go to resource for think pieces in the crypto space.

Enter *Crypto Words*. Published independently, *Crypto Words* is a journal that welcomes submissions on a range of topics of interest to the crypto community. In addition to conventional research articles, we welcome review essays blog posts, tweets as well as papers in other formats, such as distinguished lectures. Finally, wherever possible, content on this site is licensed under a Creative Commons Attribution 4.0 License. Authors retain ownership without restriction of all rights under copyright in their articles. *Crypto Words* is open access, and we encourage readers to "read, download, copy, distribute, print, search, or link to the full texts of these articles…or use them for any other lawful purpose." We want our ideas read, spread, and copied. We welcome discourse and debate.

Support Crypto Words

The posts and journals published here have been carefully curated and crafted as a true labor of love. If you've found any of this content useful here's how to show your thanks and keep the project going.

Send Bitcoin | tippin.me | Send CashApp | Send PayPal

Spread the word

Have a website or use social networking sites like Twitter, Facebook, or LinkedIn? Please consider sharing the content found on Crypto Words or linking to https://cryptowords.github.io.

Follow us on social media

We post regularly on Twitter and use it as our main form of communication. — We don't rapid fire posts but add commentary where we see fit. Posts are typically links to our content here, trolling nocoiners, sarcastic remarks, and other things regarding development of this site.

If these sorts of things interest you, follow along on:

Twitter

Subscribe to our newsletter

We publish our journal monthly and share it via Twitter and via newsletter. Consider subscribing to the newsletter. If you're not on Twitter all day, it might make sense to subscribe so you never miss a publication.

Our pledge

- We will never sell you out.
- We will never shill you shitcoins.
- We will only deliver what is promised.

Subscribe

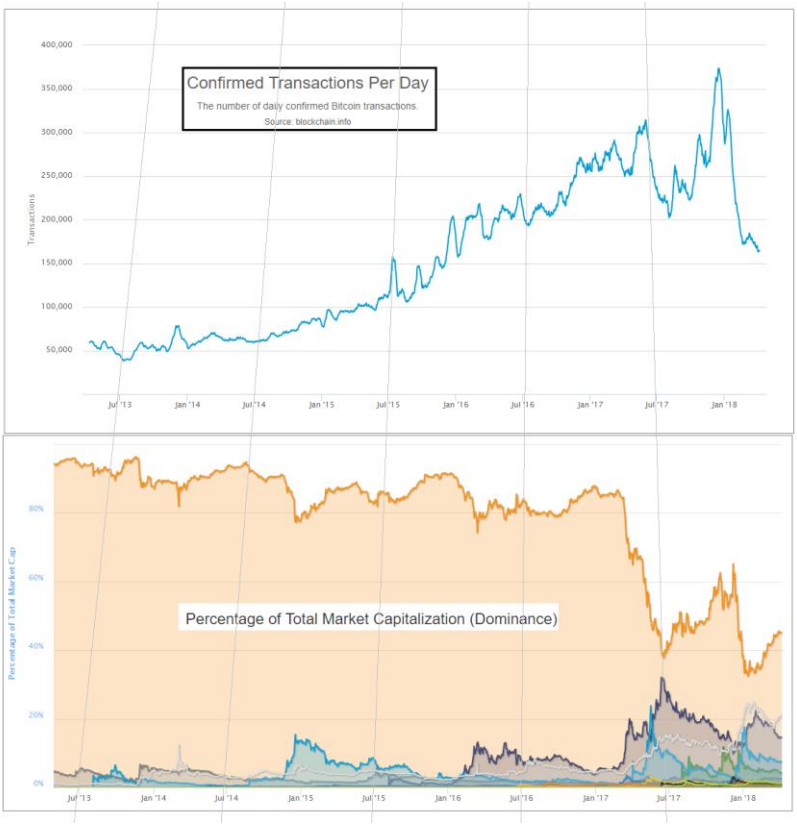
Bitcoin Post-Maximalism

By Paul Sztorc

Posted April 7, 2018

Has something happened to "Bitcoin Maximalism"? Is Bitcoin "Qwerty" (established and standardized) or "Esperanto" (impractical and perfectionistic)?

```
Israel is a scattered sheep;
--Jeremiah 50:17
```



Why Bitcoin Maximalism

In the past I've been very dismissive of Altcoins, and endorsed the philosophy of "Bitcoin Maximalism" (that Bitcoin is all you need).

Here were three of my biggest reasons:

1. **Network Effects** – In the past, Bitcoin had 90+% of the crypto market, and the remaining 10% was distributed among many terrible projects. Thus, Bitcoin's network effect effectively shielded it from competition.

2. **Utility** – With a few important exceptions, no Altcoin offered anything new to the user. Instead, the alternative explanation (that the Altcoin-creators were just taking people's money) was overwhelmingly accurate.
3. **Sidechains** – I assumed sidechains would be eventually invented. At which point, they'd absorb features of rival blockchains.

The first reason, in particular, has not held up well over time.

1. Network Effect Collapse

The market share (among investors) of "Bitcoin Core" (the healthiest candidate) has fallen to 45% or so.

Many people, myself included, believe that this downturn is only temporary. But as new data rolls in, it is time to take seriously the alternative theory: that the "network effect shield" has departed – or at least significantly weakened. These data are: that it has now been over two years since Bitcoin was last above 90%; it has been 12 months since it was above 80%; it has been under 50% for 7 of the last 12 months, including the most recent three months; during which it reached its all-time low (of 32%, under a third).

Most perplexing is the relentless, stable, multi-year progress of the "Others" category. This is a direct challenge to the logic of network effects.

Furthermore, many of Bitcoin strongest defenders have jumped ship, and a few of these are even socially secure enough to admit so openly. Roger Ver, indefatigable promoter and angel investor, owner of the Bitcoin.com domain, now favors Bitcoin Cash. Brian Armstrong, CEO of the $1B Unicorn company Coinbase (which occasionally has the most-downloaded smartphone app), now favors Ethereum.

Reasons 2 and 3 have not done as well as I've hoped, either.

2. Rising Altcoin Utility

The Altcoins of today *do* offer their users a real value proposition – two in fact.

Value Prop 1 – Cheap Blockspace

At minimum, Altcoins offer users "cheap blockspace".

Some users rely on "round trip" transactions. I define these as situations where:

1. User wants to buy something, for (say) $20.
2. User then spends $20 on "crypto", and immediately spends that crypto on their desired end-product.

3. The merchant receives the crypto and immediately sells it for $2 For such txns, cheap blockspace is ideal.

Furthermore, services (ShapeShift, Poloniex, et al) have evolved to make these markets user-friendly and liquid. It is now easier than ever to accept currency from "exotic blockchains" for payment. These days, if you can accept one crypto, you can accept them all.

Value Prop 2 – New Ideas

Today's Altcoins do more than just offer cheap payments.

They also offer *permissionless innovation*. Certain ideas, such as Monero, Siacoin, Namecoin, and Zcash, cannot efficiently be tested in any other way. Many ideas that I originally felt were laughable, such as Dash's "marketing budget", have nonetheless proven to be effective[1].

Of course, these experiments *should* have been done on sidechains (which we will turn to in a moment).

In Comparison to Bitcoin

As to the first value proposition (cheap payments), much has already been said. I have written an article with my nuanced views on the subject.

More important is to discuss the second value proposition. For, while the Altcoins try 1001 new ideas a day, (most bad, but occasionally one or two good[2]), Bitcoin instead has retreated into an overcautious and highly-pretentious paternalism. I can count on three hands, the number of times I have been personally given the "we need to make sure the airplane doesn't crash" metaphor, by extremely-senior members of our industry. Somehow, these people are oblivious to the fact that [1] they don't own the plane in question, [2] it is being flown via remote control, from the ground, and [3] the plane's owner can freely make a near-infinite number of copies. It is not so much a passenger-laden plane, but a flight simulator virtual-reality videogame.

The Altcoin ideas are judged, appropriately, by the user. Bitcoin, in contrast, is now tending to choose its ideas based on how impressive they are to other members of a pseudo-academic pseudo-bureaucracy. The emphasis is not on scientific progress, it is instead (I am sorry to say) on racketeering – in other words, on generating a need for "experts" (ie, paid consultants), and building a justification for an endless series of prestigious "conferences" (ie all-expense-paid "parties" in exotic locations).

Let me be more constructive with my criticism.

A scientific environment requires certain features, including: a tolerance for dissent, an appreciation for discussion, a rejection of arguments-from-authority, and an optimistic outlook (one that we can make "win win" improvements to situations through better ideas, rational debate, and criticism).

Above all, a scientific environment requires Karl Popper's demarcation for science: that in addition to looking for *confirming* evidence of theories, we must try to *falsify* (ie, break) our favorite theories. Altcoins represent one method of falsification – trying the idea and watching to see if it fails. Sidechains are an even better method. "Peer Review" is only science if the peers are helping the author meet some *objective* external criterion (ie, one that exists independently of the peers and their opinions). Otherwise, peer review becomes a self-referential popularity contest. The point, since so pervasively and consistently misunderstood, bears repeating: Peer review is supposed to be a *cheaper* realistic "simulation" of reality. It is not a popularity contest!

Unfortunately, for the significant questions[3], the atmosphere of science is departing from Modern Bitcoin.

A Contempt for Measurement

One smoking gun is the reaction of both LargeBlockers and SmallBlockers to the idea of fork futures.

Futures prices, (unlike "tweets" or "conference presentations), have the unique ability to speak for everyone, and not just their author. For that very reason, they have the unique ability to singlehandedly predict the fate of any fork[4].

Despite this, there was no interest in creating such markets. When they were created anyway, the losing side refused to acknowledge them as legitimate. When I proposed a way of making them more legitimate, the losing side was not interested!

This disinterest parallels the shameful behavior concerning the "bitcoinocracy" numbers – the evidence changed for, and then against, the SmallBlockers, and so they shifted from rejecting, to endorsing, and finally to rejecting the legitimacy[5] of this empirical evidence.

This is a root and branch rejection of the value of experimental testing. Very shameful. It acts to establish a "technocracy" of ruling bureaucrats (again: accountable to each other, and not to the customer). It exploits the power in the remaining network effects, and uses it to enable a monopoly – in other words, uses it to ensure that dissatisfied customers have no recourse.

Partisan Media

As the scientific atmosphere declines (and, please, do not confuse science with engineering), standards of discourse have declined as well. Today, it is impossible to express any view on "scaling" without it being immediately pigeonholed into a "Republican" (SmallBlocker) or "Democratic" (LargeBlocker) category. We even have our own RNC and DNC, and our own party leaders and campaign managers. Any project or solution which attempts to be "non-partisan" might as well be a third party candidate seeking election. One can even be "found guilty" of *listening to the wrong Bitcoin show* or *attending the wrong conference*. But a real scientist, embracing falsifiability, would be *sure* to attend conferences given by people he or she doesn't agree with.

Instead we have two different flavors of a dystopian One Party State – authoritarianism on the right, and on the left a kind of "intellectual communism" where everyone gets a trophy for their ideas, no matter how dumb these ideas are (especially the low standards of Classic / Unlimited leading to assert(0) bugs etc etc).

Clever Altcoiners have noticed these deficiencies (and the insecurities they inspire) and exploited them. Buterin, for example, is careful to back the minority side in the BTC BCH conflict; and Dash started up a meme about their "governance" solution (whatever that was) to profit off of dissatisfaction with Bitcoin governance. One can dismiss these maneuvers are mere campaigning, but they are **only** possible because of real flaws that actually exist in the Bitcoin community.

3. Sidechain Apathy

Finally, what of the hope that sidechains tech will obliterate the Altcoins?

No one alive is in a better position to answer than I. After Blockstream gave up on sidechains in 2015[6], I wrote my own idea in November of that year. It remains, to this day, the only concrete proposal for P2P sidechains, let alone the only implementation. I've presented on sidechain theory and risks, and at Scalings II and III (the II presentation was a small "WIP" and these were not recorded, I believe).

My view is that the scaling conflict *is* important, and that sidechains are the best way to resolve it. In fact, my current view is that *sidechains are the only way* to resolve the conflict. This is because the disagreement is actually about "node costs", and not about transaction throughput.

Blockchain technology is inherently "consensus"-based. But since each person is different, there is a limit to how large a community can grow before there is infighting. Sidechains resolve these issues.

The Problem of Low Interest

Despite this, interest in Drivechain has consistently been low, among Democrats and Republicans alike. Democrats (supposedly) control >50% of the hashpower, and could unilaterally gain a blocksize increase by adding sidechains. But they are not interested. MimbleWimble, originating deep in Republican territory (#bitcoin-wizards), now plans to launch as an Altcoin.

What explains this profound apathy for sidechains?

I will first give some prudent reasons (ie "happy" reasons), and then what I believe to be the "real" reasons.

A. Prudent Reasons

Here are some simple and, at first glance, believable reasons for sidechain apathy.

1. Sidechains Are Hapless / Useless

Perhaps sidechains are just a bad idea? And, since there's no point wasting time talking about a useless feature, people rightly don't talk about it.

I think that the causality here is reversed: people become disinterested first, and this disinterest drives them to make lazy, error-prone comments.

First, I can't find any evidence that sidechains are a bad idea – I give a lengthy defense of drivechain (currently the only sidechains proposal) in this video and in my FAQ. Instead, there is profound evidence of bad *claims* that sidechains are a bad idea – most critics admit that they made no effort to understand the idea before criticizing it[Z].

However, more fundamentally, even if drivechain *were* bad, it is a soft fork. So, it can be **freely and completely ignored** by disinterested users. And it actually **cannot be prevented** if miners decide to use their hashrate to unilaterally activate it.

So I don't see how talking about it could be a waste of time. If bad, it should be talked about, because it is unpreventable. Criticisms shouldn't be of *it*, they should only be of the opt-in-ers.

So a better explaination is that the "bad idea –> therefore disinterested" causality must be reversed, I think. The trust is "disinterest –> therefore ignorant comments".

2. Sidechains Are Inherently Boring

Perhaps sidechains are inherently boring.

But this does not square with the attention they get from Bitcoin Media, and their disruptiveness to the competitive landscape and to people's investments.

3. People are Busy

Perhaps people are just too busy with everything that is going on.

But this explanation would apply equally well to every new idea. And drivechain is a very *old* idea, it is much older than SegWit and the spec was published in Nov 2015.

And there is no "moral" or "religious" objection to talking about sidechains, either. I can get principled developers to review the code, if I pay them. Disinterested third parties have also offered to pay for drivechain code review, with some success.

So the *potential for interest* in sidechains is there, but the inherent interest is just disproportionately low.

So, then, why is that? Here I present what I feel are the "real" reasons.

"Real" Reasons

4. Reputational Downside / Different Risk Profile

Crypto-commentators care a lot about their reputation[8], as it can lead to lucrative job offers (in every sense – paid well and no *work* required), access to capital, invitations to luxurious conferences, and personal fame/prestige.

Moreover, BTC-professionals take pride in their work (as they absolutely should).

And thus, no one wants to be on the hook for a "BTC disaster".

And sidechains, along with their benefits, do present a scary new risk. Unlike the "code risks" in, for example, P2SH or CLTV, these risks can not be systematically hunted down and eliminated. So, commentators may see themselves as in a similar situation to the FDA [in the US], or an academic IRB: they will be disproportionately blamed if something goes wrong, but will not disproportionately benefit if everything goes well.

These risks are freedom-enabling, and entirely opt-in. But who knows if the YouTube Audience / VC Investors / Program Committee / etc will see it that way? Instead it pays to "care" about the user's funds, especially very loudly and at no cost to oneself.

5. Training One's Replacement

For existing Bitcoin Core developers specifically, the above position might be taken even further.

Instead of the example of the FDA commenting on a drug, we might instead give an example where some of the world's most prestigious doctors are asked to comment on a "magic infinite health pill" that was invented by a non-doctor.

It is a clear conflict of interest – if society adopts the pill, the doctors will be out of a job. But we wouldn't expect these sophisticated doctors to object so directly!

Instead, we would expect them to resort to various pretexts – they would say "Well, we had better make sure that this pill is absolutely safe before we tell everyone to take it". Even though by the time that absolute safety is established, many people will have likely died needless deaths.

Hence it really is true that the scaling debate is "about control". Since sidechains take control away from current elite Core developers, we would expect them to oppose sidechains.

6. Extreme Polarization

Human disputes, of all kinds, will reliably collapse onto a single dimension. This is why, in America, those who are "pro-choice" also tend to be "pro-Union", despite those positions being objectively unrelated. This is simple math: a large unified group has the muscle to sequentially crush a set of smaller uncoordinated groups.

Those who are audacious enough to vote third party have a vague awareness of this, and usually know that they are "throwing their vote away" and hurting their own causes. If you join one of the two major groups, your influence will be very small. But if you join neither it is likely to be zero.

A LargeBlock sidechain is a competitor to both the LN (favored by Republicans) and a hardfork blocksize increase (favored by Democrats). Each party campaigns on a platform where they are the wisest (or, more accurately, they "have" the wisest) and they know what's best.

Thus, to support a largeblock sidechain would be to oppose the party leadership. But as I've just explained, this leads to one being rejected by one's own party, and being unable even to join the rival party. So commentators (wisely) downplay their interest in a sidechains solution. Today, nearly every person, and every media outlet, has allowed themselves to be captured exclusively by one party.

7. Free Rider Problem

This one is very simple. The free rider problem is a one of immense practical importance.

For Altcoins, of course, there is no free rider problem, because there are alt-owners who profit (*disproportionately to everyone else*) from the success of the Altcoin.

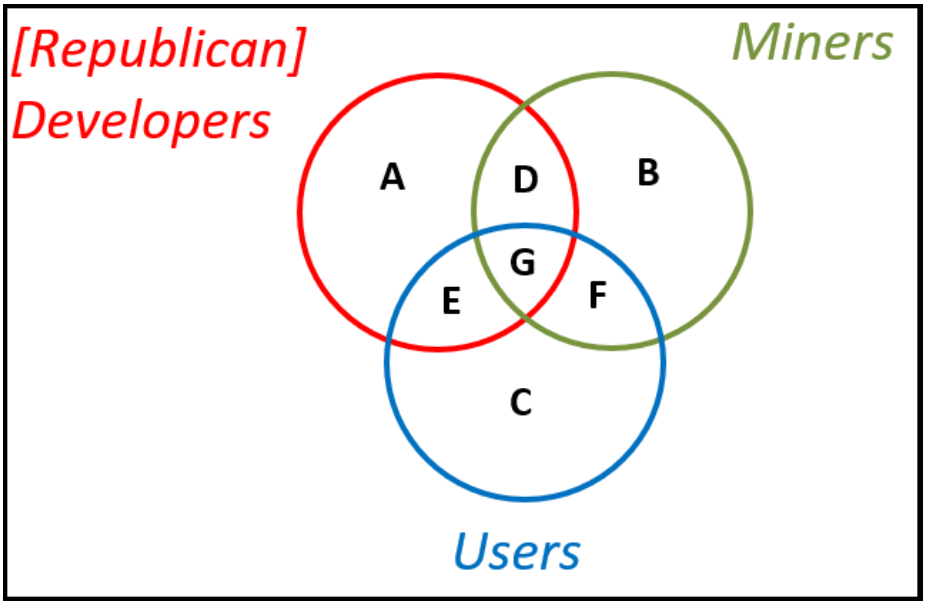
But with sidechains, we have a situation where someone must do the work, at some cost to themselves, and yet the benefits are diffused across all Bitcoin owners.

But this problem is common to all Bitcoin R&D, so I don't see why it should apply especially to sidechains.

Misaligned Incentives in Bitcoin Generally

Incentives in Bitcoin are not always perfectly aligned.

Below I present some Bitcoin ideas/events/projects, and sort these into the groups that support them: Miners, Users, and [Republican] Developers.



Alignment

G – All Bugfixes; "Scaling Bitcoin" 1 and 2; CheckLockTimeVerify; Fraud Proofs (?)

Some Alignment

D – High txn fee-revenues[9] (see here); the unending scaling conflict (think 1984 "War is Peace", and govt 'racketeering')

E – Lightning Network (miners prefer on-chain); "Scaling Bitcoin" III; Blockstream/ChainCode/etc; soft forks (miners prefer hard, although I honestly don't know why)

F – Decentralized Sidechains (devs prefer bitcoin-dev-based permission, and federation/subscription); SPV/SPY Mining (devs prefer FIBRE)

Less Alignment

A – Federated/subscription-based sidechains ; "Scaling Bitcoin IV" ; high txn fee-rates (?)

B – [SegWit-Incompatible] ASICBoost (although I/others strongly dispute the relevance); SegWit2x Project (users prefer 1x, according to data from futures markets)

C – Low fee-rates; The SegWit UASF; Fork Futures; Spinoffs/Altcoins (empirically)

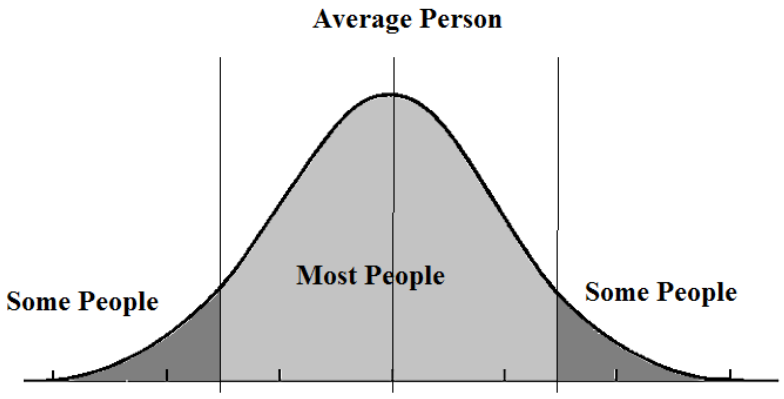
I could have included more groups, especially "Industry", "Democratic Developers", and "Cults of Personality", but of course a two-dimensional figure simply cannot capture all of that. My point is that incentives do not always align.

Network Effects as "Rule of the Average"

Qwerty and Esperanto

For better or for worse, the dumb people of the world form an intransigent minority, because they literally *cannot* appreciate good ideas. This is one reason why we are stuck with the QWERTY keyboard, for example (even those who type in Dvorak are doing so *on* a QWERTY keyboard layout). English is the international language, despite being awkward and hard to learn – Esperanto, in contrast, is very easy to learn but is spoken by no one. If you disqualify Esperanto, then consider French: it is easy to learn (compared to English), and was once the international language of diplomacy, once learned internationally by the upper classes and backed by a powerful nation. Yet today they all speak English in Quebec.

The fact of the matter is that anything with network effects is going to ultimately be ruled by *the middle* of the bell curve: the average people.



"Think of how stupid the average person is, and realize half of them are stupider than that." – George Carlin

The Centrality of Timing

In these analyses, **timing** is important: every project starts with zero users, so networks effects are small (and meritocratic effects dominate). But as time goes on, the project will attract more users, and so the network effects will become more important. Eventually, the network effects outweigh the meritocratic effects.

Now, I'm just guessing here, but I think we have reached the post-expert phase. Anyone who can compare the 2018 TNABC to even the TNABC a year earlier will discover that the knowledgeable folks are vastly outnumbered. (And by "knowledgeable", I mean "knows what a private key is".) And, if the playing field is too ambiguous, these un-knowledgeables are going to glue the future to whatever shiny object can attract their attention first[10].

Secondly, the *marginal* meritocratic effects do not seem to be that significant. By this, I mean that the "Bitcoin vs Altcoin" differences are very tiny compared to the "Bitcoin vs Traditional Money" differences. Someone who *needs* financial sovereignty must abandon modern fiat currencies, but whether they transact in BTC or LTC or ETH will make no difference to them, and investors will need to invest in whichever money is the most recognizable. The differences in node cost, or in privacy, are not as relevant (most lay users care about neither)[11].

Conclusion

In a video discussion that accompanies today's articles, Daniel Krawisz advised me to deploy Drivechains on both BTC and BCH (by hard forking one or both if necessary). But he did not advise me to also add drivechain to Bitcoin Gold or Diamond etc.

It struck me as good advice – "Post-Maximalism" doesn't need to mean "Altcoin Pluralism" – instead it can just mean "two competing Bitcoins". So we don't need to go from 1 to infinity, we can stop at 2. And perhaps we only stop at 2 for a short time, before returning back to 1 (for example if sidechains do, in fact, eliminate Altcoins).

The emphasis on **two** competing projects reminded me of Karl Popper's argument in favor of a two-party state. He argues that a system can become healthy, as long as the losing team becomes *desperate* upon their loss, and willing to take risks and make changes. Pretty good advice!

Update (16.04.2018)

Sergio Demian Lerner correctly points out that he did write a drivechain BIP+code in 2016. It had pros and cons relative to mine, some of which the two of us discussed together at Scaling Milan in late 2016 (he definitely helped improve our work).

He also finishes my unfinished 7th point, by reminding us that Altcoins/ICOs can draw away sidechain talent. But this does not happen as easily for SegWit or P2SH.

It also occured to me that working on sidechains is probably perceived as disloyal to Bitcoin. After all, what you're really saying is that you want the freedom to leave this party, and go start your own party somewhere else. That's bound to insult the current party-hosts, to some extent. So we can understand why they wouldn't put sidechains on top of their list (although it is certainly a little presumptuous/creepy that they do not).

Footnotes

1. As Nassim Taleb says in his most recent book, "Rationality does not depend on explicit verbalistic explanatory factors; it is only what aids survival, what avoids ruin … Rationality is risk management, period." Many Bitcoin supporters today have switched from being pro-Taleb to militantly anti-Taleb, and I think they've switched without even realizing it. ⮌
2. Again, we know that they are objectively good, by Taleb's criterion. ⮌
3. By "significant questions", I mean: "How do we protect Bitcoin from arbitrary changes via a hard fork?", "How do we handle situations where stakeholders disagree?", "How do we prevent an Altcoin from competing with [ie replacing] Bitcoin?", "Can we find a block size solution that works for everyone?". For trivial, non-controversial matters, we may still see some scientific features. ⮌
4. See "if a miner would rather hold B2X, they could earn it four times faster by mining B1X and trading it for B2X" by Dan Robinson here. ⮌
5. It was always better than nothing but highly flawed, as I discuss in the Fork Futures article. The point is that people's reaction to it should not depend on their political position, when in reality it depended strongly. ⮌
6. I feel it necessary to explain that Blockstream has two projects which it repeatedly claims are sidechains but which actually are not. They are not sidechains because they lack the distinctive feature –the "two way peg"– and as a result neither can be used *functionally* as a sidechain (ie, Bitcoin users cannot opt in to new innovative features). The two projects are "Elements" and "Liquid" – the first is just an Altcoin (in precisely the same way that LTC, XCP, and ETH are Altcoins), and the second is just a multisig wallet. In fact, the phrase "federated sidechain" is nonsensical: the major innovation behind all "chains" (especially Bitcoin itself) is **mining**, and mining is distinctive because it *lack* a set of signers. — — A few brave people spoke openly about their utter shock and confusion. But most people made a strong effort to mask their disappointment. ⮌
7. In fact, many of the comments are so bad, and come from people who are [in non-sidechain contexts] relatively smart and reasonable, that I think

"preemptive disinterest" can be the only explanation. For example, Luke Dashjr and Matt Corallo on bitcoin-dev did not see that "most PoW chain" is equivalent to "most $$ spend on chain", despite the fact that this is just one inferential step (indeed, a single multiplication operation) away. Jorge Timon still does not understand that sidechains are supposed to be optional – these are sidechain paper co-authors! Peter Todd insisted that sidechains be compared to his (nonexistent and, I believe, flawed) "client side validation" project, somehow not realizing that "client side validation" [ie, "opt in validation"] is exactly what sidechains do. — — If these people were putting in their best effort, and the resulting commentary were this bad, we would be forced to determine that all of the critics were just hopelessly unintelligent. And I do not believe that that is the case. (LargeBlocker critiques of sidechains, of course, are even worse. Not to belabor the point, but technical critique is not their forté.) ⮌

8. Again, disturbingly, this is one of NNT's indicators for an IYI. And I'm afraid that I do think it applies here – remember that we have a situation where people are being assessed by their peers, and not by their customers. ⮌
9. Unfortunately, Bitcoiners often use the word "fees" to refer to two different concepts (of, occasionally, *opposite* meanings). On one hand, "fees" are the "fee rate", the "satoshi per byte" scalar that an economist would call "the price". But in other instances, "fees" is used to refer to "the value [PPP] earned by a miner for mining a block, if we exclude the block subsidy" – this is what economists might call "total revenue". On a standard supply-demand curve, the first is a coordinate on the y-axis, and the second is a two-dimensional area multiplied by an exchange rate. ⮌
10. From what little I've heard (which is very little indeed), it seems that technologists (especialy cryptographers) are making the same mistakes they made earlier with the internet itself, and for the same reasons. The mistake is to dogmatically embrace the principles of "independence", and especially to neglect the principles of "inter-dependence". Many of the Internet's earliest adopters believed that the Internet would ultimately take on a form that was much more private, more "flat" and more decentralized than it is today. But it has today's shape in order to achieve maximum usability and convenience. While "the stacks" were taking shape, dogmatic cryptographers were, in a sense, asleep at the wheel. While GMail and Facebook were growing, they were busy virtue signaling to each other about to make perfect email encryption. In my experience, *elite* cryptographers are very similar to the rest of us: they care about winning the esteem of their peers, and not as much about "making the world a better place" (although they know that they must pretend, convincingly, to be interested in MtWaBP, or they know to just turn their brains off when it comes to the matter). Thus they will *proudly* refuse to interact with the real world, regarding "persuadability" to be an attack vector or (worse) a character flaw. But in network-effect-related matters such as these, "persuadability" is just another word for "rational decision making". So these technologists mask their ineptness in faux stoicism; fiddling out

impressive-seeming tunes to each other while Rome burns (but the tunes will be "about" fire prevention, of course). For success, we must find the technologists who do not care about the opinions of other technologists. ⮌

11. And privacy is far less relevant in crypto than it is in fiat, because confiscation is impossible (the extortionist may kidnap the victim, but must still *persuade* him/her to part with the coins – the victim can still choose to withhold the coin. This seemingly-trivial distinction is, I think, underappreciated. The victim may [if imprisoned] retain his option to cut a special deal with the prosecution, or with specialist lawyers or corrupt politicians. Or else they may still send [or bequeath] coins to friends/family. When police confiscate physical cash, the victim has no negotiating leverage. Users who *do* care about privacy will always be able to use mixing techniques. These include the simple ability to repeatedly send coins to oneself, imparting plausibile deniability. ⮌

The Many Faces of Bitcoin

By Murad Mahmudov & Adam Taché

Posted April 10, 2018

To nocoiners, gold-bugs, and Keynesians, the cryptocurrency space is best seen as a parasite infecting millennials with technobabble, forcing them to spout economic gibberish, and sucking them into believing the pipedream that a crypto-anarchist society could exist. To believers in the technology, cryptocurrencies represent an escape from the imprisonment of the traditional financial system in which they are forced to participate by being born; a system which has been plagued by inflationary monetary policy, monopoly by nation-states on money creation, malinvestment, and debt. To believers, cryptocurrencies are a starting point to rebuild honesty and a true measure of value in society among borderless, apolitical and decentralized systems.

The most prominent and powerful of the cryptocurrency communities — the Bitcoin community — has fractured into multiple factions based on desires for various directions to take the protocol and tribalism over different projects altogether. This article will explain some of the current motives driving these ideologies and try to express the reasoning behind this schism.

Although this article will be split into four main sections, there is certainly some overlapping thought between individuals who espouse these theories. The two schools of thought which we outline initially — bitcoin, first and foremost, as a store of value, and bitcoin cash as digital cash — are typically considered more mainstream, whereas the last two — Bitcoin as catalyst for something John Nash called "Ideal Money," leading to bitcoin-backed fiat currencies, and finally, looking at Bitcoin from the perspective of information theory — are less commonly known.

Four Theories About Bitcoin

Bitcoin was the first decentralized cryptocurrency ever created. It was released in 2009 as the culmination of nearly two decades of discourse on the key concepts within the cypherpunk community. It was cited by the anonymous founder(s) Satoshi Nakamoto to be inspired by Bitgold by Nick Szabo and B-Money by Wei Dai, two earlier attempts by well-known members at creating functional electronic currency.

Most individuals within the Bitcoin community envision an endgame where an implementation of Bitcoin will be a massively adopted cryptocurrency that is both a store of value and a medium of exchange. They see bitcoin eventually being the predominant global currency. However, the ideology then diverges sharply on on

how this can be accomplished, and which priorities should take precedence along the way.

Bitcoin As Money

Bitcoin presents us with an opportunity to reinvent gold, or even rethink money for the digital future. A number of economists have suggested that it may be more appropriate to evaluate items based on their degree of *moneyness.* According to this thinking, it isn't that something either is or is not money; on the contrary, many items can play a monetary role and some items can play this role more effectively than others. In a number of ways, bitcoins have a high degree of moneyness. They are more portable, durable, divisible, and scarce than both gold and government fiat currency.

As of today, bitcoins can best be described as digital commodities with monetary properties. According to the Bitcoin Maximalist interpretation of monetary history, it is likely that a new, scarce form of money would evolve roughly along the following lines:

1. Collectible
    2.
    - Store of Value
    3.
    - Medium of Exchange
    4.
    - Unit of Account.

Proponents of bitcoins as digital cash believe that utility should initially take precedence over store of value, and prioritize attaining the medium of exchange role before store of value by making payments as cheap as possible.

Those who believe bitcoin will become the future global monetary standard ascribe current volatility to the fact that bitcoin is undergoing the process of monetization, and that a global cognitive shift is slowly occurring. In their view, despite great volatility, the long-term parabolic ascent of the price is a testament to more and more people believing in a future world where Bitcoin is widely used.

Crypto-Austrians who consider themselves Rothbardians, such as author Saifedean Ammous, believe that bitcoin's disinflationary nature and cap on supply makes it the most sound money ever invented. They believe that bitcoin, with its fixed monetary supply, is the only fair form of money, as well as one which allows for the most efficient capital allocation by individuals and most efficient price signalling by the market as a whole.

Many individuals in this group are against the idea of fractional-reserve banking and consider it to be fraudulent. They believe that a fractional-reserve banking system is unlikely to emerge atop bitcoin, as bitcoins lack the physical centralization of gold, which forced settlements and clearance to necessarily pass through centralized choke-points, allowing governments to have complete control over the money supply, transmission, and the monetary regime at large. The governments had so much control that they were able to get rid of the gold-standard (which was organically chosen by the market over centuries) and introduce their own fiat standards, not backed by any commodity.

These individuals believe that fractional-reserve systems are simply unsustainable in the long run without lenders of last resort, which do not inherently exist in Bitcoin, and that people would be unwilling to accept bitcoin-substitutes in the market.

Those in the "Free Banking" wing of the Austrian school, such as George Selgin and Lawrence White, believe that bitcoin's strictly fixed-supply and lack of lenders of last resort do not technically prevent a competitive system of fractional-reserve banks and entities arising atop bitcoin, or in an economy where bitcoin is the defacto monetary standard.

It is clear that there is a chance that bitcoin can, at the very least, emerge as a mildly volatile digital commodity, a store of value akin to digital gold. However, doubts remain whether it will transcend the raw store of value role and achieve low enough volatility to become a global medium of exchange and a unit of account.

Some believe that, due to its strictly inelastic supply, bitcoin is unlikely to be stable in its purchasing power anytime soon, if ever, and that people prefer for their day-to-day currency to be stable in purchasing power. These people have expressed excitement about the emergence of cryptocurrencies with more flexible and self-regulating monetary policies built in. For example, stablecoins aim to peg their market value against another form of value, such as the USD or a basket of goods, using an algorithmic central bank.

Others believe that, despite bitcoin's strictly inelastic supply, bitcoin is a perfect solution to John Nash's Ideal Money proposal that he worked on for over fifty years. Nash, a Nobel Laureate in Economics, proposed that central banks could inflation-target their currencies against an apolitical index to achieve international relational stability of all state currencies. In response to increasing demand for bitcoin, some believe banks will value target their currencies against bitcoin as a basis for the standardization of the value of money.

Deflationary Death Spiral

Mainstream, Keynesian, and Monetarist economists have expressed concerns with Bitcoin's fixed-supply. They fear the possibility of harsh deflationary pressures if

bitcoin becomes the predominant currency through the process known as hyperbitcoinization.

Their fear is that the inability to expand the money supply would result in bitcoin's purchasing power growing by 2–3% per annum, roughly in line with the growth rates of global economic output. Some have expressed concerns that deflationary economics might reduce aggregate demand in the present and the near-term, result in excessive savings and hoarding of money, and produce less consumption, investment and entrepreneurial risk-taking by individuals.

Austrian economists believe that the fears associated with a deflationary form of money are overblown and that the 'deflationary spiral' is a myth. Austrian's counter the Keynesian and Monetarists concerns that the delay in spending doesn't last in perpetuity by reminding them that this spending is merely delayed into the future. People will now have a lower time-preference and that instead of buying "useless" things with their "hot potato" decaying money, they will turn their attention to long-term productivity.

They also believe that business profit margins will not be hurt because not only would product prices, but also business costs, deflate at the same rate, leaving the profit margins unchanged. Austrians believe that deflation is absolutely normal, and absent central control on the money supply, both capitalism and technology are naturally deflationary phenomenons. This can be seen in the less-regulated electronics industry, where increased storage/memory/compute capacities are becoming cheaper every year.

According to Austrians, it is the *central bank inflationary fiat printing* that exacerbates recessions and business cycles, as the perpetually-decaying money embeds the citizenry constant anxiety and stress, resulting in not well though-out investments and expenditures, collectively referred to as 'malinvestment'. These malinvestments are typically inefficient allocations of capital, which are unlikely to result in personal gains, societal gains, productivity, or capital stock.

First Theory: BTC as a tamper-proof store of value.

Tenets: Sound Money. Set-in-stone monetary policy. Full-node affordability. Sovereign-grade censorship-resistance. Maximized decentralization and security.

Individuals that fall within the Bitcoin-as-sound-money camp generally believe BTC is the only legitimate cryptoasset, with everything else ranging from being entirely useless at best to blatant scams at worst. Commonly called Bitcoin Maximalists,

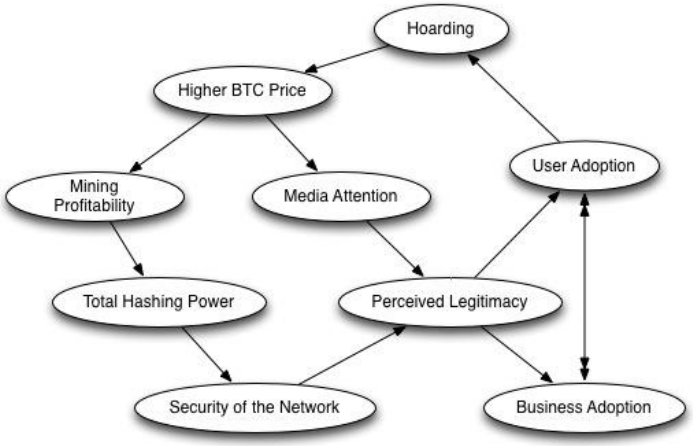
they desire "sound money" by the Austrian Economic School definition, that cannot be inflated away or be at risk of confiscation, as the case was in 1933 when Executive Order 6102, issued by Franklin D. Roosevelt, made hoarding gold illegal in the United States.

These individuals believe that, for the foreseeable future, the goal of the Bitcoin project isn't to facilitate the buying of coffee, but to become "high-powered" money, an even better form of gold. They claim it to be a digital asset superior to physical gold due to a truly limited supply and more deflationary emission. They also claim that, if used properly, it is unseizable, unhackable, arbitrarily unprintable, and is an attempt to engineer a superior form of money. Bitcoin is often discussed as a settlement network where the raw block space is not meant to facilitate small-value individual transactions. It is believed, rather, that its usage is for settling transactions of a larger value, where fees are less of an issue. This would likely include once-in-a-while settlement transactions of secondary payment solutions, for example, settling millions of Lightning Network payments in one finalizing transaction on the blockchain.

Although there are many brilliant concepts that were first brought together in Bitcoin, many Bitcoin Maximalists believe the mining difficulty adjustment may be the most ingenious, as it allows for true digital scarcity that is tied to the external, physical world. Saifedean Ammous is one of the most vocal and prominent BTC Maximalists, and in his new book, The Bitcoin Standard, he states that it is Bitcoin's high stock-to-flow ratio coupled with its untamperable monetary policy that will eventually make it both the most attractive and the most robust store of value.

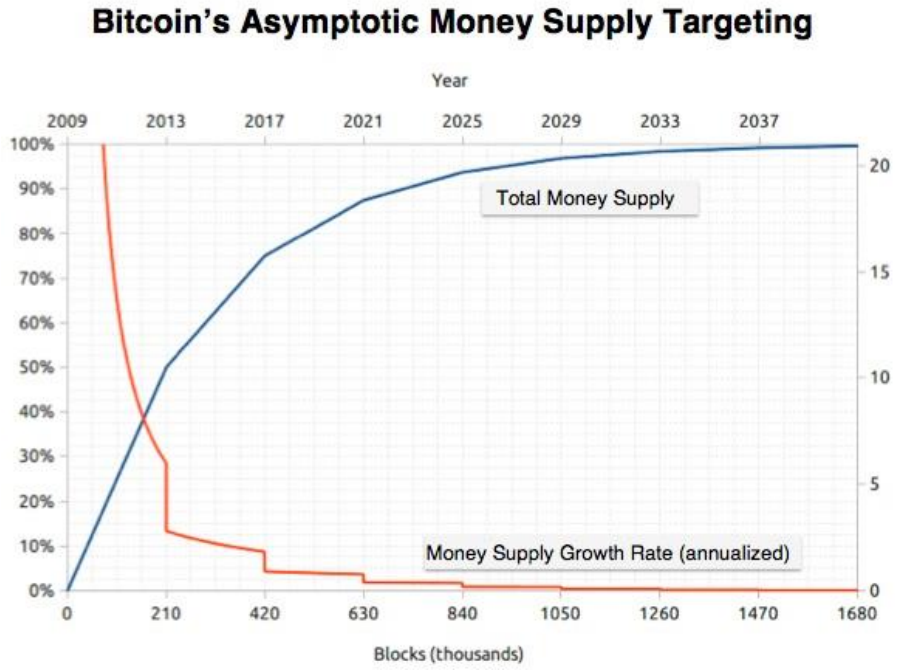
As of now, most bitcoin holders insist on not spending, a common statement being, "it would be foolish to spend when the price can still increase by a factor of 100x or more." Instead, many are hoarding the asset, which has become known colloquially as "hodling." For them, hodling is the main use case of Bitcoin during the time before the Tipping Point. The positive feedback loop of hodling and the price increasing encourages an ever-growing army of hodlers. This army in turn collectively increase both the value of the asset and the desirability to hoard it, as supply available on the market becomes increasingly scarcer.

This logic is nicely illustrated by Pierre Rochard in the diagram below to show that hoarding may create a positive feedback loop to increase the BTC price, resulting in increased mining profitability, hashing power, user adoption, and more.



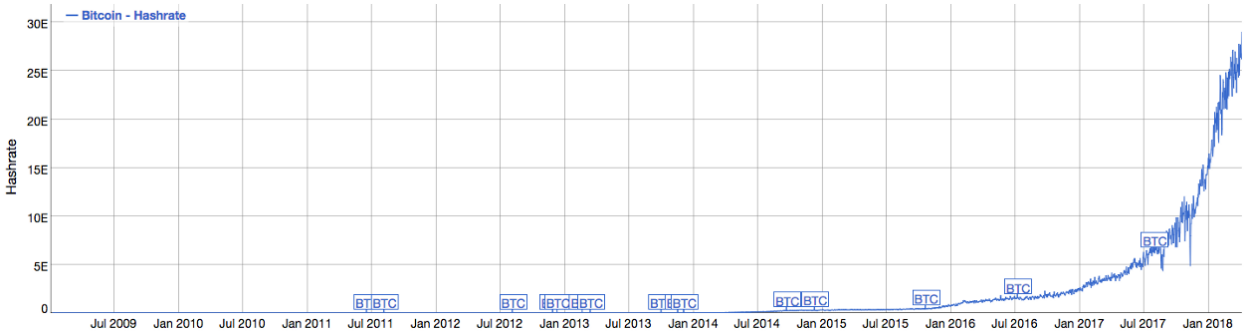
Bitcoin Market Components

Bitcoin Maximalists hold the opinion that the key element of Bitcoin is the money it represents, rather than technology behind it. They cite Bitcoin's "perfect monetary policy" (illustrated in the graph below) combined with the Lindy Effect due to first-mover advantage to explain why BTC will become and remain the dominant currency. The caveat being: Bitcoin must maintain its status as a peer-to-peer, decentralized system which produces a new block randomly, roughly every ten minutes. These Bitcoin Maximalists believe that as long as this is done, eventually hyperbitcoinization will occur, resulting in BTC being the dominant currency in existence.



In terms of monetary policy, Bitcoin Maximalists tend to believe that the hyper-deflationary total money supply of Bitcoin gives it the best monetary policy of any existing asset, and fractional reserve banking is rotten to the core. They prioritize saving and capital accumulation as opposed to superficial consumption. They believe, in line with Austrian school economists, that government meddling, especially with the money supply, causes malinvestment, makes interest rates artificially low, and enriches a select few at the expense of many.

Bitcoin Maximalists believe that bitcoin as sound money is to be accomplished through maximizing both the collective and individual security within the system. Currently, the Bitcoin blockchain is by far the most difficult to tamper with of any cryptocurrency in existence, based on hashrate alone. As of April 2018, it is estimated that transactions in Bitcoin are currently secured by confirmations from a network of computing power that produces over 29 exa-hashes per second. This rate is estimated based upon the mining difficulty, which has approximately tripled in the last six months and has grown every year since the release of Bitcoin in 2009.



Bitcoin Hashrate (as of April 2018)

Bitcoin Maximalists believe that the network hires miners to do one specific job: mine the blocks that full nodes determine to be valid. Thus, they believe users are in control of what Bitcoin validates, not miners. To facilitate this, Bitcoin Maximalists emphasize that users should attempt to be self-sovereign by controlling their own private keys and verifying their own transactions by running full nodes. By minimizing block size and data stored on-chain, users can still manage to run full nodes even on low-bandwidth connections. There are currently over 9000 reachable full nodes among over 100,000 total, which all store copies of the Bitcoin blockchain.

Bitcoin Core is the most dominant open source project that uses the Bitcoin protocol. The developers currently responsible for Bitcoin Core are primarily focused on supporting the Lightning Network and other payment channels. They also support CoinJoin for privacy, and are developing more support for side-chains as future second-layer or even third-layer solutions for payments emerge, MAST, and Schnorr signatures and signature aggregation in order to maximize how efficiently block space is used on the Bitcoin blockchain. They are also looking into

implementing confidential transactions, potentially using Blockstream's Elements Project. There is also a proposal to implement confidential transactions as a softfork, using segwit.

Second Theory: BCH as peer-to-peer Digital Cash

Tenets: Peer-to-peer, censorship-resistant, borderless cheap transfer of value, without middlemen. High on-chain throughput and on-chain utility. Set-in-stone monetary policy.

The members of the Bitcoin Cash community believe that Bitcoin should have unrestricted block sizes in order to facilitate peer-to-peer payments without bounds, and that Satoshi's original intention was to create a peer-to-peer electronic currency, as opposed to something like digital gold. They generally cite the title and abstract of the Bitcoin whitepaper as proof of their correctness, as well as Satoshi's statements regarding a phased-in approach to increase the block size, among other writings. These supporters generally believe that the Bitcoin system should not be a settlement layer solely for those who can afford to pay increasing fees, such as banks and other wealthy entities. They are completely against that use case for technological and ideological reasons, and they want to see most activity on-chain.

The implementation of Bitcoin that these individuals prefer is called Bitcoin Cash. It aims to gain adoption as a medium of exchange before becoming a store of value by keeping blocks large enough for nearly infinite transactions to take place. It aims to allow all users to transact on-chain, including those who may be underbanked or unbanked, some of whom earn as little as the equivalent of 2 dollars a day.

This Bitcoin fork to an alternate implementation was the result of increasingly differing opinions on the technological and social directions the Bitcoin community should head. One example is the disagreement between prominent developers on the implementation of a protocol change implemented in February of 2016, referred to as replace-by-fee. Bitcoin and Bitcoin Cash supporters clashed on this point of contention: some siding with Mike Hearn's replace-by-fee counterargument, and some in favor of 0-confirmation transactions to allow for instantaneous payments to maximize utility of Bitcoin, and allow it to essentially be used the same as cash. The side in favor of bitcoin as cash often cites Satoshi's vending machine example as reasoning for this always being desirable within the system. Still, others believe in researching alternative methods altogether instead of relying on controversial 0-confirmation transactions.

Bitcoin Cash supporters believes that a cryptocurrency can only become the dominant currency in existence if it is primarily used in a transactional capacity. Therefore, instead of encouraging hoarding within the community, they maintain that a certain percentage of an individual's bitcoin cash should be used for spending each month, and some encourage constantly replenishing the spent BCHs. By doing this, they hope to encourage the adoption of Bitcoin Cash as a payment system by incentivizing as many merchants as possible to accept the currency. This seems to be rooted in a desire to smash the nation-state monopoly on money and create a closed loop, with people earning Bitcoin Cash, spending Bitcoin Cash, and merchants paying suppliers and employees Bitcoin Cash.

Bitcoin Cash choose not to add segregated witness to their implementation and believe that full nodes that receive and validate transactions but do not mine are irrelevant to the base security of the system. Instead, they believe that hash power is the only thing that can determine the direction of Bitcoin. They believe that miners are the only true full nodes: serving as competing entities, forming a consensus state and generating new blocks. They believe it is normal for large mining farms to arise in such a system, and as supporting proof, they often cite Satoshi Nakamoto's server farm post.

In "Proof of Work as it relates to the theory of the firm," Bitcoin Cash supporters describe the system as a multi-leader-follower Stackelberg game where miners serve as rational actors controlling hash power. Under this type of Stackelberg game, miners are to be in constant, non-cooperative competition with each other to maximize profits by optimizing their efficiency in generating new blocks by handling their quantity of hashrate.

In the medium-term roadmap, Bitcoin Cash developers plan to re-enable certain scripts included in Bitcoin transactions, known as op-codes, which would allow for more utility with smart contracts. They want to launch tokenization on-chain over the next year as an upgraded version of Colored Coins, through a competition for a five million pound prize. The goal being to both increase merchant adoption of the cryptocurrency and consume alternative smart contract platform use cases by unlocking the full ability of scripts in Bitcoin. They also plan on launching on-chain privacy through Oblivious Transfers.

A paper investigating Bitcoin Cash as infrastructure for internet commerce discusses miners being divided into specific task groups without modifying the underlying Bitcoin protocol. For example, processing nodes could reference a limited subset of the blockchain, others could store the complete blockchain, others could be for monitoring, and still others for propagating information. The paper also introduced distributed autonomous corporations as systems that live on additional layers to allow for more efficient information propagation. These corporations could also be autonomously verified for integrity by third-parties. The paper goes on to describe

hypothetical fast payment networks which would operate through on-chain assurance contracts for merchants to pay for "preferentially propagated transactions," and be operated by distributed autonomous corporations. The paper also proposes that distributed autonomous corporations could be used for double-spend monitoring to allow merchant rejection of consumer payments within seconds, or in time for vending machine to stop from releasing an item.

Bitcoin Cash supporters view it as more than a simple payment system; for example, some view Bitcoin as a robust dual-stack pushdown automaton(2PDA) from the alt-stack and main-stack present in the Bitcoin scripting language. As discussed in this video from a Bitcoin Cash supporter conference, it is hypothetically possible that Bitcoin can operate as a Universal Turing Machine, which means Bitcoin would allow any computable functions to operate as a script executed on-chain. Some computations, such as cellular automata, would require multiple transactions.

In another Bitcoin Cash paper, a Bitcoin Cash supporter states that an unbounded single-tape turing is analogous to an unbounded blockchain, and can store a genetic algorithm that will be able to provide Turing Complete results on any given mathematical problem. Therefore, the paper posits, the eventual result of Bitcoin Cash will be to create the Church-Turing-Deutsch Principle Machine, as described by David Deutsch in his 1985 paper"Quantum theory, the Church-Turing principle and the universal quantum computer" which states "every physical process can be simulated by a universal computing device."

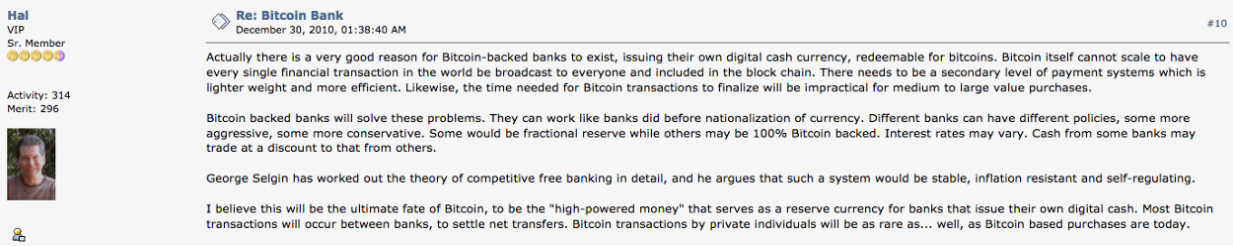
Third Theory: Bitcoin is a catalyst for Josh Nash's Ideal Money

Tenets: Apolitical store of value. Mining difficulty as solution to John Nash's theoretical Industrial Consumer Price Index. Idealized settlement layer between central banks issuing their own currencies.

A niche number of individuals, the most prominent of whom posts under the names Juice (Medium) and SoakerPatoshi (Twitter), generally agree with the Bitcoin Maximalist thesis that Bitcoin is likely to become the new modern Gold Standard, and that it is likely that it becomes a massive, global asset, trillion-dollar asset. However, they have a different view with respect to the endgame. This group

believes that even if bitcoin grows such that is surpasses the market cap of gold, nation-state backed fiat currencies will nevertheless remain. Instead of causing the collapse and disruption of that fiat money, Bitcoin will instead act as a catalyst to force central banks to manage their fiat currencies in a more responsible manner.

This possibility was initially thought of by Hal Finney, who is best known for being an early Bitcoin developer, being the first person to transact with Satoshi Nakamoto, and being a developer of the secure communication method known as Pretty Good Privacy. He posted on the bitcoin forums his thoughts about hypothetical Bitcoin banks in 2010.



Hal
VIP
Sr. Member
Activity: 314
Merit: 296

Re: Bitcoin Bank
December 30, 2010, 01:38:40 AM
#10

Actually there is a very good reason for Bitcoin-backed banks to exist, issuing their own digital cash currency, redeemable for bitcoins. Bitcoin itself cannot scale to have every single financial transaction in the world be broadcast to everyone and included in the block chain. There needs to be a secondary level of payment systems which is lighter weight and more efficient. Likewise, the time needed for Bitcoin transactions to finalize will be impractical for medium to large value purchases.

Bitcoin backed banks will solve these problems. They can work like banks did before nationalization of currency. Different banks can have different policies, some more aggressive, some more conservative. Some would be fractional reserve while others may be 100% Bitcoin backed. Interest rates may vary. Cash from some banks may trade at a discount to that from others.

George Selgin has worked out the theory of competitive free banking in detail, and he argues that such a system would be stable, inflation resistant and self-regulating.

I believe this will be the ultimate fate of Bitcoin, to be the "high-powered money" that serves as a reserve currency for banks that issue their own digital cash. Most Bitcoin transactions will occur between banks, to settle net transfers. Bitcoin transactions by private individuals will be as rare as... well, as Bitcoin based purchases are today.

Hal Finney on Bitcoin Banks

John Nash, a Nobel Prize economist who made significant contributions to game theory such as the Nash Equilibrium and Bargaining Problem, believed that although Keynesian economic policies were, in theory, intended to be for solely noble objectives and general welfare of the people, in practice these policies simply gave governments the ability to literally print money, collecting seigniorage by way of inflation of the money supply. Nash often likened Keynesians to Bolshevik Communists, as he saw that both groups gave credence to the notions of a centrally managed system and a lack of transparency surrounding decisions, especially with regards to the nation-state's currency issuance.

Nash wrote that by discussing inflation targeting, central banking officials are essentially revealing that is possible to control inflation by controlling the supply of money. Central banks, in their calculations, use a cost-of-living index made up of domestic prices for goods in a given region of their nation-state. Nash introduced a notion he called the Industrial Consumer Price Index, or ICPI, which would provide an international standard for value comparison of goods via a formula incorporating differing prices of goods in a variety of locations.

Nash believed that a return to the Gold Standard was sub-optimal, because he believed that technological changes resulted in increasing unpredictability of the future cost of gold production. He also considered the locations of gold mines to not be "politically appealing" nor ideal, and that a return to the Gold Standard would arbitrarily enhance the economic importance of those particular areas.

Nash's Ideal Money proposal, in a nutshell, is an idea that although we cannot design a perfectly stable money, a money that approaches 'stable' would also

approach a limit that would be comparable to an optimally chosen basket of commodity prices. While an ICPI would be a step on the path towards Nash's vision of an Ideal Money, pegging a currency to the ICPI is not a solution, as it could fluctuate with major technological breakthroughs, and the subsequent readjustment could also be prone to political pressures.

Currently, global reserve currencies face the Triffin Dilemma, resulting in a conflict of interest between short-term domestic and long-term international objectives, such as a desire to increase inflation to spur economic growth, versus keeping a strong domestic currency with stability of purchasing power. Nash believed that money would be stronger if it were put on a stage of competition where it must compete to survive, and improve itself. Nowadays, however, currencies don't really compete in a typical way like that which results in better products over time, but rather, they compete in a race to devalue. For Nash, rather than focusing on the utility of money for everyday transactions, of paramount importance was for the global economy to arrive at the same incorruptible value standard.

Bitcoin is seen by some as the catalyst for the evolution of global monetary systems toward something that would resemble in stability to an optimally chosen basket of commodity prices. Some believe that Nash's writings from 1960's may have even predicted the emergence of something like Bitcoin. Nash wrote: "Here I am thinking of a politically neutral form of a technological utility. To be quite respectable, in a Gresham-advised sense, money needs only to be as good as other material commodities that might be hoarded."

Coincidentally, over the last several years, a global consensus over the nature of Bitcoin has slowly been converging on phrases like "digital gold". Bitcoin has all the characteristics to acquire a global monetary premium, much the same as gold. The relation between scarcity and new supply is actually more important than scarcity of supply. In the next several years, bitcoin's stock-to-flow ratio, the relation of its scarcity to its new supply, will drop below that of gold. Bitcoin's annual inflation will continue to decrease. Many believe that during this time, bitcoin will draw growing interest as an inflation hedge from many around the world.

It is plausible that if Bitcoin continues seeing infrastructural improvements and growing place in the market, central banks and fiat currencies will themselves be forced to compete with Bitcoin in the future for relevance. It is likely that citizens of nation-state will put pressure on their central bank to print less money of a superior quality, resulting in a slower rate of inflation. If this were to occur, Bitcoin would likely usher similar effects to what an ICPI basket would achieve, without ever implementing an actual ICPI. This would force fiat currencies closer to Nash's vision of Ideal Money. Nash's goal is believed by some to be near, as Bitcoin represents competition to the nation-state control of money for the first time in centuries.

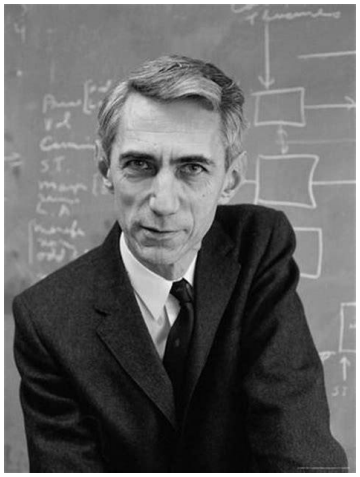
Fourth Theory: Bitcoin is an information and energy black hole that will result in the evolution of traditional money

Tenets: Perfect information markets and computational markets. Bitcoin is fractal and the sum of its forks. Peer-to-peer, censorship-resistant, borderless, cheap transfer of value, without middlemen. High on-chain throughput and on-chain utility.

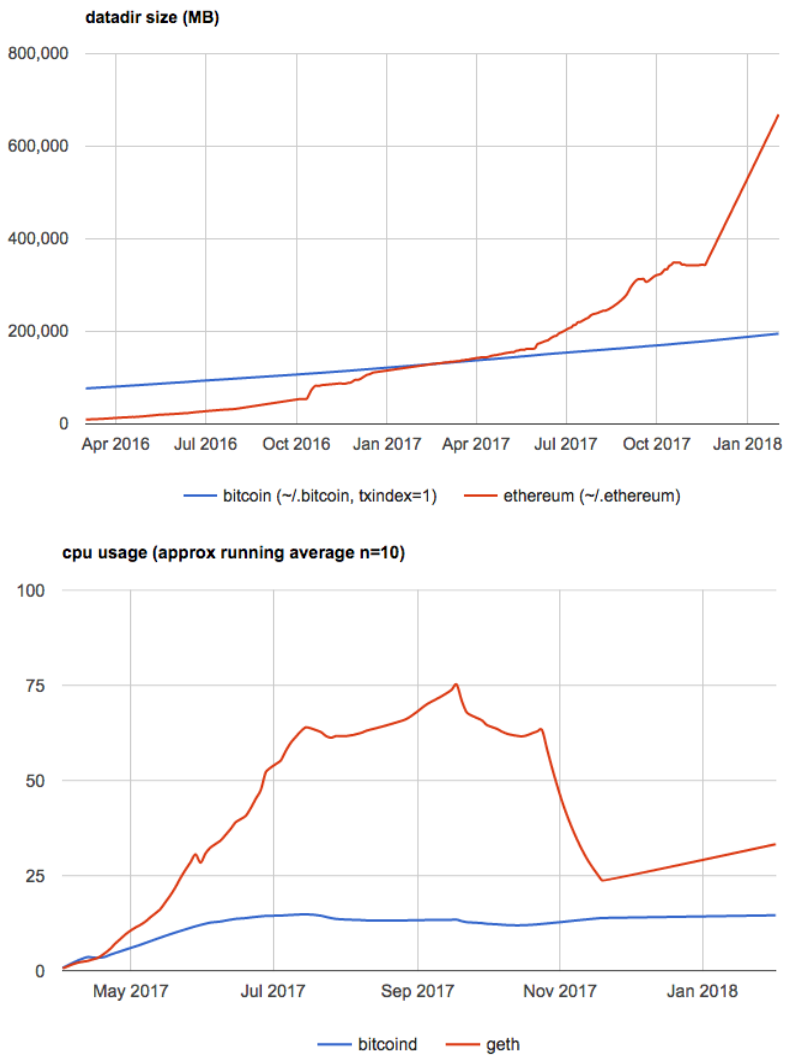
"Bitcoin isn't money. It's past money, which is scary because it's actually a new paradigm. We've never had access to perfect market information before, so the concept of money will have to evolve to fit reality, not stay the same because legacy deems it so." — anonymous

There is also a niche number of individuals, the most prominent of which are George Gilder, author of a number of books on the monetary system and capitalism, Andrew DeSantis, former engineer at the Bitcoin startup 21.co, now earn.com, and Mark Wilcox, Chief of Strategy at Nyriad, that discuss Bitcoin primarily as interpreted by Claude Shannon's information theory. In simplistic terms, information is defined as surprise under information theory.

Claude Shannon, the founder of information theory



This group believes Bitcoin is a breakthrough in information theory because it allows anyone to conduct verifiable, timestamped, tamper-proof and transparent transactions without any third parties. Information theory says creativity requires a stable medium to experience fractal growth, and these individuals view Bitcoin as an extremely stable medium for doing so. In Knowledge and Power, Gilder argues "it takes a low-entropy carrier (no surprises) to bear high-entropy information (full of surprisal)." This camp also agrees with Bitcoin's deflationary policy because they view capitalism and technological progress as a fundamentally deflationary system.



Similarly to Bitcoin Cash supporters, these individuals favor pushing Bitcoin to its limits to maximize the utility of an open data layer, and they are not fond of a future of Bitcoin where throughput is limited so that all users can verify their own transactions with a full node. This group views one use case of Bitcoin as an oracle machine to prove that a specific piece of data existed at a given point in time, and the bitcoin scripting language as much more dynamic than Ethereum due to the parallel nature of Bitcoin in comparison to the current serial execution forced by contracts on the Ethereum platform.

This group views Bitcoin as a platform to re-build computer software and the web upon. For example, they are interested in the parallels between Ted Nelson's Project Xanadu, the first hypertext project, and Bitcoin. Project Xanadu was envisioned to bring about a highly interconnected, parallel universe of documents for reading, writing, and learning through hypertext, "non-sequential writing — text that branches and allows choices to the reader, best read at an interactive screen"

that was to operate through worldwide distributed servers and facilitate micro-transactions across the web.

As discussed in "Blockchain Control Flow," Ethereum has made design decisions that allow the network to have control over contract execution, and thus users' money. Wilcox writes "For a peer-to-peer network to be politically decentralised, it needs to have decentralised control, so we should at all times try to keep control completely in the private section." He also writes that the "limitations" of Bitcoin as cited by Vitalik Buterin in the Ethereum Whitepaperare *protections* not limitations.

Individuals within this camp generally have negative opinions towards the Lightning Network and other secondary-layer solutions. DeSantis states the "Lightning Network makes the base chain strict, or predictable," and thus reduces Bitcoin's information theoretic value by constraining experimentation space and reducing the chances of surprise discovery. Wilcox views the Lightning Network "as a scam designed to function as an abstraction layer between you and the miners."

When Wilcox discusses transaction processing, he is referring to verifying a transaction and hashing it into the merkle tree. Transaction processing can refer to almost anything, and he proposed a thesis that the same economic incentives that allowed Bitcoin's hashrate to grow exponentially over the last nine years could be used to exponentially grow transaction processing, which is currently done serially on a CPU.

Nyriad, the company Wilcox co-founded, created the Nsulate for the Square Kilometre Array project, the world's largest radio telescope. The Nsulate innovatively uses the GPU as a storage controller and makes processing and storing data the same thing. It has built in blockchain support through cryptographic hash algorithms, which would allow miners to process transactions in parallel.

Many of Wilcox's arguments, therefore, are based on seeing Bitcoin as a platform to enable competitive general purpose computational markets where users and companies submit transaction puzzles via scripts for miners to compete to solve using GPUs and write-on chain to seek rewards. Transaction puzzles can mean nearly anything here, from deep learning to CRISPR searches. With computational markets in Bitcoin, a user looking to submit computations to miners will care a lot about efficiency, to get the most computation per unit of reward they are including in a puzzle, and hashrate, to ensure system they are submitting to is as secure as possible.

Wilcox and DeSantis typically argue against the traditional supply and demand outlook of economic markets for blockchains. Wilcox discusses the implications of Proof of Work for transaction processing and scalability in his blog, including Fundamental Misconceptions. Computational markets sitting atop Bitcoin are

particularly likely to expand if they prove to achieve cheaper and more efficient computation than established behemoths of the industry.

Source: Twitter

Conclusion

The four schools of thought presented in this article do not necessarily contradict one another, and, in fact, oftentimes overlap. In particular, the First Theory — Bitcoin being like a digital gold — and the Third Theory — Bitcoin leading to Nash's Ideal Money — run pretty much in parallel to one another, with the key difference being that the latter states that fiat currencies survive and adjust, while the former states that hyperbitcoinization will disrupt fiat currencies entirely, with everyone eventually demanding payment for their goods, services, and labor in bitcoin.

Similarly, the Second Theory — Bitcoin Cash being a dominant peer-to-peer digital cash — and the Fourth Theory — Bitcoin being the key element in further developing information theory — have many of the same supporting points and arguments, with the key difference being that the latter is not fork-biased and believes that any possible fork that can happen, will happen, and they will compete against each other.

Thanks to armor123123 *and others for giving us feedback on earlier versions of this post.*

Meditations on Fraud Proofs

By Paul Sztorc

Posted April 14, 2018

Toward a coveted O(log(n)) blockchain validation, for (?) ~ $50 a month. Plus: compensation for full nodes. Fraud proofs are a very complex, nasty business.

But if you would to learn my thoughts, sit here by the river and we may meditate together.

(River art by Benihime Morgan)

tl;dr

- Fraud Proofs allow "SPV nodes" to have similar security to "full nodes". SPV nodes are very easy to run and scale much better.
- Here, I require an "SPV+ mode". Whereas regular SPV requires headers only (~4 MB per year), this mode also requires the first and last txn of each block (~115 MB per year).
- "SPV+ nodes" must have a payment channel open with a full node, or a LN-connection to a full node.
- "SPV+ nodes" will have to make micropayments to these full nodes. To validate every single block, I estimate these payments to total ~ $50 a month.
- From there, it is just a few new OP Codes, one [off chain] rangeproof, and a second "SegWit"-like witness-commitment trick.

1. Background

A. Making Bitcoin More Like Physical Gold

Bitcoin is designed to rival gold. And in many ways, BTC is far superior, but one deficiency is when it comes to *receiving money*– how do you know you've been paid? With physical gold it is straightforward – as simple as any other hand-held exchange. But with Bitcoin (a *digital collectible*) your guarantee of ownership is much more abstract.

Satoshi's solution was a fancy piece of software, "Bitcoin", that tells you when you've received money.

But this is an infinite regress! What is the software doing?

Well, the software has a unique way of synchronizing with other instances. Kind of like Dropbox, but where your files would never have version control issues. It asserts its own synchronicity. "Knowing you have been paid" and "knowing you are synchronized" are the same thing!

Satoshi's whitepaper advocates two different ways of 'knowing you have been paid':

1. [Positive, Traditional] Run the software and wait until you are fully synchronized.
2. [Negative, Experimental] First, run a 'lite client' which would strategically synchronize some "easy parts" only. And then be on the lookout for 'alerts'.

The first is the so-called "full node", and it relies on *positive proof*– you are shown X, and when you see the X you known that you've been paid. The second is the so-called "SPV Mode"[1] and it additionally relies on *negative proof*– you are *not* shown Y, but if you were to see a Y, you would know that you have NOT been paid. This Y is called an "alert" in the whitepaper, but today it is known as a "fraud proof".

B. Theoretical Support for Alerts

To me, the most interesting aspect of the *negative proof* method (ie, 'alerts') is that it resembles *the way our world actually works* in most respects.

Consider these examples:

- We don't try to always make murders 100% impossible. But, if someone experiences a murder, we go to great lengths to catch the murderer (and establish their guilt in court, and punish him/her).
- We don't try to make "bad businessmen" 100% impossible. But, if there are incompetent businessmen, we expect that they will eventually go out of

business or be bought out, and thus replaced with someone more appropriate. If there is too much of a conflict of interest, we use tort law or regulation (to get rid of what we don't want).

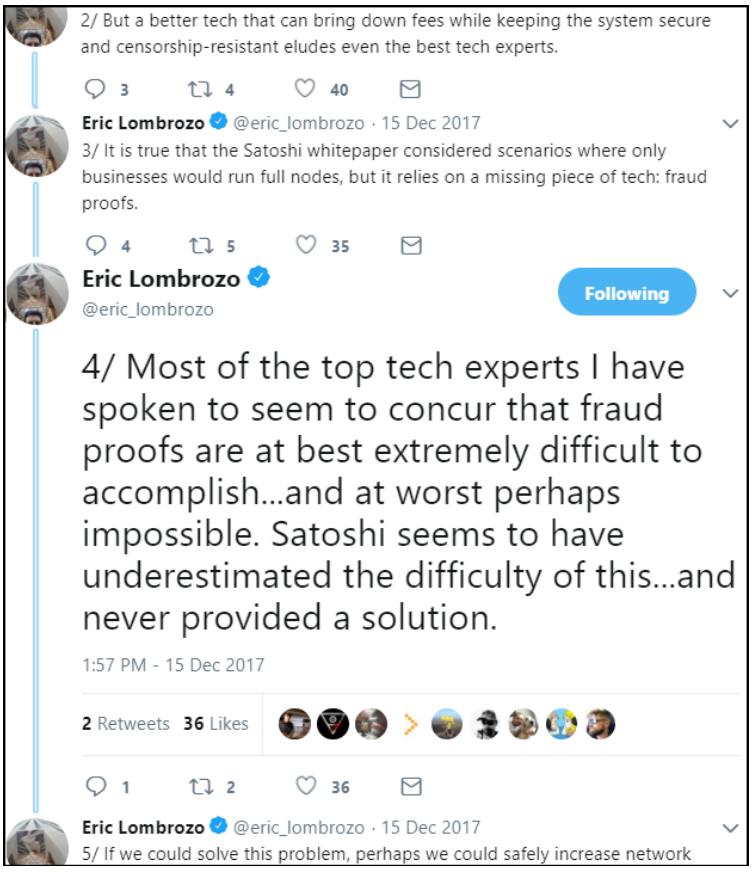
- We don't [even] publish scientific research as though it were 100% error-free. Instead, we make it maximally open to later criticism and future correction.
- We don't try to prevent judicial corruption 100%. But we do require all legal proceedings to be written down, and be audit-able by the public and by future legal scholars.
- We don't try to "know everything". But we expect to be able to "look it up" in a book/website, and we expect that specialists will correct those references over time.

In general, we have a kind of standing assumption that everything is satisfactory. And when severe-enough errors pop up, we act to correct them.

But, otherwise, in the real world, we find it far too difficult to 100% validate every single thing.

C. Theoretical Problems with Alerts

The problem with alerts is that Satoshi never actually implemented them, as you can see from this tweet from Eric Lombrozo last month.



Here are some of the outstanding puzzles:

1. **DoS Resistance:** Bitcoin full nodes are robust to DoS thanks to huge asymmetries in proof of work – namely, that it takes ~10 minutes to make a block but far less than 10 min to validate one. However, is this the case for alerts? Do they have PoW? Who pays for it? If not, what prevents a malicious agent from spamming innumerable false alerts at all times, making any true alerts un-discoverable?
2. **Proving a Negative:** a malicious/negligent miner can just discard parts of the block. But, even more extreme, a miner can actually mine a block without knowing *anything* about what is inside of it! If these "unknown" parts of the block contain bad txns, how can **we** ever learn about it? If no one can *learn* of them, how can we warn others?

The first problem asks us to find a spam-resistant signal that is not the blockchain itself – we will solve it using Payment Channels.

The second problem asks us to draw our [very scarce] "auditing attention" to specific parts of the block – we will do this by allowing people to claim that they do, in fact, know the entire block (all sections included), and by allowing those people to back up this claim with an audit.

2. The Problem

A. SPV Mode

Satoshi's SPV Mode (whitepaper section 8) observes that:

1. Bitcoin block headers are very small (4 MB per year) and easy to validate, regardless of how many txns each block contains.
2. It is very easy to demonstrate that a block *contains* some item "X" (as this requires only "X" itself, the block's header, and a valid Merkle Branch containing both).

For those who are unaware, consider this example:

```
Imagine that we have three Bitcoin headers: headerA, headerB, headerC.
Each header contains a hashMerkleRoot: hA, hB, hC, respectfully.

Is [Tx] in any of these blocks?  [header A] , [header B]

Yes, because h( [tx] ) = ht , and
     h(  ht, hs1 ) = hi1
     h( hi1, hs2 ) = hi2
     h( hi2, hs3 ) = hA

ht is the hash of the transaction [Tx].
hs1, hs2, hs3 are the Supplied Hashes, given by the Full Node ("Fred").
```

```
hi1, hi2 are intermediate hashes, calculated by SPV user ("Sally").

In practice, no one can link hA to anything other than h(hi2, hs3). In turn,
no one can link hi2 to anything other than h( hi1, hs2 ), ..., and no one can
link ht to anything other than [tx].
```

The Merkle Branch (the "supplied headers", as well as information about their order and position) is small and grows at the coveted log(n) rate. The payer can easily obtain/produce it, and they can easily send it to you while they are sending the transaction itself. So it is quite negligible.

Which means that this demonstration suggests that, in order to "know you've been paid", only the Bitcoin headers are needed. And the headers are very easy to obtain.

So SPV mode seems to suggest unlimited throughput, at tremendous ease.

B. The Problem with SPV Mode

The catch, is that you can never be sure if a given 80-byte header really *IS* "a Bitcoin header" or not.

The only way you can know *that*, is if you examine its contents in full. If a *single* invalid or double-spent transaction is hiding in there somewhere, the block will be invalid (or if the block is "flawed" in any other way, see "Block Flaws" below).

C. The Good News

While one cannot know if an 80-byte header is a Bitcoin header, we fortunately *can* validate the proof-of-work on a header. By simply hashing the header we can check it against the current difficulty requirement.

(And headers are so kind as to themselves contain a critical pair of information: the difficulty requirement itself, as well as the timestamp information – each can be used to check the other).

So, we can check that hashing effort has been expended, but not that the hashing was done *on* a worthy target. If you were buying a box of chocolates as a gift, and you had Matthew the Miner go to the store and purchase them for you, then our situation would resemble one where you could easily verify that Matt spent $300+ on the chocolates, but you can't know if the chocolates in question taste any good or even if they contain any actual chocolate.

D. Positive and Negative Proof, Revisited

You could eat all of the chocolate yourself, which would give you "positive proof" that each and every chocolate was delicious.

Or else you might rely on "negative proof", perhaps by reasoning as follows: "This box of chocolates is tamper-resistant, and it does not seem to be tampered with. And this product have a brand associate with it, and my country enforces laws protecting brands/trademarks. This brand is purchased by many, and so, if the chocolates were of unreliable quality, I would probably find a news story (or bad Yelp reviews, etc) if I looked. In fact, **I have looked for such a story but haven't found one**."

Another example is a money back guarantee. Imagine you are shopping for a car (an item of uncertain quality, just like our newly-found but unexamined Bitcoin block), and three competitors (call them "A", "B", and "C") each offer to sell one to you. Perhaps you are most interested in Car C.

Positive proof would be driving Car C for thousands of miles and having various mechanics check each piece of the car, and report back to you with problems.

Negative proof would be to observe that Competitors A and B each offer a *legally-binding money back guarantee* if their cars break down in the next 40,000 miles; but Competitor C **does not make such a guarantee**. This is negative proof that C is of low quality.

For Bitcoin Fraud Proofs, we need something that *always* shows up if the block is valid, but *never* shows up if it is invalid. (Or vice-versa.)

In game theory this is called a "separating equilibrium" in a "signaling game" (or more precisely a "screening game"), where the fraud-proof-senders are of two types, "Honest"-type and "Dishonest"-type, and we are trying to cheaply screen them for dishonesty.

E. Our Requirements

We need a way **to promote "block flaws" in our attention**. And ideally it must do so quickily (ie, "before transactions settle" ie "before 6 confirmations", or [to be safe] within 20 or 30 minutes).

Specifically, the following must happen:

1. "Sally" (SPV node) gets paid, in BTC, for something. Her counterparty shows her his txn, and she can see that her txn appears valid.
2. Sally wants to know that her txn has 6 confirmations, without running a full node. So she first downloads all Bitcoin headers, and second asks for a Merkle Branch that contains both [1] her txn and [2] a recent header. She gets one, but unfortunately for her [and little does she know]: the header is invalid for some reason. …
3. Simultaneously, "Fred" (Full node) must become aware that something is amiss. Specifically: that a block contains one or more "flaws" (see below).

4. Fred must have an incentive to provide some kind of warning (ie, the "alert").
5. In all other cases, Fred must have a disincentive to provide these warnings (ie, no "false warnings" … no warnings when there is actually nothing to worry about).

F. Classes of Block Flaw

A block can be flawed in many ways (see validation.ccp, especially "CheckBlock"). I have arranged them into four classes:

1. "Class I" – **Bad Txn** (invalid txn, doublespent txn, or repeat txn).
2. "Class II" – **Missing block data** (the Merkle Tree "neighbors" of Sally's txn are unknown and undiscoverable – this could be intentional or accidental).
3. "Class III" – **Bad Block (Other)** (misplaced coinbase, wrong version, witness data missing, [drivechain] most updates to Escrow_DB/Withdrawal_DB)
4. "Class IV" – **Bad Accumulation** (the infamous blocksize/SigOps limits, the coinbase txn fees [which must balance total fees paid by the block's txns], [drivechain] sidechain outputs – the "CTIP" field of "Escrow DB")

Class I

The first class is very straightforward. Sally can verify that a txn is invalid by simply trying to validate it and reversing the outcome (so that "false" validation returns "true"), more details below. In SPV mode, even nLockTime and CSV items can be checked, because Sally will have the Merkle Branch and all block headers. A doublespent txn can be checked even more easily, by just examining two txns and observing that they share an input. A repeat txn would fail the same test, as it necessarily would be a double-spending txn (unless it were a coinbase txn – see Class III Flaws).

Class II

The second class is of particular relevance to SPV users, because they must assume that the rest of the block exists, while [by definition] being prohibited from examining any of it. To make matters worse, miners can (and do) generate new blocks without checking the block contents. So it is possible the new blocks will have content that *no one* knows. Thus the assumption seems unjustifiable.

I will show that, conditional on a valid "header + Merkle Branch" being shown to Sally, a "full Merkle Tree" [ie, one containing Sally's txn, as well as a known finite number of other valid Bitcoin txns] exists in theory, even if not in practice. Therefore, all flaws involving missing blockchain data (all "Class II" block flaws) are "problems of a missing Merkle Tree neighbor". Therefore, they are **problems of an unknown**

hash preimage (much more tractable). More specifically, they are **problems of sampling** unknown hash preimages.

My solution to this problem will require Sally to obtain the last transaction (plus Merkle Branch) of each block[2].

Class III

The third class is quite general, but I believe that each trip-up can be solved with a simple trick that is specific to it. For example, the block "version" can be obtained from the header, which is already SPV-mandatory.

Most other information can be obtained from **coinbase txns**, so SPV users might be required to have all coinbase txns (in addition to all headers). From these, they can learn: that the coinbase appears only once and in the correct spot; that the witness commitment exists, and what it is; and that all of Withdrawal_DB is correct, as is most of Escrow_DB.

For one section of drivechain's Escrow_DB, mainchain[3] SPV nodes must become aware of the cumulative effect of the block's inter-chain txns. This will be handled as a Class IV flaw (section 7).

So we need to add some overhead – making a kind of "SPV Plus" mode (or "Surround SPV"). Instead of merely needing Bitcoin *headers* (80 bytes per 10 minutes), **"SPV+ nodes" also need the first and last txn of each block, and a Merkle Branch for each**.

- Old [Satoshi's Classical SPV]: 80 bytes, per block in the blockchain ; + one (txn, Merkle Branch) combo per txn received.
- New [This "SPV Plus"]: 80 bytes + (coinbase txn + last txn) + two [non-overlapping] Merkle Branches, per block in the blockchain ; + one (txn, Merkle Branch) combo per txn received ; + channels open with a few Nodes.

How many extra bytes? Well, we can't know for sure, but if coinbase txns average 1000 bytes, and 'last txns' average 280 bytes, and each block contains about 5000 txns, then the overhead would rise to 2192 bytes per block[4], instead of a mere 80. And the overhead grows at O(log(n)) instead of O(1).

At 52,596 blocks per year, the annual overhead would be ~ 115 MB instead of ~4 MB. This is a big relative increase, but small absolute one. Furthermore, Sally only needs to download this extra data for the blocks which she wants to *fully check* for validity: this could be the last 6 months of blocks, or all of the blocks in which she receives BTC (and the ~10 blocks surrounding it), and/or perhaps some random audits strewn here and there.

Class IV

The fourth class is especially interesting. In Section 7, I will describe how we may turn Class IV flaws into Class I flaws. In short, I will force each transaction hash to commit not only to itself, but also to its *contribution to all cumulative metrics*. For example, a txn will not only commit to "being 277 bytes" but it will also commit to "increasing the size of its host block from exactly 500,809 bytes to 501,086 bytes". Then, any problematic "liar txns" can be singled out and identified. It also means that the last txn will reveal valuable information (the total number of txns, the total size of the block, total txn fees paid, etc).

But before I give more technical details, I am worried about losing people in the weeds. So I will now present the "big picture", in the form of a story.

3. A Story

This story will star "Fred", a full node; and "Sally", an SPV node.

For simplicity, the story focuses on Class I and II flaws. (Class III flaws should be checked by downloading additional data, see above, and Class IV flaws will be recast as Class I in section 7.)

I

- Fred: "Here's that transaction you wanted. Wow, it says '300 BTC to Sally'. Is that you?"
- Sally: "Yes. I'm selling a SpaceShip to Peter Thiel so he can visit Jupiter.""
- Fred: "Cool, that sounds normal. Ok here is your Merkle Branch and here are all of the recent headers."
- Sally: "Yes, I can easily check the Merkle Branch by taking a few hashes, and it is also easy for me to check that the headers all meet the difficulty requirement. Wow. Praise Satoshi."
- F: "Praise Satoshi."
- S: "But how do I know this header is valid? Maybe the miners are misbehaving, or slacking off? Peter Todd told me that SPV sucks and stuff."
- F: "Ah, well you may be interested in some of my new services."
- S: "Oh? What are they?"

II

- F: "The first is called 'Invalidity Insurance', and you pay me $ 0.007 , but if you later find that an invalid [or double-spent] txn was included in this block [identified by hashMerkleRoot], you submit proof of this to the *real* blockchain, and I will owe you $1000 over there."
- S: "And any flaw will do?"

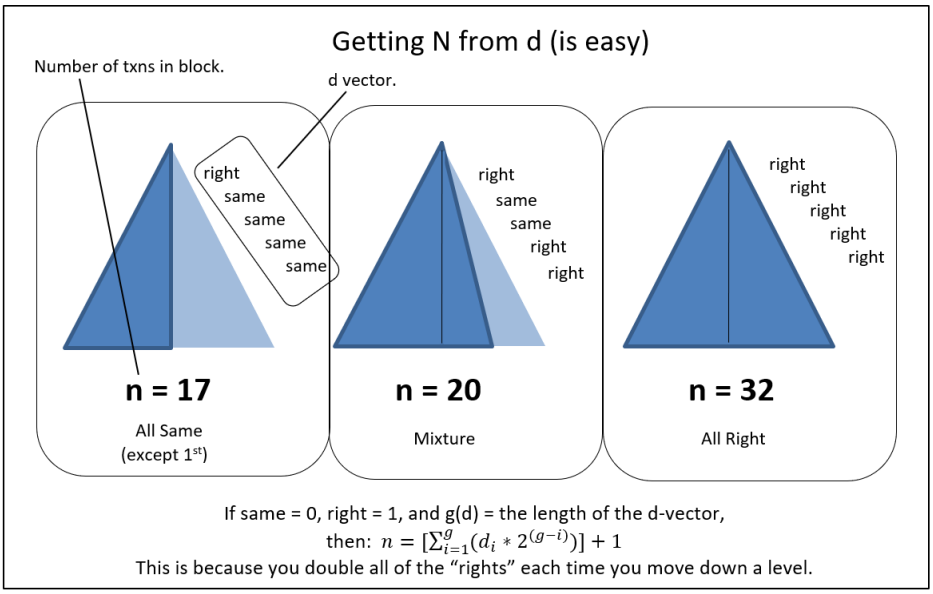
- F: "Yes, any type of evidence that the block is invalid."
- S: "Wow, you wouldn't do that unless you were certain that it wouldn't happen."
- F: "Yes, my computer has checked this block, and all of its txns, for invalidity. They are all valid."
- S: "Interesting."
- F: "But if you haven't found any flaws in 12 hours [72 blocks], the insurance expires."
- S: "I see. But this block should have fully propagated to the entire network of full nodes within 10 minutes."
- F: "Yes, and 12 hours is longer than 10 minutes by a factor of 72."
- S: "Well, as long as there is an *incentive* for fraud-laden information to propagate, it should definitely become available to the public within 12 hours."
- F: "Yes, incentives are the crucial thing."

III

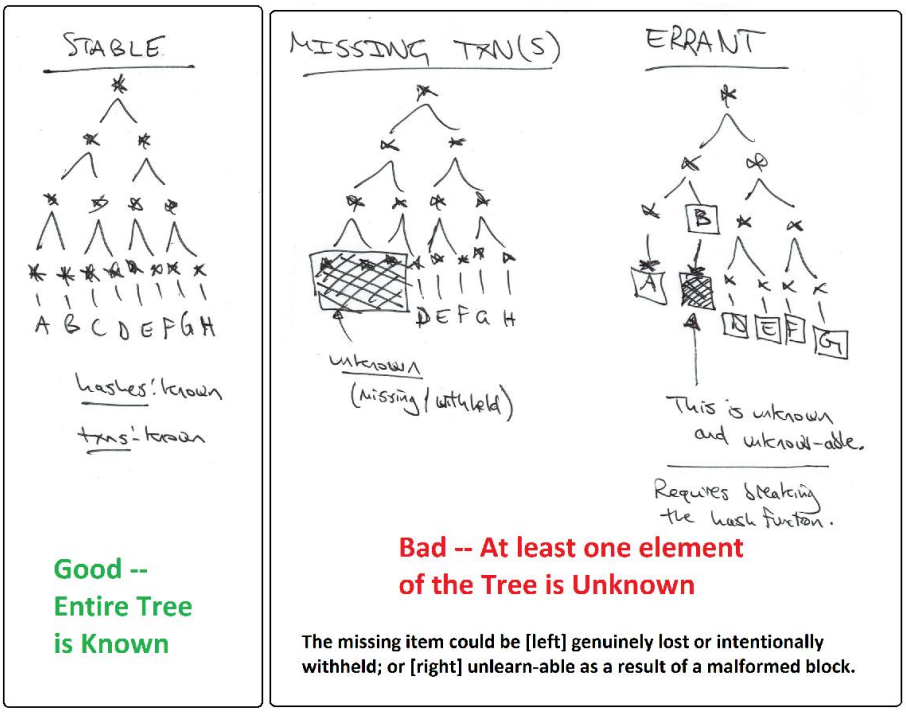
- S: "Wait, maybe part of the block is *missing*. Sometimes, I've heard, miners will mine on a block without even knowing what's inside it! What happens if they never learn? How will we know what's there!?"
- F: "Well, I actually do have the entire block, its all here."
- S: "Really?"
- F: "I definitely do."
- S: "Can you prove it?"
- F: "Yes. In fact it is my second new service offering."
- S: "Cool."
- F: "First: here's the last txn in this block. You can tell because we only went 'right' down the Merkle Tree, never left; or else we hashed something twice (which indicates that this level had an odd number of items). And you can compare this Merkle Branch to the one I just gave you for your transaction. They have the same root."
- S: "Yes, you seem to have indeed given me last transaction, of the Merkle Tree that my txn is in."

IV

- F: "The tree is eleven items deep, so you know that there are at most 2^11 = 2048 txns in this block. And you know how many times you hashed something with itself, and when, so you know what the Tree looks like. In particular, you know the exact length of its base."
- S: "Wow, I guess I knew more than I thought! Do I really know all of that?"
- F: "Yes."



- F: "All items need to be the same *depth* into the Merkle Tree. Or else there will be a mismatch – a situation where one is taxed with finding X that, when hashed twice, produces a value that is equal to hashing "a real txn" once. In other words, an 'uneven Merkle tree' would require its maker to find X such that h(h(X)) = h(transaction). But this is a 'hash collision' (considered impossible) – such an X cannot be found."



Fun fact: something cannot be both a Sha256 hash and also be a valid Bitcoin txn. For starters, the minimum Bitcoin txn size is 60 bytes (too big to be a 32-byte hash).

- S: "I think I understand – a Merkle Tree is geometrically similar to an Isosceles Triangle when there are no duplicates, and it approaches a Right Triangle when the last transaction is duplicated many times. The depth of the final transaction is depth of the triangle itself. And because of the Repeat Rule, if I just know the right edge, I know how many of the base elements (elements at the base of the triangle) are duplicates."
- F: "You do indeed! This block, the one with your 300 BTC txn, has a Merkle Tree that is eleven units deep – ie a triangle that is 11 units tall. And, from the final txn (I just sent you), you know that you only double-hashed once, and only at the very end. So you know that this Merkle tree contains exactly 2047 elements."
- S: "Ah, very clever… And if, instead, the depth were ten units, and all were self-hashed (except the first of course), I would know that there were only 513 unique final elements."
- F: "Exactly."
- S: "Wow."
- F: "And what if I gave you a new tree, and a new "last txn" with a Merkle path: [A, (self), B, C, (self)] ?"
- S: "It is five units deep, and has 23 unique elements in its base."
- F:"Precisely correct."
- S: "I've learned so much!"
- F: "Now, one last thing: each of these final elements is either known or unknown. And, if known, it must either be a valid txn or NOT be one. So each final element must either be: [1] a valid txn, [2] an invalid txn, or [3] a piece of information that no one can discover."
- S: "That seems straightforward."

V

- F: "Great. So, you can see that your block contains L=2047 txns."
- S: "It seems to."
- F: "Now here is my second service…I charge even less, just $ 0.0001 for this one! You pick a bunch[5] of integers[5] randomly, between 1 and L, and I'll give you their txns and paths. If I can't do as promised, I'll give you a boatload of money."
- S: "You seem really confident that you can do it."
- F: "I sure can! You can coordinate with your friends to pick the specific txns that you think I'm missing. I promise I've got them all!"
- S: "Wow, cool. I'll take it."
- F: "One thing, though. If we sign our deal and you don't reveal your integers[6], it will look like I can't meet the challenge. I mean, I totally can, but I don't know which txns to reveal because you didn't tell me. So if you don't hand them over in a timely fashion, I will need to be reimbursed in full, and then some."

- S: "Eh, OK I guess. I guess for my 300 BTC txn, I really shouldn't be stingy about locking up a much smaller amount."
- F: "You only need to lock it up for a few seconds. Believe me, I don't want my money trapped pointlessly in this channel for any length of time either."
- S: "Ok!"
- F: "Great. In our payment channel, we are currently at pair (A,B). You will need to make pair (C,D), and then reveal the random integers after we move there."

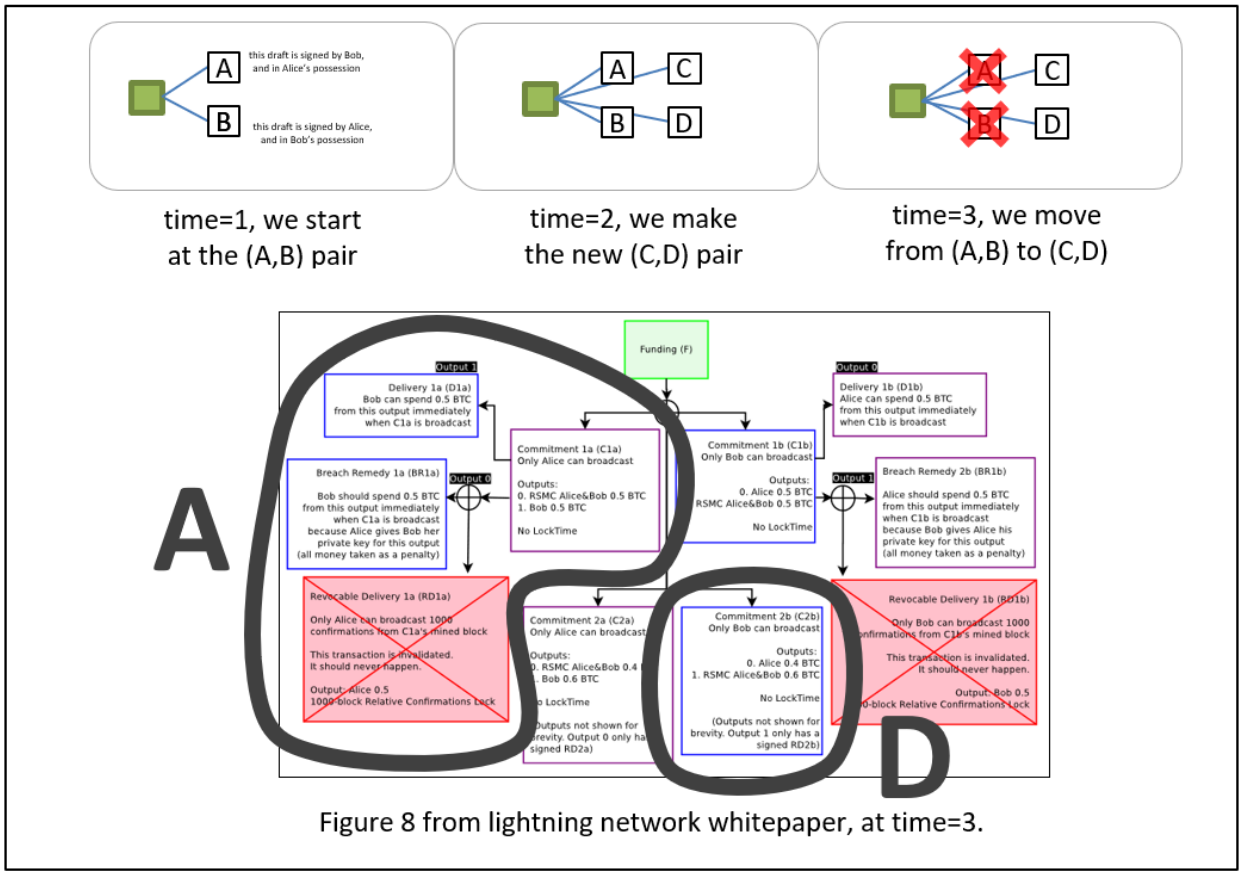


Figure 8 from lightning network whitepaper, at time=3.

VI

- S: "OK. I picked some Rs [random number]s, and made C and D [the next payment channel iteration]. And I signed D for you, so here you go."
- F: "Great thanks. Hmm…good job, this looks like you formatted everything correctly."
- S: "And here are those Rs, the random integers I cho–"
- F: "No! No no no, not until later!"
- S: "Oh, sorry!"
- F: "That's OK."
- S: "But if I only show you the Hs, ie H(R) the hash commitment of R, then how will you know that I am actually following our scheme? Maybe I didn't pick integers in range(1,L)? Maybe, instead of choosing numbers like (5, 470, 4, …), I

instead picked random nonsense like ('fish', 0x78965, '_', 987987987, …). Then, when I reveal my nonsense, you will not be able to show me the 'fish' txn…"
- F: "Ah…great question. There are professional cryptographers who have all kinds of ways of doing that. We will choose one and you will send me 'Gs' instead of Rs."
- S: "Ok, I've used the Rs to make Gs, here you go."
- F: "Ah yes, from these Gs I see that your Hs do in fact refer to integers in the range(1,L). Since I know all L txns in this block, I'm confident I can meet the challenge."
- S: "Great, so are we moving forward?"

VII

- F: "Yes. Here is your signed C. And I've invalidated my B [per the rules of payment channel iteration]."
- S: "Ok…don't you need me to invalidate my A?"
- F: "Yes, but if you don't, I can just broadcast my D…"
- S: "What if I broadcast my A first?"
- F: "It will be as if this transaction never took place…"
- S: "Aha! But I already *know* that you are willing to agree to do this! So you already *do* know that the block is valid– "
- F: (raises his eyebrows dramatically) "Do I?"
- S: "–which means I got what I wanted and now I don't have to pay."
- F: "Well, you know that I've agreed to do the challenge, but not that I actually *can* do it."
- S: "…"
- F: "And you knew, before starting this process, that I was *offering* to accept the challenge. So you haven't really learned that much more. I don't see why you would decide to back out now."
- S: "…"
- F: "Maybe I *knew* you'd try to back out. And so I offered to sell you an Audit, but really I don't know anything about the block's validity."
- S: "…"
- F: " *You know*, one one-hundredth of a cent really is quite small, compared to the cost of a whole Spaceship."
- S: "Fine…I'll invalidate my A. Here you go."

VIII

- F: "Ok, now you need to reveal your Rs"
- S: "Great. They were 453, 531, 14, and 2011."
- F: "Oooh good ones! .. Ok here are txns #453, #531, #14, and #2011; and here are their Merkle Branches. You can see that I had them all."
- S: "Wow, great. This is so cool. I love Fraud Proofs."

And finally, before I explain Invalidity Insurance and Fullness Audits, I offer a review of Payment Channels and the Lightning Network.

4. Payment Channels – An Overview

A. Regular (Non-Channel) Payments

With regular (non-channel) payments, the money is transfered when a "message" containing it is included in the blockchain.

It is as if Friend A emails you saying that you now have 4 of their 12 BTC. And then you click 'reply all' and say that Friend Q now owns 2.7 of that 4 BTC. If 'the blockchain' contains a copy of these "emails", then the txns are said to have happened.

B. Payment Channels

In payment channels, two people first "fund" (or "create" or "open") a channel. They do this by activating some of their BTC – taking it and paying it back to themselves. I take 90 of my BTC and pay it back, while you also take 15 of your BTC and pay it to yourself. We do this together in one txn, and it is broadcast and included just like a regular txn.

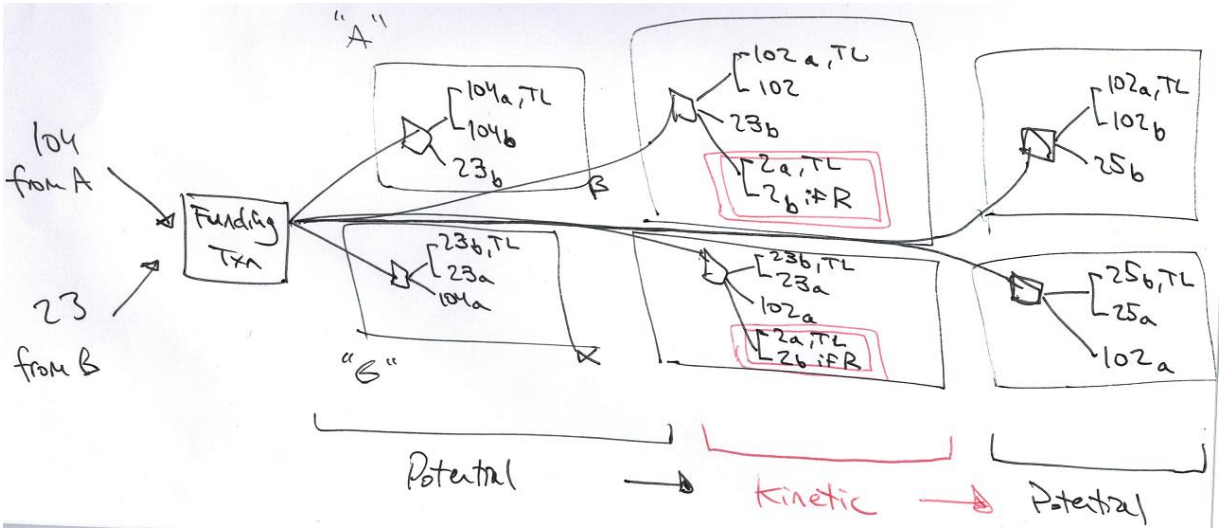
After the channel has been opened, however, the parties do things differently.

While before we actually sent "emails", with channels we instead keep multiple "drafts [of emails]", that we do not send. In fact, there will always be two parallel versions of the same draft – I will keep the version which is more convenient for you, and you will keep the version which is more convenient for me. The draft in my possession has not been signed by me, but it has been signed by you. The draft in your possession has not been signed by you, but it has been signed by me.

C. Updating the Channel

You and I will update these drafts together, by cycling the parallel pair of drafts through alternating states, that I call "potential" and "kinetic".

Channels mostly remain at rest, unchanging, in the Potential state. But occasionally users will want to alter the channel, and when they do this they will move the channel to a Kinetic state, and then quickly return it to a new Potential state.



Above: The state is "kinetic" – the 2 BTC amount (pink, double-boxed) is up for grabs, but [in practice] only very temporarily. The channel will spend most of its time in "potential" states, which reflect the most recent BTC balances.

As I said above, regular transactions are said to have happened, if 'the blockchain' contains a copy of your "email".

But payment channels are very different – the txn is said to have happened, if we have jointly "moved" from one pair of drafts to a new one. And this moving process is itself quite different – it does not occur when we jointly build, sign, and exchange the new pair; we must take an additional step of building, signing, and exchanging a pair of messages that torpedoes the old pair. Only then is the payment said to have "happened".

D. Hash-Locked Contracts / "Lightning Network"

The commonly-discussed kinetic state is called a "hash locked contract". When part of the kinetic state is shared by an interconnected line of channels (ie, a "circuit" of people), it can be interpreted as "the Lightning Network" which is explained as follows:

1. A "customer" wants to pay 10 BTC to a "merchant".
2. The **customer** makes a secret random number "R", and a public version of R called "H". "H" is simply the hash of R (or "h(R)").
3. Customer finds a "line" of connected people between himself and the **merchant**. Customer gives H to everyone in this line[Z].
4. Customer offers a kinetic update to "Friend_1", the the person standing next to him in line. Specifically, customer will pay Friend_1 10.0004 BTC if Friend_1 can guess the R.

5. Friend_1 knows that he cannot guess the R, but he knows that it will soon be revealed, and has nothing to lose, so he greedily accepts. The [Customer, Friend_1] channel moves from "potential" to "kinetic".
6. "Friend_1" repeats the process with "Friend_2", who is one step closer down the line from "Customer" to "Merchant". Friend_1 will pay Friend_2 10.0003 BTC if Friend_2 can guess R.
7. The process repeats, with Friend_3 getting 10.0002 for R, Friend_4 getting 10.0001, and Merchant finally getting 10.0000 from Friend_4 if Merchant can guess R.
8. With the circuit now complete, Customer passes R to Merchant, who *could* now use R to claim 10 BTC from Friend_4.
9. Merchant releases good to customer.
10. Merchant and Friend_4 do not want to pay a mining fee, nor do they want to wait for txn confirmations. So instead, Merchant waves the R in front of Friend_4, and uses it to negotiate a new channel state. It has moved from kinetic back to potential, but in this new potential state Merchant is 10 BTC richer and Friend_4 is 10 BTC poorer.
11. Step #10 is repeated for the other pairs: [Friend_4, Friend_3], [Friend_3, Friend_2], [Friend_2, Friend_1], and [Friend_1, customer].

E. Flexibility

The cool thing is that kinetic txns are not limited to HTLCs (ie, "pay if R is revealed"). A LN-txn can do *anything* that the blockchain beneath it can do.

We will first get the blockchain to do two new things: "invalidity insurance" and "fullness audits". After the underlying blockchain can support these "smart contracts", we can use them in channels.

F. How Channels Help

Channels are useful for fraud proofs, because they:

1. Operate meaningfully at near-instant speeds, and so have a chance of "pumping enough info" during our critical 20-30 minutes.
2. Facilitate micro-transactions (payments of very very small amounts).
3. Are robust to temporary mining failures (as they use long "custodial periods").
4. Can *prove negatives*, by: [1] asking for something, [2] offering a tiny reward for it, and [3] giving the counterparty a long time to provide it. If they don't meet the challenge, it is very likely that it is because they *can't*.

Now I can explain "invalidity insurance" and "fullness audits".

(River art by Max Pixel)

5. Invalidity Insurance

The first solution is the simpler of the two.

(Well, I'll let you be the judge of that, I guess.)

A. Overview

It is a payment (ie, a Bitcoin script) which is only valid if a given block (defined by its hashMerkleRoot)[8] can be proven to contain a Class I Flaw.

The recipient would supply four items:

1. **hashMerkleRoot** itself (abbreviated "hMR").
2. A **Bad Txn** – The TxID of a txn that is invalid for some reason.
3. A **Merkle branch** (as described here, consisting of: branch_length, branch_hashes, and branch_directions) containing both.
4. **Evidence** that the txn is bad.

B. Types of Evidence

The fourth item (the "evidence") would vary based on the type of Class I flaw – (a) invalid txn, (b) doublespent txn, (c) repeat txn; and Class IV flaws, (d) 'bad accumulator' (see section 7).

Aside: Fraud Proofs vs Smart Contract Fraud Proofs

Before I continue, I'd like to contrast "a fraud proof existing" with "a fraud proof that can be used in a Bitcoin txn".

I am aiming for the latter, because it is really a double-win: first, it gets us the fraud proofs themselves; and second, it gets us an incentive-compatible way of cheaply buying these fraud proofs from full nodes.

In fact, I really *need* the double-win, because I need to solve not only the problem of "how to make fraud proofs", but also *the free-rider problem* of "ensuring that they are supplied".

The benefit of making Bitcoin itself understand its own fraud proofs is that we can put them in transactions, and use them in Bitcoin smart contracts. But this is very difficult to accomplish. I don't pretend to have the best answers, at all[9].

(a) Invalid Txns

At first glance, one would think that we would only need to run a txn through the protocol's tx-validation function, to ensure that it *fails*[10]. And so our "evidence" would just be the txn itself.

But I'm afraid it isn't that simple. There is a hangup!

Each Bitcoin Merkle tree contains supposedly-valid transactions, as we know. But (!) it also contains its own interior nodes! What do we do if Sally tries to claim that a "transaction" is invalid, and then shows us a Merkle path to "two 32-byte SHA256 hashes strung together"? How do we know that these bytes represent an invalid Bitcoin txn, when instead Sally may have stopped halfway along the Branch? She may be lying about the depth of the Merkle tree!

And what of the reverse case (if Evil Fred deliberately constructs an invalid transaction, such that it takes the form of 64 random bytes)? Who's scamming who?!

To address this hangup, we require Sally to also provide a Merkle Branch to some other transaction that *is* valid. For maximum convenience, this might be her own transaction (the one she was given when she received money, at the very beginning of this process), or else it could be the coinbase txn (which she always has in "SPV+ mode").

Thus, in this scheme, proving that a txn is invalid actually requires two transactions total – the bad txn itself, and some "evidence" in the form of a good transaction nearby with an equal-length Merkle Branch. And so (a) is actually quite similar to (b) and (c) [the next section].

(b, c) Double-spends and Duplicates

For (b) and (c), the evidence of fraud my *also* initially seem straightforward. We just need two txns, and two Branches – then we see that the txns share an input (or a hash).

But I'm afraid there is a hangup for this one as well. It is the location of the second txn (by "second" I mean the chronologically "first" txn, aka the "real" output that someone is trying to illegally double-spend). This txn might be in a different block.

In fact, it might be several hundred thousand blocks away.

If the script interpreter can access past headers, then we may be able to do our work, with just a single additional 32-byte hash. I don't know the best way to do this – perhaps a new OP Code which is designed to fail if the next 32 bytes are NOT part of a known set of past hashMerkleRoots.

At worst, it would mean something rather cumbersome – appending an entire *sequence* of headers, from the header that we have backwards to the header of the block that contained the "real" spend of this output. From there, of course, it is a simple matter of including a second trio of [header, Merkle Branch, and txn], indicating a *valid* txn that already spent this output. (Or, for (c), a valid txn that has the same hash.)

Horribly, if the double-spend is of a coin spent many *years* earlier, we will need hundreds of thousands of 80-byte block headers just to get to it. This chain of headers could itself be 16 MB in size (!), making this transaction too large to ever include. Obviously, an area for improvement. In fact: a few people, including Blockstream, have published work (see Appendix B) on shrinking down one million 80-byte headers into something that is merely "tens of kilobytes" or so.

If you know the best way to handle this, leave a comment below!

(d) Bad Accumulator

For (d), Bitcoin needs a way to understand the data in txn's "accumulator" (or "second witness"). And it needs to check this data against the txns properties. Details to follow in section 7.

(And, again: very easy for software to do, but much harder to get it into a Bitcoin script.)

C. Putting it Together

With the script built, we can now deploy it strategically, in channels, to insure against the invalidity of a given block header.

Specifically, for each header, we will send a channel into its 'kinetic' phase, such that:

1. Sally is paying Fred a tiny amount more, unconditionally, and
2. Sally *may* be able to extract a lot of money from Fred, if she can obtain evidence that the header in question is invalid. Else, after some amount of time (1000 blocks or whatever), the money returns to Fred.

The first is kind of like an **insurance premium**, and the second is like an **insurance claim**– money to be claimed by Sally if the conditions are met, but otherwise reclaimed by Fred if they are not.

An honest Fred has *every* reason to sign txns of this kind. To him, the evidence in question will never be obtained, so the claims will never be paid, and the premium payment is just free money.

Thus the "fraud proof" here is actually in reverse – when these types of txns *stop being offered*, we are alerted to Fraud.

However, under one condition, a dishonest Fred might still offer the service. We mentioned it in the story – it is: if the txn data is missing. A Dishonest Fred knows that, if the data is missing, Sally will never be able to use it to demonstrate Fraud. And so Fred will never be "caught" – the evidence of his guilt does not exist anywhere, so Sally can never come into possession of it.

That is why we need the Fullness Audits [next section].

6. Fullness Audits

Fullness Audits allow Fred to demonstrate [to Sally] that he actually has every transaction in a block.

A. Two Ingredients

First, we need a new "weird ingredient" for this one: something that transforms "an integer" into "Merkle path directions", and back.

The funny thing is, though, we actually don't need to do anything for that. The directions are already stored (for Namecoin merged mining, anyway) as a single int32_t. It even says that this value "…is equal to the index of the starting hash within the widest level of the merkle tree." It's just how the binary encoding of integers happens to work! So: problem solved.

Our second weird ingredient, range/set-membership proofs, is allowed to make use of off-chain interaction, as you hopefully noticed in Section 3. So we're mostly off the hook for that one as well.

Let me add, that there is currently a whole team of Jedi-level Bitcoin cryptographers working full-steam on rangeproofs, for unrelated reasons (they want to use them on chain, to hide the transaction *amounts* of Bitcoin txns). Our use of them is significantly benign by comparison.

Leave a comment below if you think you have the best commitment scheme to use (see "(3) insurance claim" for details)!

B. Putting it Together

What *does* happen on-chain (or, more specifically, inside of a payment channel [such that these instructions might need to be pulled on chain at any time]) is:

1. A small payment from Sally to Fred, regardless of anything. This is Fred's fee, or the **"insurance premium"** (again: it is cheap, because it is on something Fred knows will never happen).
2. A very large payment from Sally to Fred, if Sally goes many blocks without revealing her secret "R"s. This payment must be larger than the insurance claim (below), to ensure that Sally does her part on time. It is a kind of **fidelity bond**.
3. A large payment from Fred to Sally, the **insurance claim**, if Fred cannot provide the part of the block that Sally requests.

(1) Insurance Premium

The first ["insurance premium"] is quite trivial to do, and is a normal channel update.

(2) Fidelity Bond

The second ["fidelity bond"] is also pretty easy, especially in LN-style channels that emphasize "hash time locked contracts". For example, we first move 10 BTC from Sally to Fred, holding these funds "hostage". Second, we move 10 BTC backwards from Fred to Sally, if and only if R is revealed during a "timelock" [ie, we return the hostage iff R is revealed]. Now, Sally will pay a net 10 BTC penalty if she never reveals R.

If she does reveal it, she keeps her 10 BTC, but Fred will learn the R – he can now use it elsewhere in this txn or other txns. As mentioned, this must be the largest payment, and it also must be a briefer timelock, which is to say that the second timelock (below) must be twice as long as this one, to ensure that Fred has at enough time (ie, worst case Fred still has a whole "timelock unit") to do his part after obtaining the critical R's.

(3) Insurance Claim

The third ["insurance claim"] is a little stranger.

The insurance claim takes the form of a 'fidelity bond' holding *Fred's* money hostage, not Sally's. Fred can only get his money back if he provides a number of items.

I will describe these items in two sections. The first section acknowledges the original parameters of the contract:

Fred must supply two integers (X, R) such that commitment(c(X, R)) matches a predefined H1, and hash(R) matches a predefined H2. These H1 and H2 values were supplied by Sally when she picked her random numbers and created the new channel-state. *

"X" is the Merkle tree index (see first 'weird ingredient', above), and "R" is a random 256-bit integer that was chosen by Sally. *

Sally will reveal R, in order to reclaim her "Fidelity Bond" (second payment, above). *

Finally, using this R, Fred can derive X (he merely needs to compute L hashes, where L is the total number of txns in the block [see above]). Therefore, he can supply it.

Easy enough so far. But there is still one last requirement for this first section:

- The commitment must be such that Sally can make an (off blockchain, arbitrarily large/interactive) rangeproof to Fred that X is an integer within range(1,L).
- (So the commitment might not take the form of a hash. And so it may require script-versioning or a new OP Code or some other even more advanced technique. Or it may only involve algebra. Choosing the best way is not my area of expertise, sorry! Ask Andrew Poelstra.)

The first section only acknowledges "what Fred agreed to do". Fred has yet to actually do it!

So, in order to fulfill the contract, Fred must also:

- Interpret x1 has a Merkle tree index, and provide a Merkle Branch that has the same index and one that ultimately hashes to "H3", the hashMerkleRoot of the header we are examining. Thus Fred shows that, whatever random number Sally picks, Fred can show her the corresponding hash.
- Finally, Fred must provide the preimage of that hash.

Basically: if Fred can show the preimage in question, he passes. Otherwise he fails.

If this preimage is a valid transaction, then everyone lives happily ever after. But if it fails validation for any reason (for example, because it is garbled nonsense), then it

will allow Sally to profit tremendously by taking advantage of "Invalidity Insurance" (section 5). The two services work together.

C. Repeat Audits

Obviously, Sally's likelihood of catching fraud on the first try is very low: 1/L probability. So, Sally should want to try again and again. And Fred should be happy to let her, because he makes a small amount of money on each audit.

D. [Optional] Converging on Missing Data

Top Down and Bottom Up

One can think of a Merkle Tree, visually, as a giant "triangle". Typically, the triangle is solved "bottom up" – which is to say that the lowest level is built first, and the "top" level is built last. Then the top item (the Merkle root) is compared against the blockheader.

Full nodes will always "solve the triangle" this way, for two simple reasons: [1] full nodes are expected to have the entire base layer already, and [2] each layer fully defines all the layers above it.

But SPV nodes may elect to also solve the triangle "top down" by working downward from the Merkle Root. This lets them "snipe" into exact locations of the triangle, without having to do the work of storing the whole thing.

(This may also interest "full nodes" that are in the process downloading and validating a newly-found block. They will need to store (for recent blocks) "the whole triangle" – ie all of the intermediate hashes.)

The "Whole Triangle"

How large is the whole triangle? Well, here is a table:

Depth	Items in This Layer	Cumulative Items So Far
1	1	1
2	2	3
3	4	7
4	8	15
5	16	31

Depth	Items in This Layer	Cumulative Items So Far
6	32	63

We see that, if "n", is the first column, then the second column is "2^(n-1)" and the third column is "(2^n)-1".

So the third column [the "whole triangle"] tends to always be twice the size of the second column [the triangle's "base"]. So, if we store the whole triangle instead of just the base, the storage requirements [for hashes] only increase by a factor of two.

And remember that full nodes are storing whole transactions, in addition to the txn-hashes. An 8 GB block, hypothetically, might consist of 32,000,000 250-byte txns. So the "triangle base" would be 32,000,000 32-byte hashes, or 1.024 GB. And, by the logic of the preceding paragraph, the "rest" of the triangle would only be ~ 1.024 GB. So instead of 9.024 GB, the storage / memory requirements for the full triangle would increase 10.048 GB.

(And, once the "whole" tree is obtained, the requirements go back down to their usual level of 9.024 GB.)

Auditing the Triangle

An 'Honest Fred' will have the whole triangle available to him, but a Fred that is missing part of the block will not. He will even be missing some "mid-to-upper" Tree values – these are much easier to audit because they are far fewer in number.

And in fact, Sally can easily ask for an arbitrary hash, somewhere *within* the triangle.

Her process for doing so is, actually, *exactly* the same bullet points as I have given above! The final bullet point, instead of revealing a Bitcoin transaction, would instead just reveal [hopefully] two 32-byte hashes. (Sally would not use this data to cash in on invalidity insurance, in fact the reverse – she would just use it as more evidence that Fred does indeed have the whole tree.)

7. Class IV Flaws (Bad Accumulators)

Until now, I've avoided talking about an entire category of Block Flaws – the "Class IV flaws". These concern aspects of a block that span multiple transactions.

I will first explain them, and then offer a strategy for transforming them into "Class I" flaws.

A. Evidence That Spans Multiple Txns

Consider violations of the blocksize limit – the infamous rule stating that the *total size* of *all* transactions (summed together) must be beneath a certain size. If Evil Fred makes a block that is too large, for example 1.0001 MB in [non-witness] size, then how can Sally ever become aware of this?

[Merely to keep the explanation as simple and clear as possible, I will use the pre-SegWit limits of 1 MB and 20,000.]

None of the techniques we have discussed so far will help. Class IV Flaws can occur in a block, even if every transaction is valid (ruling out Class I methods) and available (ruling out Class II methods). And "SPV+" mode alone will do nothing to help us solve it (because having the coinbase txn does not help us).

Invalidity Insurance, in particular, only highlights *individual txns* as being wrong. But Class IV fraud is not "at the txn level", it is "at the block level".

So any rules that operate at the "block level", such as the blocksize/SIGOPs limits, we cannot check. It is problematic enough, that one hidden txn somewhere might singlehandedly be 40 terabytes long and contain a trillion SigOps. But, much *worse* is the prospect of tiny txn that use zero SigOps– because of these, a "crammed block" with many txns might still be valid. Or it might not. We don't know!

In short: we need *all* of the parts, so that we may properly add them!

Or do we?

B. A Valid Contributor to the Whole

Perhaps, we *could* analyze it in parts, **if each part uniquely declared its contribution to the whole**. We'd just need a way to measure a "contribution".

Let me explain.

For the blocksize limit specifically, we would force each txn to declare the "coordinates" of the "block real estate" that it was "living on". We would not allow any txn to take up more (or less) space than it declared; and we would not allow any two txns to take up the same declared space.

So, a transaction wouldn't just "take up 250 bytes"; instead, we would arrange things so that the txn would "take up **the** 250 bytes between byte #32,880 and byte #33,130".

A metaphor: imagine that a block's txns are always printed out on strips of paper, and that these strips are laid end-to-end, forming one line segment. I am proposing that we [first] lay this line segment against a gigantic ruler, and [second] that we staple two blank 4-byte chunks[11] to each paper strip, and [third] mark these chunks with each strip's starting and ending coordinates, as given by the ruler.

Above, in "SPV+" mode, I assumed that SPV nodes would have access to the first and last txn of every block. If the "accumulator" starts at 0, and if each txn adds exactly as much to the size of the block as it *says* it adds, then the ending value of the accumulator should be exactly equal to the total size of the block. If it does not, there must be one or more *specific* txns which are flawed. Such txns would then have Class I flaws, and trigger "invalidity insurance" (section 5).

C. Accumulator Specifics

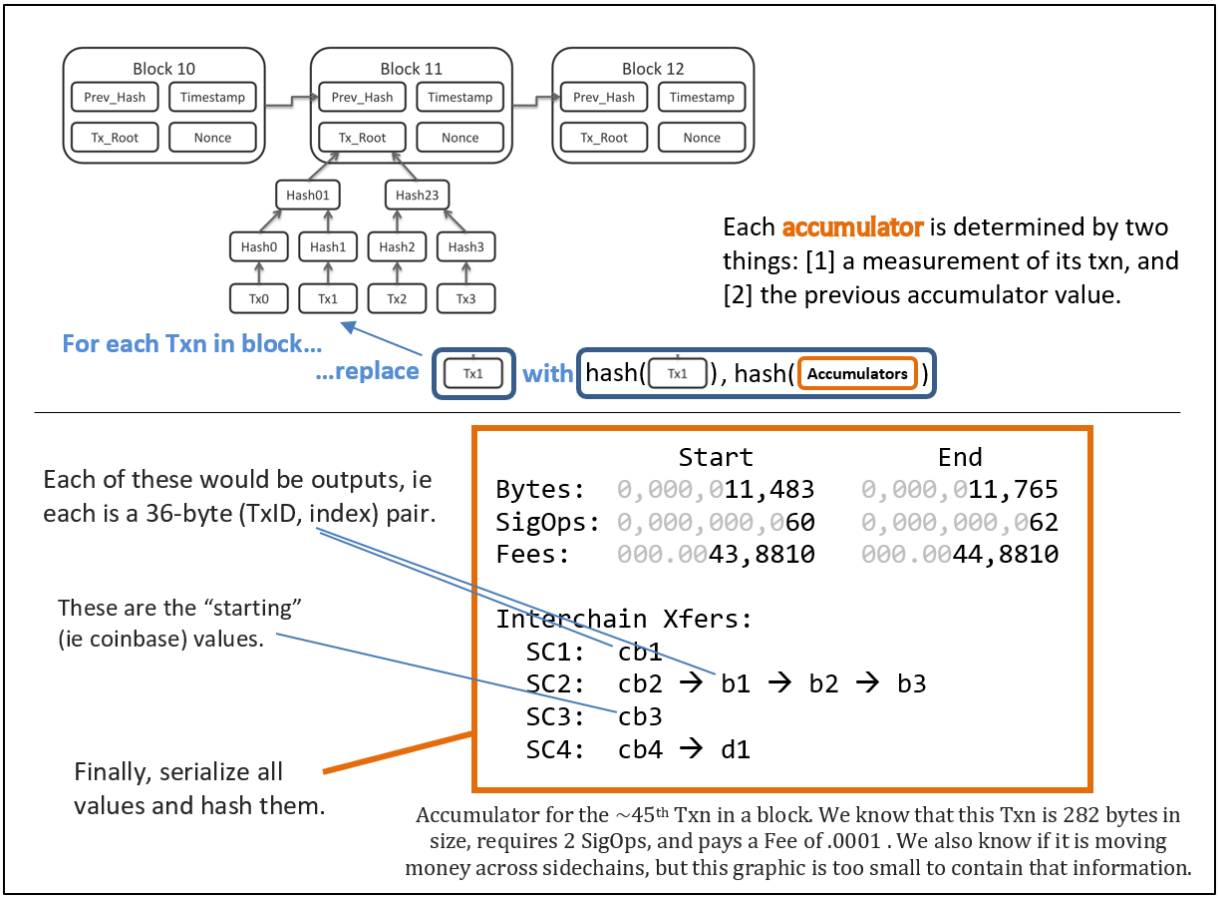
I will start with a hard fork accumulator, because it is easier to explain.

Hard Fork

[Reminder] Currently in Bitcoin, a txn "x" has hash "h(x)", and these h(x) all form the base of the block's Merkle Tree.

Instead, we might have each txn "x" be described by "y = h(h(x), h(z1, z2))". The second item would be an "accumulator" – for example, z1 and z2 would describe the starting and ending size-coordinates of the transaction (along the entire vector of all txns).

(More precisely, z1 and z2 would themselves lists [of multiple information], because they would need to describe, not only the byte-size of the transaction, but also its SigOps-use, txn fees paid, and any change made to drivechain's Withdrawal_DB.)



Accumulator for the ~45th Txn in a block. We know that this Txn is 282 bytes in size, requires 2 SigOps, and pays a Fee of .0001 . We also know if it is moving money across sidechains, but this graphic is too small to contain that information.

The new information z1 and z2 does not actually need to be broadcast over the network at all, or even stored. It can be generated and validated locally using just the sequence of x's, just as it is done today. Miners are still 100% free to arrange the txn in any order they like, at any time [before the block is found].

Anyway, this data structure ("cumulative Merkle tree"?) is helpful because it makes it very easy to demonstrate fraud. Recall that we need to submit both a "bad txn" and "evidence" that the txn is bad. We will already have "y" itself, when we describe a path to a "bad txn". So now we just need to supply its preimage [h(x),z1,z2] and then disregard the first 32-bytes.

Soft Fork

The space of possible soft forks is very large. I'm not sure what the *best* soft fork for this part is. More research is needed! Leave a comment below.

But one way would be to think of the "(z1, z2)" pair as a "second witness" to its txn "x". Just like with SegWit, we could build another Merkle Tree (a third one this time) containing *these* witnesses, and require that it exist and that there always be a commitment to it in the coinbase txn.

The downside is that in order to prove fraud you just need a very awkward and cumbersome txn, consisting of three Merkle Branches. First, you need a Merkle Branch to the offending txn, as always.

But second, you'd need one to the block's coinbase txn; and then, third, you'd need to [1] pick a specific output of that coinbase txn, [2] scan that output to make sure that it is the 'second-witness-output', and then [3] go down another Merkle Branch from there, down to the hash of the 'second witness' of "x".

And all of that would be just one part of a big channel script! What a crazy redeem script that would be!! Thank heavens for Ivy. And I guess that this txn, if broadcast on chain, would probably be at least ten times as large as a normal txn. I shudder at the thought! But I suppose "cumbersome awkward txns" has never stopped us before…

A second technique would be to require every txn to be paired with a mindless "zombie txn" – one that doesn't actually move any money, but only exists so that we can put an OP Return in there with the commitment to the second witness. This idea is quite horrible, as it wastes ~ 80 (?) on-chain bytes for every txn, and the median txn is already near-250 bytes. So txns would be about 30% bigger! How horrible! I only provide this idea to help stimulate thought for other soft fork ideas. (Note that we can't put the commitment in an OP Return inside its own txn, because users cannot easily know "where" their txn will be positioned in the block without interacting with a miner, which would be much too annoying.)

8. Economics

A. Computational Costs

First, as I mentioned above, Sally must run an "SPV+" node, which might be expected to use up 2192 bytes per block, instead of a 80.

B. Market-Clearing Price

How much are "invalidity insurance" and "fullness insurance" likely to cost?

i. Competitiveness

First, notice that these insurance services could be provided by anyone running a full node. Second, notice that, by definition, if a blockchain network exists at all, then there are multiple untrusted anonymous parties each running nodes. So "insurance providers" are likely to be under heavy competition, and will have to offer the service at near-marginal cost.

ii. Marginal Cost Overview

What are their marginal costs? Well, to provide the service, one needs a full node, an open payment channel, and they need to monitor it. Fortunately, many people **already** run full nodes, and will already monitoring their open payment channels, so those marginal costs are actually zero. Only the channels will cost anything – a small fee to open a channel (or roll over an existing one), and the working capital locked inside of the channel.

But even these costs could potentially be zero. These contracts can be deployed inside payment channels that have already been opened, so while the cost of opening a new channel is nonzero, the marginal cost of opening a channel for this specific purpose remains at zero – people can freely reuse the channels they already have open. Anyone who manage their LN-connections well, won't need to open or close very many, for this purpose or for any other.

Note: Channel Convenience

Since we are discussing the opening/closing of channels, it is important to remark that this insurance can be spread "through" the LN-network – in other words, Sally and Fred do not need to have a channel directly between them. They merely need to be part of the same "Lightning Network". So Sally can actually buy insurance from Fred, and resell it to another Sally! This minimizes the need for channel openings/closings.

Details are in this lengthy footnote[12].

iii. Time and Working Capital

Fullness Audits

The Fullness Audits need only last a brief instant of time. The required steps can happen very quickly: Sally and Fred meet, negotiate terms, create new channel states [kinetic], Fred invalidates, Sally invalidates, Sally reveals her random integer(s), Fred reveals the txns associated with it, and they jointly move to a new [potential] channel state.

Since it has almost no marginal cost, there is no limit to how cheap each fullness audit might cost. Each audit could cost one hundredth of a cent, and each user might do 40 or 50 audits per block, for a per block cost of half a cent.

I will frame all of the costs on a 'per month' basis. This makes it easier to compare the "fraud proof world" costs to the costs in "full node world".

At 4383 blocks per month, and 50 audits per block (auditing *every* block), and (1/100th) cent per audit, the monthly cost of these audits comes to $21.91.

Invalidity Insurance

The Invalidity Insurance needs to last longer, unfortunately, because it needs to prove a negative. The question of how long it must take is a **security parameter**; one which asks: "If this block contained a bad transaction somewhere, how long would it take for someone to find?".

Once a flaw has been discovered, it will spread quickly. This is because informants will quickly try to *sell it*.

These sales are easily facilitated, by taking the logic of the Invalidity/Fullness contracts and reversing a part of it. A new 'smart contract' transaction will be of the form: "You pay me $0.03 now, and in return I will either give you a flat $1000, *or* I will give you a block flaw for this block header"[13]. Since "Indiana the Informant" knows that he has a real block flaw, he is motivated to sell his knowledge of the flaw to whomever will buy it. And he motivated to make these sales ASAP, as the value of his 'exclusive information' will plummet as soon as it becomes less exclusive.

So the security parameter needs to vary with how long it takes until *the first* honest person finds a flaw[14]. How long will this be? We cannot be too sure, but 12 hours seems to be a reasonable duration.

What is the cost of locking up a dollar of working capital for 12 hours? It depends on the market interest rate. Since an Honest Fred has no "risk" associated with this project, the rate should be the [theoretically-minimal] "risk free rate", and since BTC is a deflationary unit of money, it does not need to appreciate to compensate the investor [to offset the inflation tax]. So the required rate of [annual] interest on this "project" could possibly be very low, it could be 2% or 1% – it depends on what other investment projects are out there [with this risk-reward profile]. Currently, *nothing* is out there, so the market rate could be arbitrarily close to zero.

Below I have rescaled some interest rates from annual to daily, hourly, and per 10 mins:

	Annual	Daily	Hourly	per 10 min
sub-periods (s)	1	365.25	8766	52596
	5.0%	1.34E-04	5.57E-06	9.28E-07
	2.0%	5.42E-05	2.26E-06	3.77E-07
	1.0%	2.72E-05	1.14E-06	1.89E-07
	0.5%	1.37E-05	5.69E-07	9.48E-08

Table 1. Rescales annual interest rates (to daily or hourly), per the formula r_new = [(1 + r_old) ^ (1 / s)] - 1 .

In addition to deciding the *duration* of coverage, a second security parameter is *how much coverage* one would like to buy. I have assumed $1000 worth. When a seller sets his $1000 up against a few cents, it implies that the seller earnestly believes that there is a >99.99% chance that the block is valid.

Finally, for ease of interpretation, I will assume that Sally always insures *every* blockheader. This helps compare her level of paranoia to that of a regular 'full node' user (who would also check every single block).

security parameter	12	hours of coverage, per blockheader
security parameter	$1,000	value of coverage, per blockheader
	12,000	coverage-hours, per blockheader
	4,383	blockheaders per month
	52,596,000	coverage-hours, per month

Table 2. Expected quantity / duation of coverage.

Thus, we can see the monthly cost of buying all of this invalidity insurance, for different interest rates.

	Monthly Cost	Cost per Block
5.0%	$292.74	$0.067
2.0%	$118.82	$0.027
1.0%	$59.70	$0.014
0.5%	$29.93	$0.007

And we see that it may cost as little as $29.93 a month.

Total Cost

Thus, the total cost of validating every block in this "negative" manner comes to $51.84 a month.

This cost is driven by two things: first, the number of Fullness Audits performed per block; and second, the security parameter choices (coverage amount, coverage duration) for Invalidity Insurance.

One could reasonably argue that each of these should increase as the network becomes more expensive (ie, as the "blocksize" increases). After all, if the block is larger, it becomes reasonable to pepper it with a proportionally greater number of Fullness Audits; and it may also be wiser to increase the 'coverage duration' for Invalidity Insurance.

If so, this would seem to contradict the purpose of using Fraud Proofs in the first place!

Cost-Scalability

However, I believe that one could also reasonably argue the opposite: that the value of $51.84/month could be stable indefinitely, or even decrease over time.

For example, an increase in the absolute number of independent full nodes, would inevitably mean that there would also be an increase in the absolute number of nodes *running on disproportionately superior hardware/software*. These super-nodes will detect threats sooner – through the process outlined above, some of them may *monetize* this detection.

It is even possible that greater absolute number of *SPV nodes* will make it profitable for "specialist validators" to emerge – and only a few of these are required to grant blanket protection to the entire community (a possible application of the "Many Eyes hypothesis", or of "herd immunity").

C. Partial Fullness

Sally does not necessarily need to validate *every* block, of course – millions of SPV mobile users are today getting along just fine, while validating zero blocks!

Certainly, in general, more validation is preferred to less, but Sally might opt to only validate a few blocks. Perhaps, when receiving money, Sally validates the block that contains her new txn, and the 400 blocks which precede it, and then the 20 blocks which follow it. Or, perhaps Sally just validates a few blocks at random.

9. Conclusion

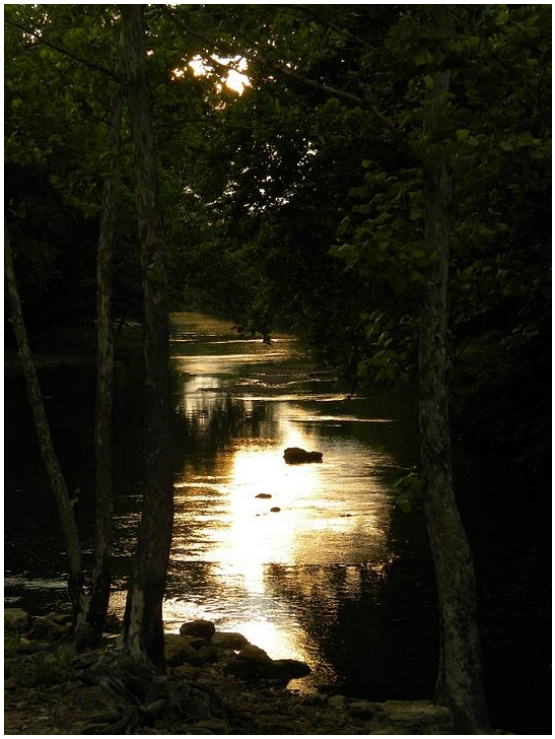
Fraud Proofs address the problem of learning that a Bitcoin header is invalid (despite it being PoW valid). Such cases are very rare. In fact, I think that 100% of the historical cases were mistakes (ie accidental), and were corrected as soon as possible.

Moreover, this post expresses many ideas, and each of them would be its own medium-sized project. If blocks are easy for the layperson to validate, then working

on fraud proofs is certainly a waste of time. However, if blocks are difficult-to-validate, then fraud proofs may be very useful.

To be honest with you, I wrote this post mostly as an "exhaust valve" to get these thoughts out of my head! They were driving me crazy!

But now that we've meditated I feel better. I hope you've enjoyed reading!



(Sunset art from Max Pixel)

Footnotes

1. The "Full" vs "SPV" distinction is actually not as sharp as might casually be believed. For starters, when a "full node" is downloading the next block, it *is* itself in SPV mode, with respect to that next block. And consider the case where 51% hashrate secretly runs a new piece of software (and, correspondingly, runs a new-but-compatible protocol) – one with a mandatory extension block. Then, despite intending to be a "full node", you have been forced to become partially-full, and partially-SPV. If the miners later change the protocol back, removing the ext-block requirement, then you will return to being 100% full. But you will not be aware of having transitioned in either direction, and in fact you have no way of knowing. So, these terms only make sense if the protocol is fixed, so that is what I assume

in this essay. In reality, however, the protocol can and does change; and miners are always allowed to run secret, customized software. ⮌

2. Well, each block that she wishes to "fully validate" by this SPV method. ⮌
3. For blockchains which are *themselves* sidechains, the inter-chain transfers also become Class I flaws. To be alerted to fraud, the sidechain SPV node would get two txns, the sidechain deposit and the mainchain txn that originated it. Then the 'Sidechain Sally' must check both blockchains (main and side) for Flaws. ⮌
4. The total per-block byte-cost is: "header + first_txn + last_txn + 2(32log_2(n))", where "log_2(n)" is "ceiling of the log_base_two of the number of txns in the block". Thus we have "80 + 1000 + 280 + 2(32log_2(5000))" in our example, or "80 + 1280 + 832", or 2192. Notice that we do not need the bitmask for the coinbase, because we already know its exact structure ("always left"), but we might use it for 'last txn' (because it alternates between "right" and "self"). (This would be smaller than, say, encoding "self" as a branch-hash consisting entirely of zeros, as I have estimated it here.) ⮌
5. In practice, Sally and Fred might do one random number at at time. If so, their worst-case-scenario script size will be smaller, but they will need to do more audits. ⮌ ⮌[2]
6. This is a simplification for the story. In reality Sally would not reveal the integers, she would instead reveal the randomness she used to compute the commitment (ie "open"on Wikipedia). ⮌
7. Each pairwise exchange can be quite customizable, and probably *will* be customized to [for example] improve privacy. But for my simple explanation, "shared H" is simple enough. ⮌
8. Recall that, at this point in the validation process, Sally already has the entire header, and knows that it meets the difficulty requirement. She only needs to evaluate the hashMerkleRoot. ⮌
9. Of particular concern is a situation where txn-validation changes over time (for example when we soft-fork-in a new OP Code) – previously valid [but not-included] txns may become invalid. Thus, if someone accuses one of these txns of being invalid, we would need to know the exact time, and consensus rules it had when it was included, vs now. What a nightmare! But for a *given* static protocol, whose txn-validation rules do not change, there is no problem (as I mentioned in footnote #1). Perhaps this can be solved by incrementing the block version number each time the consensus rules are changed via soft fork. Or else, it can be solved if Sally upgrades her software – she can stay in SPV mode, but she needs to be running the *latest* SPV mode to enforce the latest rules. ⮌
10. Ie, check that the formatting is intelligible, that the input scripts are valid, that sum(inputs) >= sum(outputs), etc. Everything that is checked during normal "full node" validation. ⮌
11. A four byte unsigned integer can count to 4.29 billion. So that enough to uniquely describe one point along a vector, as long as the blocksize does not

exceed 4.29 GB. And, even if it does, we can probably use the modulus (ie, allow it to overflow), because it should be readily apparent, from the size of the Merkle tree, whether the block is <4.29 GB, <8.58, etc. As I explain later, the data does not actually need to be broadcast over the network at all, or stored. It is just about reaching the correct hashMerkleRoot ⮌

12. The intermediate nodes take on the role of intermediaries such as "insurance originators" or "reinsurance agents" – the first group sells insurance to end-customers, while planning to immediate sell the contract to a 'real' insurance agency; the second group sells insurance to 'real' insurance agencies, thus taking contracts (and risk) off of their hands. — — For Invalidity Insurance, one can think of the "evidence" txn that completes the script as just a larger and more complicated version of "R" [recall that R is a random number chosen by the buyer, who also calculates h(R)=H and builds a "circuit" of payments based on the same H]. The "evidence", once revealed somewhere, will trigger all of the insurance payments in a given circuit – in fact, it will trigger all of the payments for *all* circuits everywhere [for a given invalid block header, of course]. — — For Fullness Audits, the "originators" would bear more risk, and could possibly extract a higher return as a result. This is because the originators ["Sally1"] would first need to Audit a real full node ["Fred"], until they are convinced that the full node really is full (in other words, Sally1 audits Fred until she is convinced that he is an Honest Fred). Sally1 can then safely sell Audits to, for example, Sally2. Sally1 won't be able to complete the audit by herself, because Sally2 will ask for random txns that, by definition, neither of them will have. But Sally1 should be able to get these txns from Fred with a new audit. — — None of these techniques make any sense at current levels of blockchain scale. But, theoretically, if blocks become absurdly difficult to validate (perhaps because they were terabytes in size), these levels of specialization would start to become efficient. They would also tend to become more efficient as channel openings/closings become more expensive – this is clear from the 'Sally2' example above, where several headaches would be avoided if Sally2 just opened a channel with Fred directly. ⮌
13. Indiana can sell Class II flaws this way, just as easily as Class I flaws. He will merely offer to buy Fullness Audits at above-market rates. All of the reinsurers will flock to him, and he will collect damages from all of them. They, in turn, will immediately try to pull the same scam on anyone who hasn't fallen for it already. So the critical information [that a specific parts of this block is missing] will still spread rapidly. ⮌
14. Notice that, by using smart contracts and free market trade, we have achieved some specialization of labor and the associated welfare gains. ⮌

Tweetstorm: Bitcoin will usher an era of unprecedented peace and prosperity

By Murad Mahmudov

Posted April 14, 2018

1/ Bitcoin will usher an era of unprecedented peace and prosperity.

2/ A modern variant of European 'La Belle Epoque' of 1871-1914, when the peak of the Gold standard put some semblance of a cap on State power. Money is Power. Bitcoin is the Neo-Gold standard. Murad Mahmudov

3/ Wars, wars on drugs, torture prison networks, mass Foucaldian panopticon-like surveillance machines, military-industrial complexes, prison-industrial complexes etc…

4/ …all become *significantly* more expensive if you have to pay punishers' salaries with scarce gold bricks (bitcoin) rather than infinitely printable (read: stolen) fiat money.

5/ For better or worse, systems like social security, medicare and medicaid will collapse as well. Thousands of commercial banks globally will collapse as well.

6/ Cryptocurrencies make vanilla taxation, inflation taxation and fractional reserve banking either significantly more difficult or entirely impossible.

7/ Prepare to enter a hypercapitalistic, hypercompetitive, decentralized, borderless world of unstoppable commerce, where the power of governments is cut in half, and the power of sovereign individuals is doubled.

8/ Deluded people who use ridiculous and laughable terms like "post-capitalism" don't know what they're talking about. Capitalism is only just beginning. This is the first reemergence of truly 'free markets' since pre-Babylonian, primitive times.

9/ Because as long as the State has a monopoly on money creation, no market is 'free'. Cryptography is a weapon. Marxism is Totalitarian slavery. Capitalism is unrestricted Freedom.

10/ Thankfully the fabric of reality itself is an anarchic, libertarian leaning structure at equilibrium. Nothing can ever be successfully top-down centrally planned in the long run. Only free will, human action and markets can point us in the direction of growth.

11/ When (not if) some Asian central bank or a large sovereign wealth fund announces they put 5% of their reserves into a basket of blue-chip cryptocurrencies, the world will never the same again.

12/ Bitcoin is a profound Economic Rennaisance, falsely wrapped in a Tech Bubble, itself falsely wrapped in a Get-Rich-Quick Scheme. The complete takeover success of cryptocurrencies is inevitable destiny. It is mathematically inescapable. Once you see this, you can't unsee it.

A Guide To Bitcoin's Technical Brilliance (For Non-Programmers)

By Lucas Nuzzi

April 15, 2018

18 minutes is all that it takes to understand Bitcoin better than most people.

In 18 minutes, you will have a good understanding of how hash functions, Public Key Cryptography and Merkle Trees are brilliantly used in Bitcoin.

The purpose of this post is to provide a *semi*-technical guide to key aspects of Bitcoin's technology. Over the years, I have found that the best way of achieving that is to dissect everything that happens under the hood when two entities transact.

Of course, I'm talking about Alice and Bob.

But bear with me. It gets dark.

Alice happens to live in country that is going through a massive economic downfall, widely regarded now as a humanitarian disaster. A country where hyperinflation, hunger, unemployment, and violent crime have prevented its citizens from the most basic human rights.

Despite economic calamity, life goes on, and people still need to purchase goods and services on a daily basis. Alice desperately needs to fix the roof of her house, which has partially collapsed. But fixing a roof takes longer than a day, and pricing this service using a hyper-inflated national currency is nearly impossible.

Historically, when a country's national currency collapses, citizens tend to adopt stable foreign currencies in order to hedge against inflation. Many South American countries that went through periods of hyperinflation informally adopted the US dollar when trying to price services.

But Alice is having problems finding physical US dollar bills. The ongoing crisis, which has now reached its third year, has put a premium on foreign currencies. In order to hedge against hyperinflation, she proposes to pay Roberto (Bob), the roofer, in Bitcoin, which is surprisingly less volatile than her own country's currency.

Bob requires 0.01 BTC to begin the repairs, but he does not have a bitcoin wallet. In addition to sending him the funds, Alice will also teach Bob how to create a wallet and use Bitcoin.

1) Alice wants to send 0.01 BTC to Bob (I will use quotations to note all the steps in this transaction's life-cycle)

To transact in the Bitcoin Network, every participant is required to download a specific software to interact with other network participants. This is what's referred to as a *client*, which sends requests for specific data to a *server*. Web browsers, for example, are a type of client that request and interpret data from a server. When a URL is accessed, the browser client sends a request to the server storing the website's content and, once served the data, it displays that content to the user.

Bitcoin clients work similarly, but there is a key difference. Instead of accessing data from a centralized server, the Bitcoin client interacts with other members of the network to source and validate the integrity of the data, which can easily be determined since every participant of this network is essentially storing the same database. This is a key aspect of the Bitcoin Network, as its decentralization grants unique security properties to the integrity of the data that is exchanged between network participants, or nodes.

Nodes, by definition, are the individual participants of a network. When we refer to the nodes of the Bitcoin Network, we are talking about the people that have downloaded and that are running a Bitcoin client. Collectively, Bitcoin nodes *are* essentially what the network is made of; a group of interconnected machines running a client and exchanging data.

Nodes may interact with the network by running two types of client; a full client or a light client. Let's first look at the differences between them:

FULL CLIENTS (ALSO CALLED FULL NODES, OR "THICK" CLIENTS)

As many of us know, a *blockchain* is simply a structure that can be used to store data. In Bitcoin, transactions are added to *block*(a group) every 10 minutes, and then *chained*(connected) to the previous block. The purpose of this data structure is to determine a single, immutable, truth of what happened and when it happened.

Full clients, also called full nodes, store a complete copy of the blockchain in their computers which consists of all blocks with all transactions that have ever occurred in the network; from the very first transaction in the Genesis Block, the first block to ever be mined, all the way to the most recent block the software is able to find. Full nodes are the backbone of the Bitcoin Network, and enforce the rules set forth by the software. These rules guarantee the security, formatting and consistency of all data that is stored on the blockchain, as well as the process by which new data is amended to the ledger.

Since full nodes store a copy of all transactions that have occurred in the network, they are able verify the validity of transactions initiated by any other user. To do that, the software looks at all past transactions that a user has engaged in, thereby preventing malicious actors from sending more than what they possess. Because this is digital money, the only way of preventing a bitcoin from being used twice (Ctl+C, Ctl+V) is to be hyperaware of all transactions that have ever happened.

Downloading an entire copy of the blockchain from other nodes may take some time as the size of the Bitcoin blockchain is currently 164GB. However, this process only needs to be done once. Upon downloading the chain, users only need to download a new block every 10 minutes. All of this is done in the background, automatically, by the client.

LIGHT CLIENTS (ALSO CALLED "THIN" CLIENTS)

Since running a full node has several requirements, such as having sufficient storage space and memory, light clients were designed to ease the interactions with the network and reduce the frictions of owning and using bitcoin. Light clients can, for example, run on a smartphone. Instead of storing a complete copy of the blockchain, light clients only store the *header* of each block, which is basically a summary of all transactions contained in it.

Only storing the block header requires less disk space, but it limits what light clients can do. A light client is not able to entirely verify the validity of a transaction, but it can confirm by looking at the header if a transaction was included in a block. Its main purpose is to broadcast transactions to the network. Full nodes then receive transactions from light clients, verify them, and add them to a separate bucket with

all other unprocessed transactions. Full nodes are only allowed to add these unprocessed transactions to their blockchain when the *miners* tell them to so. More on that later.

2) Alice helps Bob download a Bitcoin light client, which will enable him to quickly create an account and receive funds.

MULTIPLE CLIENTS AND SERVICE UPTIME

Today, there are dozens of different Bitcoin clients available for download. This diversity is highly desirable, since it diminishes the risk that a bug in a single client will disrupt the entire network. If one client fails, as it has happened in the past, users can download other *brands* and transact.

This is the reason why Bitcoin has been functional for 99.99% of the time. Not many internet services have comparable uptime, which is in itself remarkable for a 9-year-old open-source project that was launched by an obscure figure.

WALLETS

Although both full node clients and light clients can serve as a user's wallet, the definition of a Bitcoin wallet is not the same as a client. Technically, a wallet is the collection of data required to send and receive bitcoin. This data includes a public address and a private password.

BITCOIN IDENTITIES & KEYS

Since Bitcoin was built for peer-to-peer payments, all users of the network need a public identity that enables them to identify the parties of every transaction. Like a bank account, this address needs to be unique and users must be able to share it publicly. Conversely, users also need to be able to authorize transactions with a unique private identifier that proves the ownership of funds.

Bitcoin archives that through three unique identifiers:

1) PRIVATE KEY

The Private Key is a completely random combination of numbers and letters that can be used to spend the bitcoin associated with a specific Address. Private Keys, like the PIN number of a checking account, are used to authorize transactions. Below is an example of what a Private Key looks like.

```
cxprv9xg3pXGrrmSQNqRCZRFmphUZpkzt8s43ESotbcPXk5fLXt6NT3fh2tTPyQ7tW2SWAS9uWjhD
Jzzex9m8qmAHsJvTN1hctsgiYFK9Moo9Nx1
```

As you can see, it would be extremely difficult for a human to memorize a Private Key. I don't even recommend writing it down because you will lose your funds

foreverif you happen to miss a character. For this reason, you can represent the Private Key's *blob* in a human-readable format using something called a **mnemonic key**. Believe it or not, the twelve-word mnemonic key below represents the Private Key above.

```
faith joke visa range turkey expose they bacon gentle hill cushion recipe
```

Much easier to remember, right? This is one of the most under-appreciated aspects of Bitcoin. **For the first time in human history, you can store your wealth in your brain without a third party.**

This means that, if the situation in Alice's country turns into a Civil War, she can flee the country and transfer her wealth by only remembering **twelve words**. These twelve words can later be loaded into any Bitcoin wallet.

2) PUBLIC KEY

The Public Key is derived from the **Private Key** using fancy algebra, and acts as an intermediary between a user's private and public identities. Below is an example of what a Public Key looks like.

```
xpub6E9pP9ny45P14SNMCzCBFCPwr2QHgWQqZggJg6sMjnGgPo8Hf9tzPwtzHYeKXn6GdACpoKRcv
kb2w6pvcAj6kwdw5mKLyDErWXKX8Bhozed
```

3) ADDRESSES

Addresses are derived from the **Public Key** using a hash function. I will provide more details on hash functions in later sections, but for now it is important to note that **Public Keys and Addresses are not the same**.

I often get asked "what is the point of the Public Key if users transact using addresses?" There are a couple of technical reasons behind this intermediation, but it all comes down to privacy and efficiency. Users derive their addresses from their Public Keys, which makes it difficult for someone to pin point a single identity in the network.

Below is an example of what an address looks like. As you can see, the hash of the public key is smaller than the public key itself, which saves space on the chain.

```
1LGpghBaX7AGbxA5dvpVwR7vMy53R8HcXX
```

EASY TO VERIFY, PRATICALY IMPOSSIBLE TO REVERT

As you can see in the Figure 1 below, there is a mathematical relationship between Private Keys, Public Keys and Addresses. This is achieved by using a principle called a Mathematical Trap Door; a one-way mathematical function that can easily be performed in one direction (i.e. deriving Public Keys from Private Keys), but nearly

impossible to be performed in the opposite direction (i.e. deriving Private Keys from Public Keys).

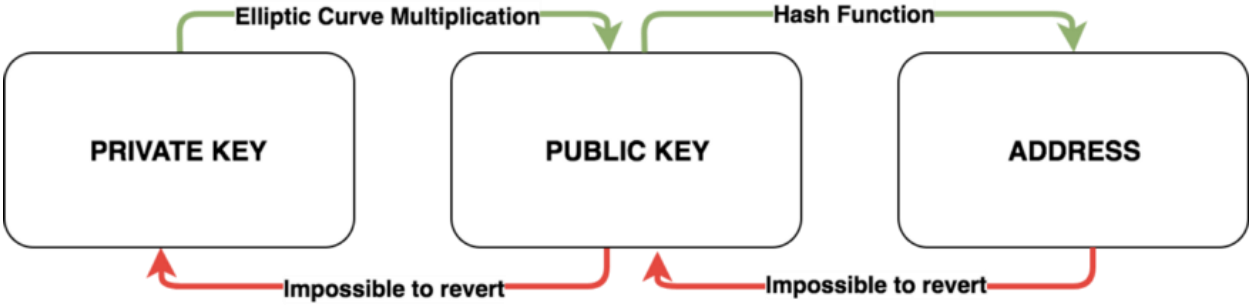


Fig 1. The Relationship of Private Keys, Public Keys and Addresses (Source: Digital Asset Research)

Even though the math behind these operations is mind-blowing (in a good way), I will save its details for more a technical post on Elliptic Curve Cryptography and other technologies being currently researched. For now, keep in mind that, when Alice *signs* a transaction, she is essentially using her Private Key to create a signature. Through this signature, everyone can easily verify that Alice has a Private Key associated with an address **without knowing what the Private Key actually *is*.**

Fun estimates that I have to share:

Q: how many bitcoin addresses can users generate?

A: 1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976

Q: how many grains of sand are there on earth?

A: 9,223,372,036,854,775,807

Have I mentioned that it would take the amount of energy in the entire sun to brute force a Private Key from a Public Key?

Math is magical.

4) Bob has created a wallet through his light client and sends Alice his Bitcoin address over email.

THE TRANSACTION

Simply put, a bitcoin transaction is a signed message that authorizes the transfer of funds from one account to another. Each transaction includes the sender's address, the receiver's address, and a signature generated using the sender's Private Key.

The Bitcoin blockchain has a native multiple-entry accounting system and each transaction has inputs (where the balance came from) and outputs (where it is going

to), much like a system of debits and credits. A user's balance is basically the sum of all debits associated with the Addresses derived from his or her Public Key minus all credits that have been sent to other addresses. A balance is considered *spent* once it's used in a transaction.

5) Alice now has Bob's address (1Few1623…), which is the only thing she needs to initiate a transaction.

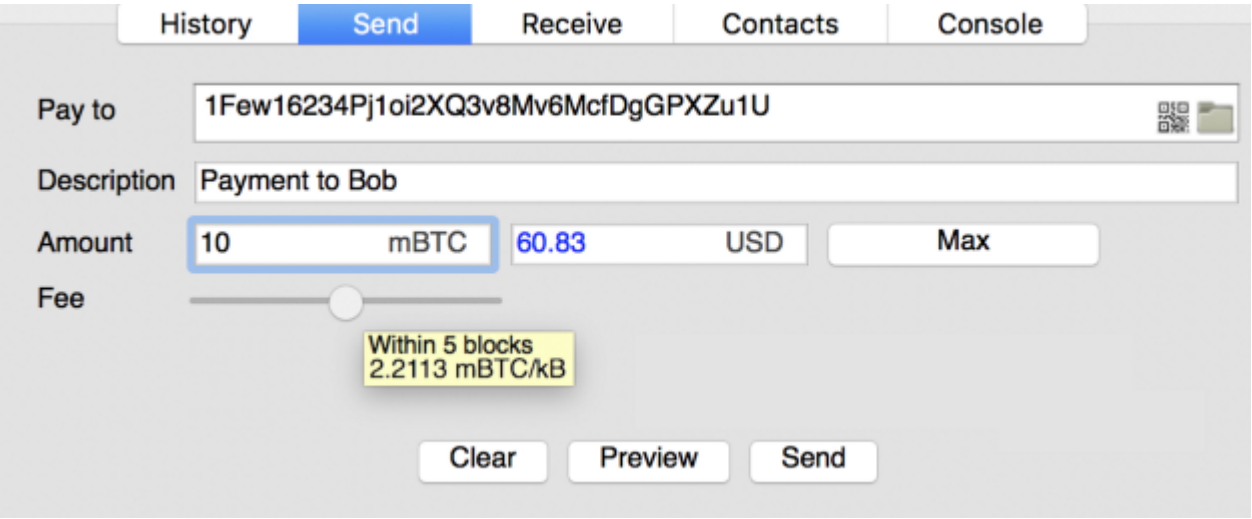


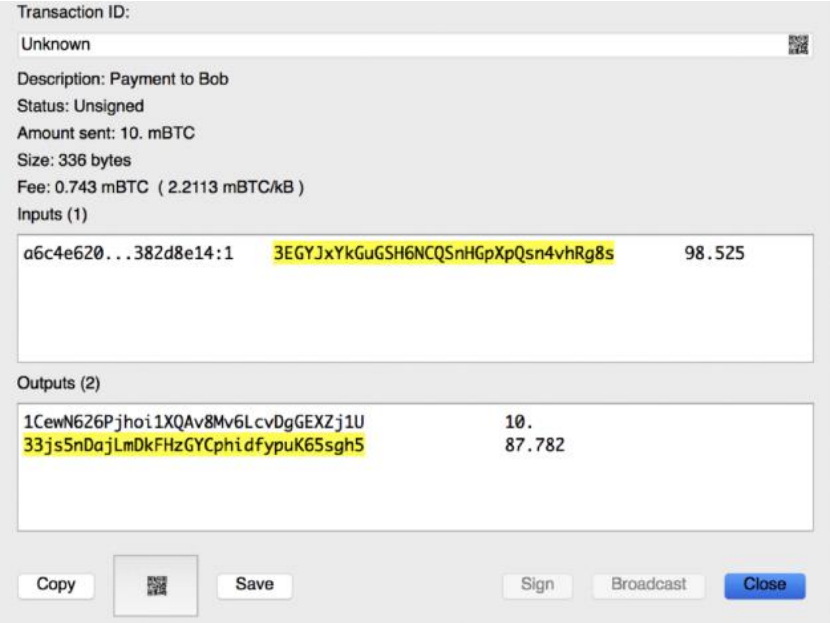
Fig. 2. Initiating a BTC Transaction using Electrum (Source: Digital Asset Research)

When Alice initiates a transaction of 0.01 BTC, or 10 millibits (mBTC), to Bob, her Bitcoin client will look at all previous *unspent* outputs (debits) associated with her Public Key, and display her total balance. Alice then scans or copies Bob's address, adds the transaction amount and chooses how fast she wants the funds to reach Bob. The speed at which Bob will receive Alice's transaction is dependent upon how much network fees Alice is willing to pay.

6) Alice decides how much she will pay in fees, which is proportionate to the speed at which Bob will receive the funds.

Bitcoin relies on certain network participants to group unprocessed transactions into a block and add that block to the ledger, an activity called mining. Miners receive all transaction fees in a block and, therefore, have an economic incentive to prioritize transactions with higher fees. If Alice wants Bob to receive the funds in less than one hour, she will have to pay the miners higher fees.

Since Bob won't need the funds immediately, Alice decides to pay a rate of 2.21 mBTC per kilobyte in transaction fees, which is enough to make sure Bob receives her transaction within the next 50 minutes, or 5 blocks. Alice previews the transaction to see how much she will pay in fees.



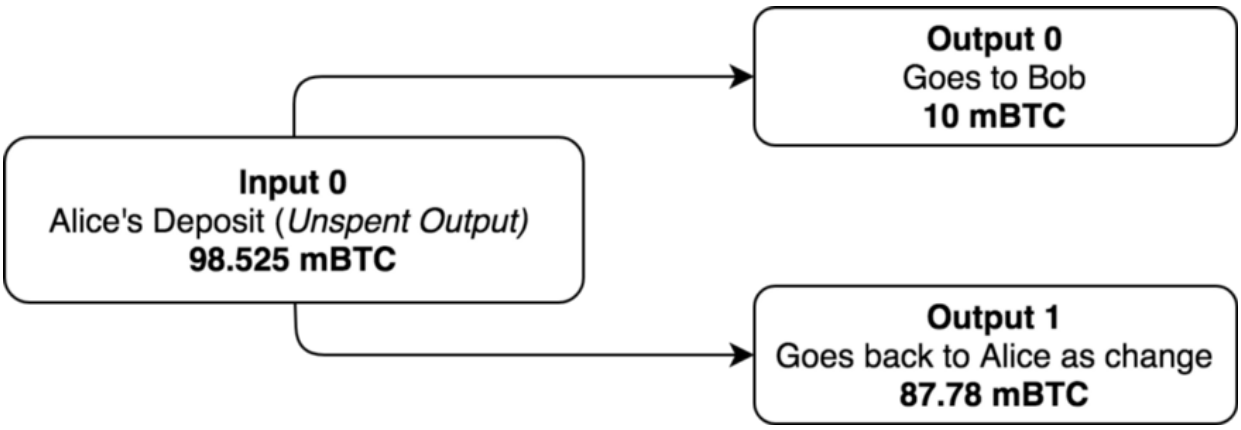
Previewing a BTC Transaction using Electrum (Source: Digital Asset Research)

At a transaction fee rate of 2.21 mBTC per kilobyte, and the size of her transaction being 336 bytes, the total amount paid in network fees is 0.743 mBTC, or $4.65 USD (2.21 * 336).

Note: the screenshot on the left was taken when Bitcoin transactions were at an all-time-high. A similar transaction today would cost less than a dollar.

In essence, a Bitcoin transaction is the exchange of ledger entries, which are represented as inputs and outputs. The specific transaction initiated by Alice contains only one input and two outputs. To send 10 mBTC to Bob, Alice's client uses a past transaction as an input to prove that she has a balance higher than the amount being sent. A day earlier, Alice deposited 98.525 mBTC to the address she will use to send the funds to Bob. Her client will use this transaction as an input since its value is higher than the amount she will send to Bob. Her client will also send the difference between yesterday's deposit and 10 mBTC (in other words, the *change*) back to Alice.

Two outputs are created; the first output of 10 mBTC goes to Bob, as it represents what Alice is sending, and the second output of 87.78 mBTC goes back to Alice's address as change. Notice that the 87.78 mBTC output that Alice will receive as change is already accounting for the 0.74 mBTC that she will pay in network fees.



The Inputs And Outputs Of Alice's Transaction (Source: Digital Asset Research)

At times, inputs and outputs can be confusing. That's because the output of a previous transaction *becomes* the input of a new one. In the example above, Input 0 is the **output** of the deposit made a day earlier. Because that output has not been used in any other transaction (i.e. it's an u *nspent output*), it can be used as **input** for a new transaction. Once used, it becomes a *spent output* that can't be noted as an input for a future transaction.

Before this particular transaction can be broadcasted to the network, Alice needs to prove that she possesses the Private Key associated with this address by signing the transaction. Alice can sign transactions by loading the text file used to store it, or manually entering her 12-word mnemonic key. Alice's client will then combine all details from the transaction and convert it to hexadecimal format.

After the conversion, Alice's entire transaction looks like this:

```
0100000001148e2d38c3689aad33912d200466fae64f5838f78b3b9f86b01c248720e6c4a6010
00000fdfe0000483045022100c30774a82e9073eddb6087f41a59072d29eaee7c3d1421d6f871
76ebdff4d7d0022062de162d52c6b547fa24691df56c9da4d6eb7ee73af414af4b60ccce9e69e
9e601483045022100f3e2f4a4b970c266a6bd8673cc4ebceea552d8f95a08a35f9961b4d82dfb
b8b1022006bd96c993d8b3abb5f709575a0bd4aaa3dfc7ef4be2332e5742dea07d0cc3a9014c6
9522102645819411857186df087f733675574e37372b4de78471b5c87b832d977f3007e21034d
f63462f237819e46a9aa586a87597fde69df7ed2b5b583de38ae0d4abc183a2103ec10bb20748
527001b900474f72440e1a9591305b991f79cbf0897413ec0cfb753aeffffffff0240420f0000
0000001976a9147fd627956048ff5b5cff26183df231540c637d2e88acd8f185000000000017a
914167a1e9c105bd7bc1a5e86d3d576078a63f0472c8700000000
```

The above string is what's shared with other members of the network. Once broadcasted, the nodes around Alice verify the transaction's validity and disseminate it to other nodes until the transaction gets processed by the miners. As mentioned earlier, every node stores a bucket of unprocessed transactions. This is what's called the *memory pool*, or **mempool**. When the mempool is full and nodes in the network are storing a large number of unprocessed transactions, the network fee rate per kilobyte increases, since memory is scarce. As result, network mining fees also increase.

Accordingly, the Bitcoin mempool is used as a proxy for network congestion and transaction costs, since a high number of unprocessed transactions = less space in the mempool = higher fees.

I must emphasize that bitcoin network fees are priced only on the basis of transaction size (kilobytes), **regardless of the amount being sent**.

THE VERIFICATION OF BLOCKS THROUGH MINING

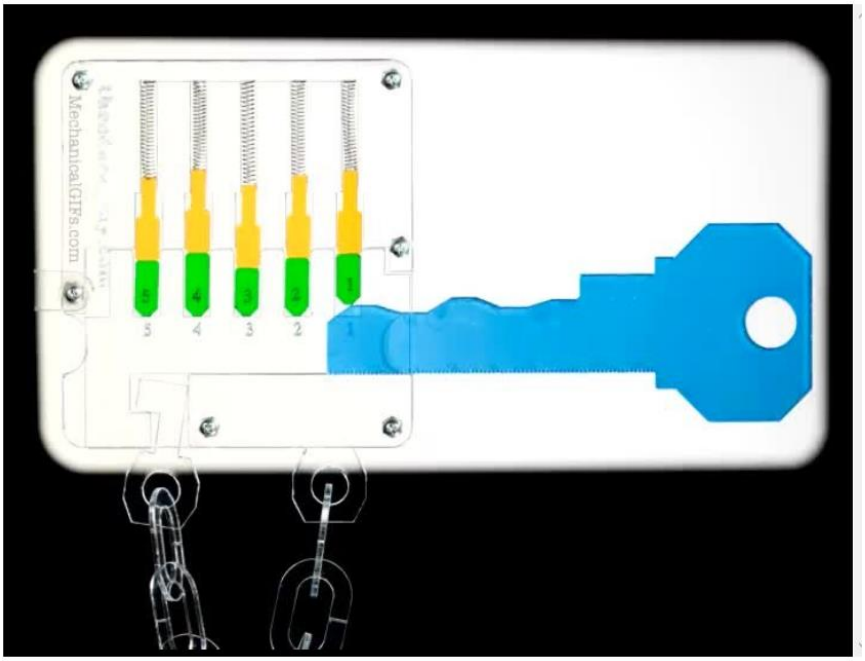
The term *mining* gained popularity because it described a parallel with a real-world activity that is probabilistic in nature, and that gives no guarantees of success.

Gold miners know that, even if they spend significant amounts of resources in the process, there is no guarantee that a precious metal will be found. Although this can be a useful analogy to describe the probabilistic nature of the activity, the term mining does nothing to help understand the process itself, which is one Bitcoin's greatest accomplishments (so far) when it comes to both user adoption and network security.

The actual activity miners perform is more analogous to the job of a real-world *locksmith* than that of a real-world *miner*.

Imagine that, every ten minutes, the Bitcoin protocol gives the locksmiths in its network a closed lock with an unknown key. Only if one of them is able to open the lock and attach a new group of transactions to the chain, a reward is issued. To find the key, the successful locksmith must try millions of different combinations, which takes both time and effort. The locksmith must prove to the competing locksmiths that he went through the hassle of trying different combinations by showing them the real key, which they can then replicate and verify for themselves. Producing the key, which is the proof that work has been done, automatically grants the successful locksmith a reward that is, in practice, proportional to his efforts.

Proof-of-Work, get it?!?



Once a locksmith finds that the coordinates 5–4–3–2–1 opens the lock, this combination becomes the Proof-of-Work. (Source: mechanicalgifs.com)

The protocol is always aware of how many locksmiths are trying to find the key coordinates to its lock so that it takes, on average, ten minutes until a locksmith can find a correct key that opens the lock. If there are only a dozen locksmiths in the network, the lock will be small and can easily be opened using a short key with very few grooves. Conversely, if there are hundreds of locksmiths competing in this challenge, the protocol will give them a large lock that requires a long key with many grooves. With regards to Bitcoin, millions of different combinations need to be tested before the correct key is found.

Consider that when Bitcoin launched, only a simple metal file was required to produce a valid key. Now, this activity can only be done with an ASIC, a piece of hardware that is optimized to compute a specific algorithm. The rise of Bitcoin ASICs is analogous to selling these locksmiths electric metal grinders that exponentially increase their abilities to test different key combinations. As the economic incentives of this activity increased in value, the lock increased in size.

This illustrative Bitcoin lock is now massive, which greatly contributes to the security of the network. Given the competitive and probabilistic nature of the activity itself, it is unlikely that the same locksmith will find the key for consecutive locks over a period of time. Unpredictability is an important feature because it diminishes the likelihood that a malicious actor will double spend funds by mining consecutive blocks. However, the increase in difficulty, when coupled with specialized hardware,

creates barriers of entry for the average user, which ultimately translates into some centralization.

Prominent members of the community are now discussing whether to embrace or scrutinize the use of ASICs. As I stated in this article, I see the confluence of ASIC manufacturer + miner as potentially dangerous. The caveat is it does increase network network security by orders of magnitude (at the expense of decentralization, of course).

PROOF-OF-WORK AND HASH FUNCTIONS

As you may know, the mathematical puzzle referred above is based on a cryptographic hash function. Generally speaking, hash functions map every piece of data in a file, assign identifiers to it, and produce an output of fixed length. In other words, hash functions are used to compress data of any size to a standard output, and that output is called *the* hash. You can put the **entire** Library of Congress through a hash function and compare the output with the hash of the word *it;* **both outputs will have the same size**.

To produce proof-of-work, Bitcoin miners use SHA-256; a hash function originally designed by the NSA for the compression of sensitive information. Before being able to add a block to the blockchain, miners must use the SHA-256 function to combine the data inside the block with a specific, unknown, number. This number is called the **nonce**. Miners need to find a nonce that, when combined with the data within a block, produces a hash that starts with the target number of leading zeroes.

Hash functions, as shown in Figure 5, are highly sensitive to the slightest change. Therefore, miners must try millions of different nonces within that 10-minute period in order to find a nonce that, when combined with a block, begins with the target number of zeroes. If you read the previous section carefully, you will realize that nonce *is* the key the locksmiths are competing to find.

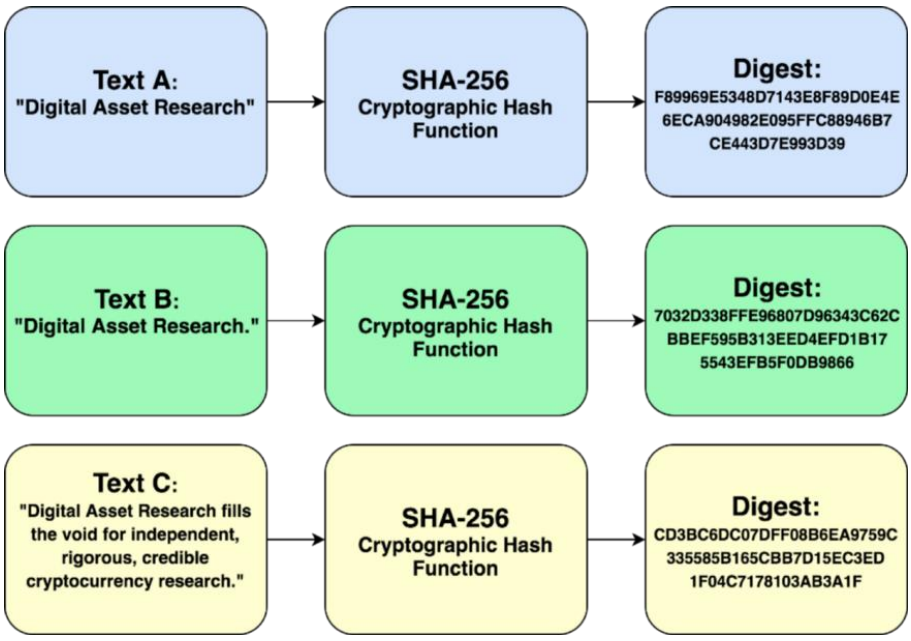


Fig. 5. Example of a Hash Function (Source: Digital Asset Research)

Changing a transaction that has already been added to a block requires changing the **entire** block. Bitcoin works because the data structure of its blockchain makes it immutable; once a block has been mined, nothing in it can be changed without affecting the entire block.

This is achieved using a data structure called a **Merkle Tree,**which basically combines every piece of data within a block so that all data is interdependent. As you can see in Figure 6, Alice's transaction to Bob (Tx1) is hashed to produce Hash1, which is then hashed with Hash0 (the hash of another transaction that happened at the same time) to produce Hash01. The protocol does that to every single unconfirmed transaction that will be added to the block until a **root hash** is found.

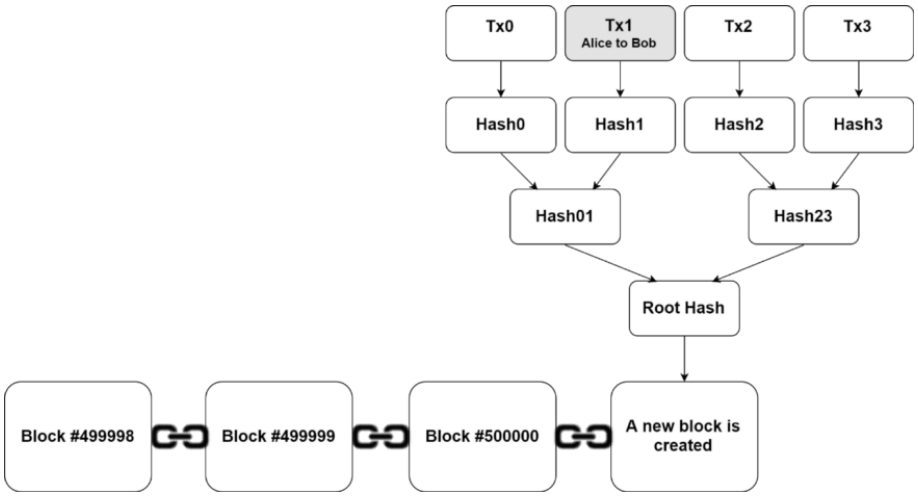


Fig. 6. Example of a Merkle Tree Structure (Source: Digital Asset Research)

The root hash is the hash of all transactions within a block. If the slightest bit of data within any of these transactions is changed, the root hash will be completely different. This is the reason why the term *blockchain* has been highly popularized; the immutability of this data structure is highly desirable for many different applications.

The **block header** referred in the section on light clients is the combination of the root hash with the block header of the previous block in the chain. This is what essentially links blocks together in a blockchain. The block header also includes other data pieces such as the nonce, the current time and how difficult it was to mine it.

7) Finally, Alice's transaction is added to a block along with many other transactions. Once the block gets mined, Bob can download it and verify that the transaction has been confirmed.

INCENTIVE STRUCTURE AND BLOCK REWARDS

The Bitcoin protocol has a native incentive mechanism that rewards miners that are able to produce valid Proof-of-Work. This amount is divided by half every 4 years. At the time of writing, it is 12.5 BTC per block. As stated earlier, miners also receive network transaction fees from every transaction they choose to include in a block.

When the Bitcoin Network launched in January 2009, block rewards were 50 BTC per block. As expected, block rewards halved to 25 BTC in November 2012, and to 12.5 BTC in July 2016. In the year 2140, the Bitcoin protocol will stop generating new coins and mining rewards will only be based on network fees. At that point, around 21 million bitcoins will be in circulation.

Bitcoin's brilliant incentive layer has greatly contributed it its adoption since it provides a unique economic structure based on scarcity. Incentives are defined by an algorithm, which allows Bitcoin's inflation to be precisely modeled. This makes Bitcoin a good candidate for the coin of the future, since its monetary policy is determined by math in the form of software.

FINAL REMARKS

The first time I played around with Bitcoin's codebase was in the Fall of 2012, when I was still in college. I remember being dismissive of it at first, which I now understand was because of ignorance. Bitcoin's simplicity can be very deceiving, and it is easy to overlook its elegance.

Ignorance is the real reason why Bitcoin has been declared dead so many times. This shouldn't come out as a surprise, after all, understanding this mix of Cryptography, Computer Science and Economics requires both time and effort.

But once you *get it*, its technical brilliance is undeniable.

I want to send a huge thanks to Andreas M. Antonopoulos for writing "Mastering Bitcoin." It opened my mind to so many under-appreciated aspects of the Bitcoin protocol. If you don't have a technical background, I highly recommend his latest book, The Internet of Money.

If you find this post useful, please share it on social media and join the battle against crypto-ignorance! Help this post gain more visibility on Medium by clicking the clap button as much as possible. If you are interested in the space, you should also sign up for DAR's free daily newsletter. It's a great way of keeping up with what's happening in cryptoland, and you will be notified when we post things this.

Bitcoin Data Science (Pt. 1): HODL Waves

By Dhruv Bansal

Posted April 17, 2018

This is part 1 of a series

- Bitcoin Data Science (Pt. 1): HODL Waves
- Bitcoin Data Science (Pt. 2): The Geology of Lost Coins
- Bitcoin Data Science (Pt. 3): Dust & Thermodynamics

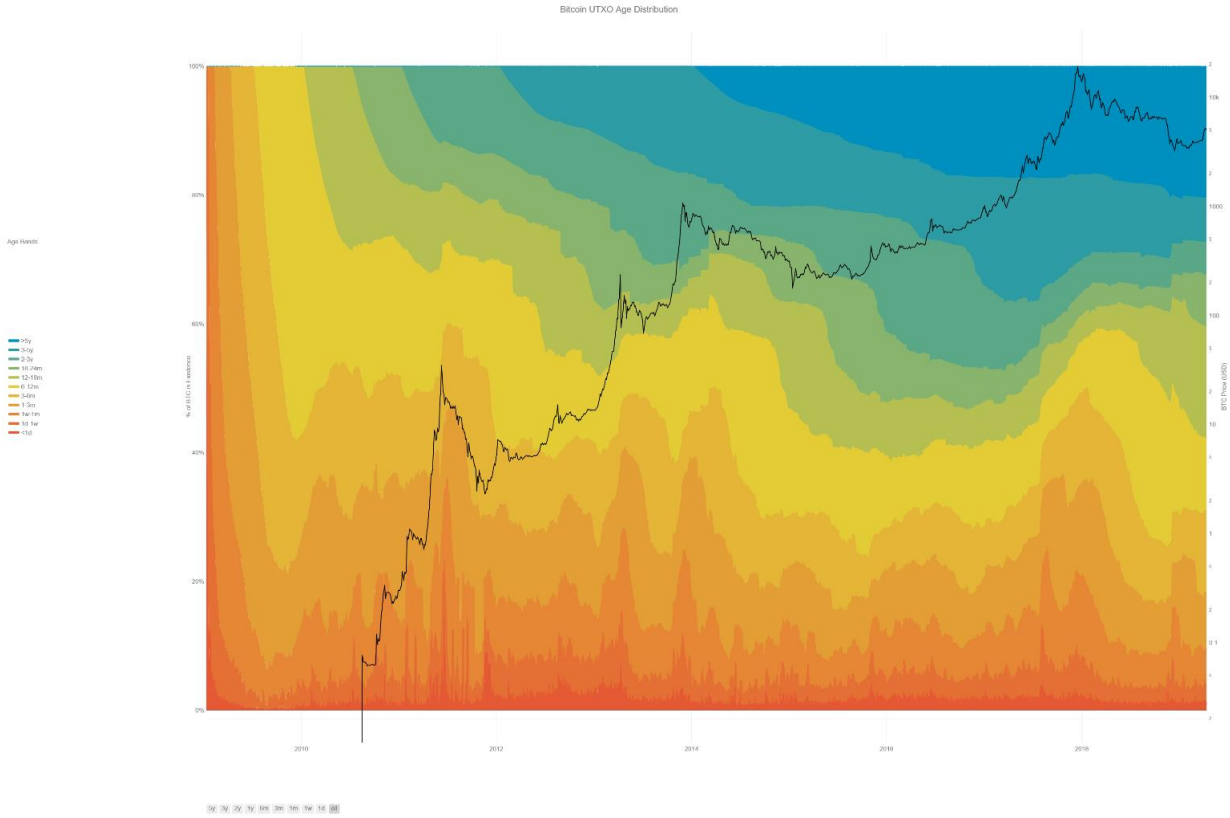
Bitcoin uses a curious accounting structure called a UTXO—an Unspent Transaction Output. All UTXOs are timestamped by the transaction/block in which they were created. Since all bitcoin in existence is contained in some UTXO, this means that all bitcoins have an *age*: **not** the age/time when that bitcoin was *first mined*, but when it was **last used in a transaction**.

Since Bitcoin stores its full transaction history in the blockchain, it is possible to look backwards and analyze the age distribution of UTXOs over time. Unchained Capital first analyzed Bitcoin's UTXO history a few years ago and what we learned encouraged us to start our crypto-lending product. We are now sharing our analyses publicly because we think they are fascinating and informative. Let us know if you agree.

On to the data science!

The Bitcoin UTXO Age Distribution

The following chart displays the age distribution of Bitcoin's UTXO set historically back to the genesis block (Note: this chart does not display correctly on mobile devices.)



The colored bands show the relative fraction of Bitcoin in existence that was last transacted within the time window indicated in the legend. The bottom, warmer colors (reds, oranges) represent Bitcoin transacting very recently while the top, cooler colors (greens, blues) represent Bitcoin that hasn't transacted in a long time. Bitcoin's money supply grew from 50 BTC to ~ 17M BTC over this time period, so the chart has been normalized by the BTC in existence at each date (left y-axis). The black line shows the USD/BTC price (logarithmically, right y-axis). Chart lovingly made by Nelson Morrow based on prior work by @jratcliff [Direct Link]

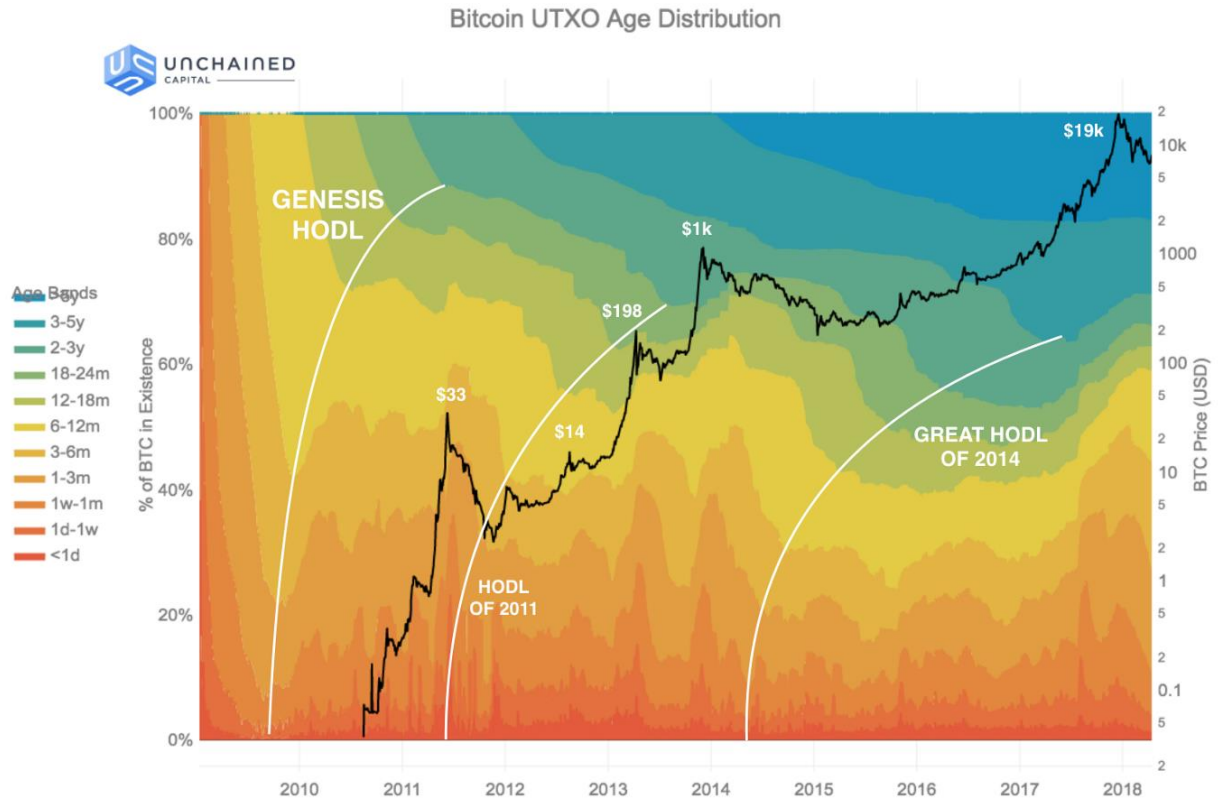
This chart is fascinating because it displays the macroscopic shifts that have occurred in Bitcoin's ownership through history. Spikes in the bottom, warmer-colored age bands (<1 day, 1 day—1 week, 1 week—1 month) indicate large amounts of bitcoin suddenly transacting. The steady growth of the top, coolor-colored age bands (2–3 years, 3–5 years, >5 years) shows bitcoin that's not being transacted with, idling between rallies. The interaction between these two patterns illustrates the behavior of Bitcoin's investors during market cycles.

It is not possible to make charts such as the one above for traditional asset classes. It's only Bitcoin and other public blockchains that meticulously track these data throughout their whole histories. This enables post-hoc analyses of large-scale market behavior.

Introducing: The HODL Wave

A common pattern after every rally in Bitcoin's price is what we have named a "HODL wave." A HODL wave is created when a large amount of Bitcoin transacts on the way up to and through a local price high, becoming recent BTC (1 day—1 week old), and then slowly ages into each later band as its new owners HODL.

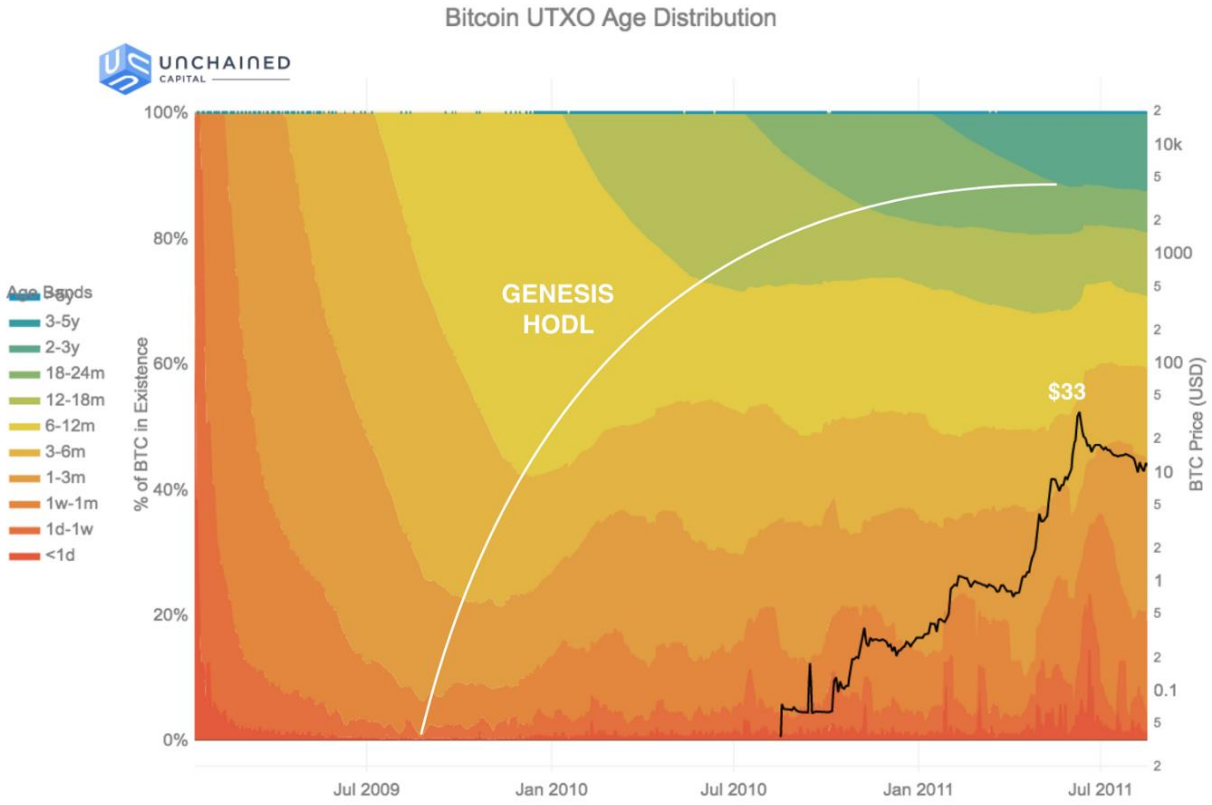
A HODL wave manifests visually on the chart as a pattern of nested curves caused by each age band becoming suddenly much fatter (taller) at progressively later times from the rally. The image below traces a few of the largest HODL waves.



An annotated image of the UTXO age distribution. Local price peaks are labeled. The solid white lines trace "HODL waves"—a pattern of newly recent Bitcoin aging into each subsequent band, indicating that its new owners are HODLing. Only the three largest HODL waves are traced—many smaller HODL waves are also present.

A Short History of HODL Waves

The Genesis HODL: January 2009 — June 2011 ($0 — $33)



The Bitcoin UTXO age distribution zoomed in to a timespan covering the "Genesis HODL" — the first HODL wave in Bitcoin's history.

The first HODL wave — the "Genesis HODL" — was not caused by a price rally because Bitcoin had no price at that time. Instead, it was caused by the initial acquisition of Bitcoin by Satoshi and the other first miners.

During the first year of Bitcoin's history the community was extremely small, transaction volume was low, and there were no exchanges to establish a USD/BTC price. The coins being mined during 2009 weren't included in transactions very often for these reasons. They sat around and progressively accumulated into later age bands.

Consequently, each of the colored age bands appears suddenly in the diagram once sufficient time has passed since the genesis block (e.g. — the green 12–18 month age band appears exactly 12 months after the genesis block). The age band grows for a time, but then begins shrinking as all the existing Bitcoin ages into the next band.

Because there was nowhere to sell Bitcoin at the time, the Genesis HODL is one of the clearest HODL waves on the chart: early miners had no choice but to hold their Bitcoin and surf the wave. Later HODL waves are much frothier as holders could exit into fiat whenever they wanted.

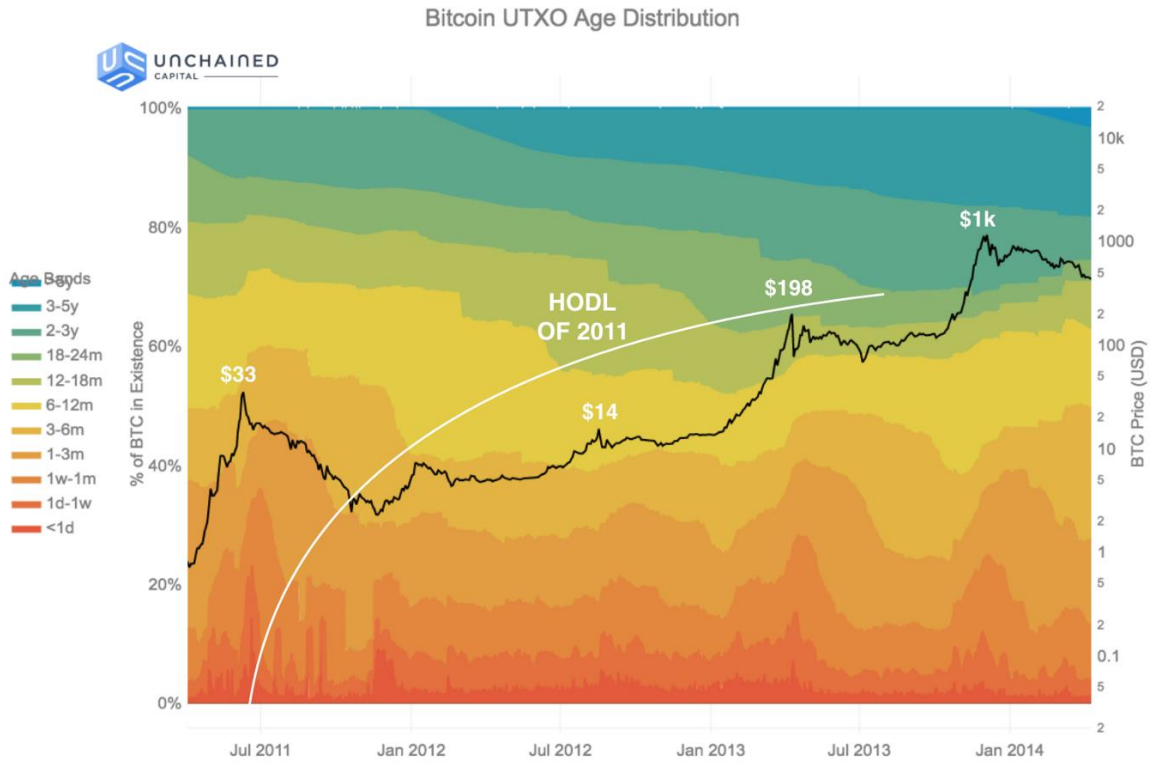
The first time this pattern shifted was in mid-2010, going into 2011. The first Bitcoin exchanges, including Mt. Gox, launched in 2010. Bitstamp, Kraken, Coinbase all launched in 2011. The fraction of coins older than 12 months stopped growing in June 2010 for the first time. This is the first era where holders could trade Bitcoin online. Bitcoin's price wouldn't reach even $1 till February, 2011, but early miners likely had many thousands of BTC. Why not make a little cash?

By April 23rd, 2011, Satoshi had left Bitcoin, right as Bitcoin reached $1. Satoshi is estimated to hold ~ 1M BTC, so he/she/they/it was already a millionaire at this point. Maybe that was enough?

I've moved on to other things.—Satoshi Nakamoto, April 23rd, 2011 (1 BTC = $1).

What a casual bastard. :)

The HODL of 2011: June 2011—December 2013 ($33—$1k)



The Bitcoin UTXO age distribution zoomed in to a timespan covering the "HODL of 2011"—the second major HODL wave in Bitcoin's history.

Starting in June 2011, Bitcoin's price suffered its first major collapse, from $33 all the way down to $2–3 by November 2011. It took almost two years to recover when it rallied to $198 in April, 2013.

During the rally up to $33 in June 2011, all the holders who sold were early miners by definition. No one else could really have acquired BTC to sell.

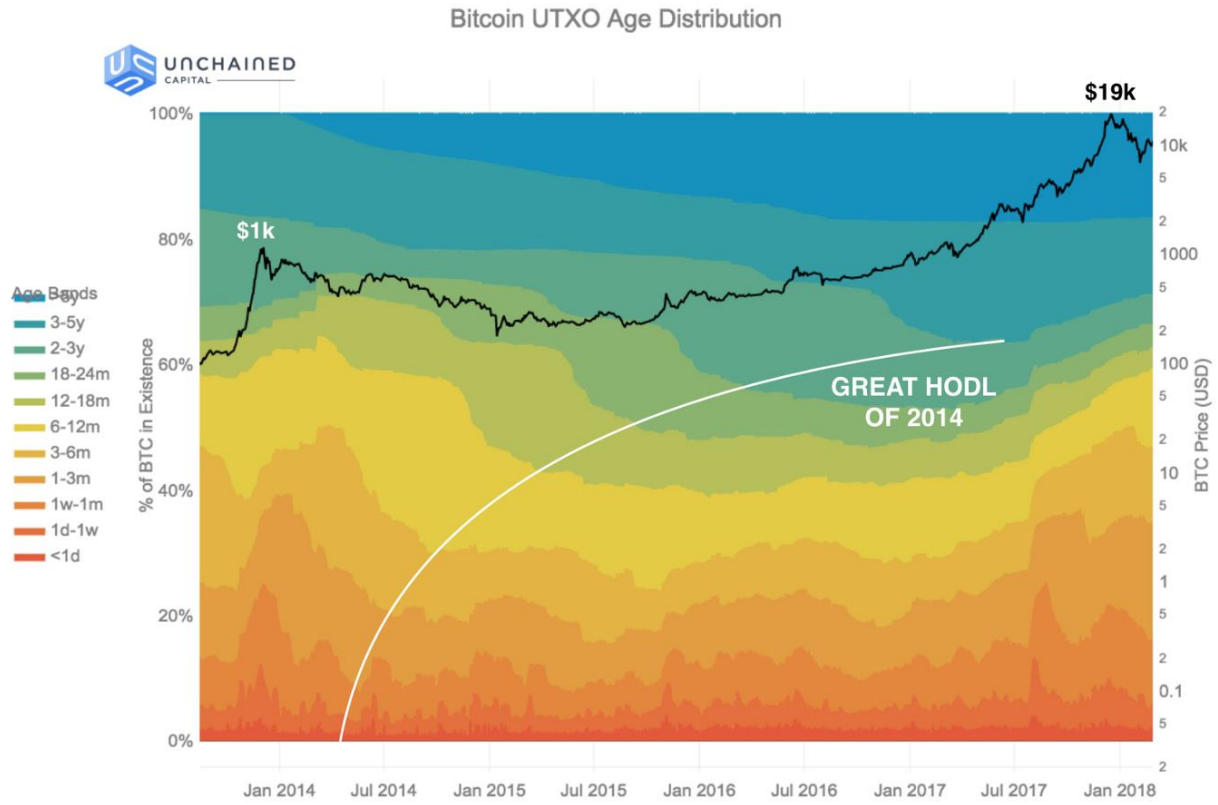
But the rally up to $198 was different. The age bands which shrunk the most leading up to the rally were between 12 months to 24 months. These were likely the first wave of investors—not miners—selling to realize gains. These investors would have acquired their BTC leading into the prior $33 price rally and afterwards.

Bitcoin collapsed again from $198 in April, 2013, down to $69 in July, 2013 only to soar past $1k by December, 2013. There wasn't much time for panic-selling before a new surge of euphoria.

This was the first major rally that was covered in the news. Many major exchanges such as Bitstamp, Kraken, & Coinbase—not to mention Mt. Gox—had been around for a few years and were mature enough to service a large wave of demand for the first time.

Right after the rally to $1k, more than 60% of BTC had been spent within the last 12 months. This was the most "recent" moment for BTC's money supply in history—the moment at which the average last time of use of a Bitcoin was lowest. Who sold? Once more, it was the investors who purchased in the prior 2–3 years, through the $33 peak and the $198 peak.

The Great HODL: December 2013 — December 2017 ($1k — $19k)



The Bitcoin UTXO age distribution zoomed in to a timespan covering the "Great HODL of 2014" — the third major HODL wave in Bitcoin's history.

Ahh, the long winter of Bitcoin. After collapsing in December 2013, Bitcoin wouldn't reach $1k again till February 2017, more than 3 years later.

But the rally to $19k, reached by the end of the year in December 2017, was truly spectacular. There was much more mainstream press coverage and many more new investors, including some extremely "traditional" investors such as institutions and funds.

As the price was crossing $1k in February, 2017, almost 60% of Bitcoin was older than 12 months. A year later, just after the peak at $19k, only 40% of Bitcoin was older than 12 months. During 2017, 20% of Bitcoin in existence was transacted with for the first time in years. Why? We see three separate, related reasons.

Capturing Gains

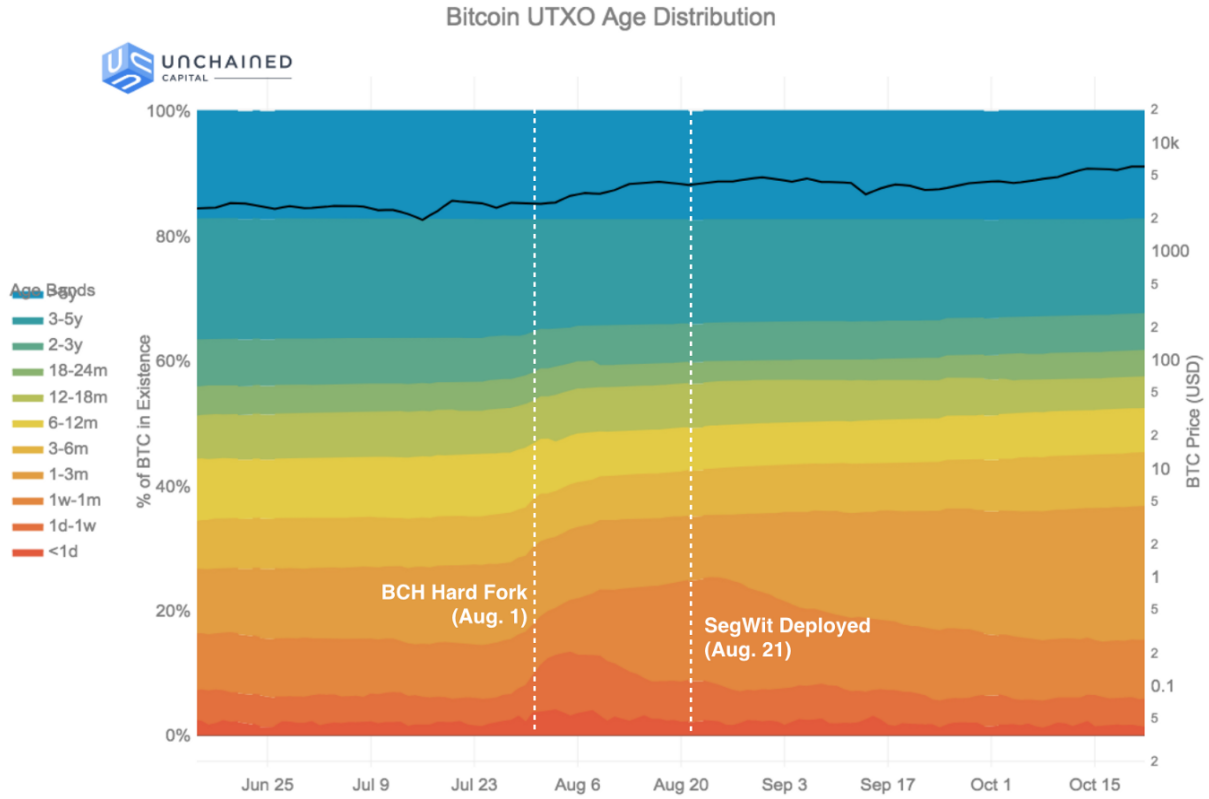
Some of the transaction volume in 2017 was about capturing gains. Investors who'd held BTC for 12 months or more were the sellers, with particular emphasis on those who'd held for 2–5 years. Fully 15% of BTC moved out of those age bands and

became young again during this rally. The selling started almost as soon as BTC crossed $1k again, in February 2017.

ICOs

But this time period also corresponds with the rise in Ethereum, ERC20 tokens, and ICOs. Many ICOs accepted Bitcoin and many Bitcoin holders felt that investing their BTC into ICOs was a way to capture the meteoric rise in value of the ETH-ecosystem, as no similar growth was yet occurring in Bitcoin. So perhaps it was ICO fever that compelled Bitcoin holders into the warm embrace of ETH and ERC20 after enduring many years of frosty Bitcoin winter?

Bitcoin Cash & Segregated Witness



The Bitcoin UTXO age distribution zoomed in to a timespan covering the Bitcoin Cash hard fork (Aug. 1) and the deployment of segregated witness (Aug. 21).

The final factor was the Bitcoin/Bitcoin Cash hard fork of August 1st, 2017, and the subsequent upgrade (on the Bitcoin side) to using segregated witness, on August 21st, 2017. Both these events caused large amounts of Bitcoin to transact for the first time in years as holders acted to claim coins on both sides of the fork and move their Bitcoin to new segwit addresses.

The data clearly shows the significance of this event. In the month of August 2017, 25% of bitcoin became less than one month old. That equates to nearly 4M BTC, or $17B of value at then prices.

The Next HODL: December 2017 onwards ($19k — ?)

Today, after the rise of 2017 and the fall in 2018, the fraction of Bitcoin older than 12 months has dropped to just 40%, making the average age of Bitcoin almost as "recent" today as it was just after the last big rally to $1k.

And after every great rally, there's been a great HODL. As the data shows us, there is already the development of another generation of holders settling in for the long haul. Beginning in January 2018, the category of bitcoin that are 6–12 months old rebounded from a low of 7.76% to 14.63%, a doubling of its population.

It will be interesting to follow this new HODL wave over the next few months and years. What price will be required for the wave to break and a new cycle of gains-taking to occur? How much older will the average bitcoin get before the cycle begins again? How much larger will the next cohort of hodlers be?

If you're curious about the answers, check back on this blog post over time. We will be continuously updating the UTXO age distribution chart above (also available as a direct link).

Many thanks to Taylor Pearson, Kyle Samani, Tushar Jain, هيهات منا الذلة, & Orie Steele for their invaluable feedback when reviewing this post.

This post is the first in a series using data science to tell stories about Bitcoin. It describes the behavior of Bitcoin HODLers during market cycles.

Stay tuned for

- **Part 2:** In which we quantify how much Bitcoin is lost.
- **Part 3:** In which we analyze UTXO dust in the chain.

Unchained Capital has been doing data science on blockchains for years. Discovering the large amount of Bitcoin UTXOs older than 12 months convinced us to start a lending business to help cryptocurrency owners get value from their digital assets today while continuing to hold them into the future. If you are holding BTC (and soon ETH) and you'd like to borrow against your holdings, please sign up for an account on our website and apply for a loan.

Remember: Friends don't let friends sell Bitcoin.

The Long Game in Crypto: Why Decentralization Matters

By Spencer Bogart

Posted April 25, 2018

The future of crypto hinges on a critical, yet-to-be-answered question about the role of decentralization in a blockchain-based network. The answer has tremendous implications for the industry and the way investors allocate capital across crypto-assets.

The question is how important is decentralization and to what degree can we compromise on decentralization for particular use-cases? I will make a case that the only medium- to long-term viable decentralization strategy is one that supports so called "sovereign-grade" censorship resistance.

The trend toward centralization

First, a little background: Decentralization is one of many "features" that a blockchain could offer but it's an "expensive" feature in that blockchains which are willing to compromise on decentralization can offer their users greater throughput and/or range of functionality. As such, most of the newer coins make this tradeoff for improved throughput and/or functionality at the expense of decentralization.

For example, relative to Bitcoin, Ethereum has placed a relatively greater emphasis on functionality. Newer 3.0 smart contract platforms like EOS have gone much further along the spectrum toward centralization — to the point where EOS will ultimately be run by a relatively small handful of entities but can offer a much greater range of functionality and improved throughput as a result.

It's not surprising that new users and developers have gravitated toward these newer networks: Improved throughput and functionality are things that users and developers can immediately appreciate whereas the benefits of "decentralization" as a feature are seemingly amorphous.

The importance of decentralization

The reality, however, is that without decentralization these crypto networks lose their most important qualities of being "permissionless" and "censorship-resistant" — that is, that anyone can use the network and anyone can build on top of them.

After all, the entire point of a decentralized blockchain is to provide a hard-promise — an immutable ledger with open, non-discriminatory participation. In a sense, we bear the inefficiency of decentralization because it is the only way to enable a network with these qualities.

Still, the question remains: how decentralized does a network need to be? And is the trend toward centralization sustainable for networks that aim to be "permissionless" (like the internet)? The problem is that we don't yet know what level of decentralization is safe and, to make matters more challenging, decentralization itself is multi-dimensional and difficult to measure.

Censorship resistance: "sovereign-grade" vs. "platform-grade"

Many have suggested that the level of decentralization necessary is dependent on the use-case — and that there are two broad categories: Those that need "sovereign-grade" censorship resistance and those that only need "platform-grade" censorship resistance.

The former ("sovereign-grade") typically refers to something like Bitcoin, which is perceived to be a greater target for nation-state attackers than 3.0 smart contract platforms (e.g. Tezos, EOS) which might only need "platform-grade" censorship resistance. The idea here is that the nature of Bitcoin makes it more likely to be subject to nation-state attackers than a smart contract platform.

The argument for "platform-grade" censorship resistance is that there's tremendous uncertainty building on today's centralized platforms like Facebook, Apple, or Google in that they can suddenly change what is and isn't allowed on their platform and radically affect the economic prospects of businesses that depend on them. As a result of this uncertainty, developers are more hesitant to build on these platforms and investors are more hesitant to invest in companies that depend on them — a net loss for development and economic activity.

In this "platform-grade" narrative the "3.0" smart contract platforms solve this problem by distributing control from one authority to a relatively small handful of authorities. The narrative continues that by doing so these platforms are valuable in that they can provide **stronger assurances** to developers which will drive more activity and development on the platform. This assertion of "stronger assurances" is core to the "platform grade" narrative.

At the highest level, my concern with this narrative is that these platforms simply *can't* offer meaningfully stronger assurances without being highly decentralized —

that it's only through high degrees of decentralization that we can offer strong assurances.

More specifically, my issues with this "platform-grade" narrative are two-fold: First, permissionless platforms will inevitably demand sovereign-grade censorship resistance and, second, if not truly permissionless then these platforms will trend toward the same outcome as today's centralized platforms (censorship and permissioning) but with less efficient infrastructure.

Issue 1 of 2: Permissionless platforms require sovereign-grade censorship resistance

If these semi-decentralized platforms deliver truly "permissionless" functionality (anyone can build anything), it will only be a matter of time before someone builds a DApp that draws the ire of "sovereign-grade" nation-state attackers. For example, someone will build the "money laundering DApp" or a "classified documents" DApp that allows people to buy and sell nation-state "secrets".

If the platform censors these activities then it has failed to deliver strong assurances and it is neither "permissionless" nor "censorship resistant". Still, some might argue that's a positive: That they wouldn't want to support or build atop a platform that enables these types of activities and therefore the lack of censorship resistance or strong assurances for these use cases is actually a feature that allows "us" to eradicate "bad things".

This brings me to my second issue with the concept of "platform-grade" censorship resistance.

Issue 2 of 2: Subject to the same pressures, these platforms will trend toward the same outcome as today's centralized platforms (but via less efficient means)

These semi-decentralized platforms are subject to the same s **ocial and economic pressures**that motivate centralized platforms to censor certain users and activities and therefore will trend toward the same outcome they're supposed to correct. Even worse, they will do so via less efficient means than their centralized counterparts.

Let's take a step back to examine the problem that "platform-grade" censorship resistant platforms are trying to solve: Tech platforms like Facebook, Google, Twitter, and Apple change their policies ("censor") in response to either **social pressure** (e.g. a user-demanded ban on gun videos) or **economic pressure** (e.g. someone using the platform to compete against it).

I'd argue that these social and economic pressures will drive the same outcome whether control is in the hands of a single entity (e.g. Facebook or Apple) or in the hands of a small handful of operators in a semi-decentralized system. If so, then

these platforms haven't achieved "platform-grade" censorship-resistance, haven't delivered stronger assurances than the platforms they came to replace and, instead, have only produced a less efficient means for accomplishing the same ends.

To summarize: Either these platforms will offer strong assurances ("permissionless-ness"), in which case they will attract "sovereign-grade" attackers (and "platform-grade" censorship resistance will be insufficient) OR they will embrace censorship and permission-ing, in which case they will end up as less efficient varieties of today's centralized platforms. Regardless, neither path appears sustainable.

The path forward: Highly decentralized base layer with increased centralization (and efficiency) on higher layers

Why does all of this matter? With crypto valued over $400B and new capital inflows daily, the question of decentralization is important as developers choose which platforms to build on top of for long-term viability and investors allocate capital to the industry.

I've made the case as to why I think highly decentralized blockchains with "sovereign-grade" censorship resistance might be the only viable strategy in the medium- to long-term but what does this mean going forward?

What these "3.0" platforms are implicitly acknowledging in their compromise on decentralization is that there is efficiency in centralization. I think that's an important point and if we can safely leverage the efficiencies of centralization, we should.

However, as explained above, centralization at the base-layer appears unsustainable and the only viable way to make this tradeoff might be on layers built atop a highly-decentralized network like Bitcoin or others. In this way, we can leverage the strong assurances ("hard promises") that highly decentralized, truly censorship resistant blockchains offer while also leveraging the efficiencies of centralization at higher layers.

For a more in-depth discussion about the value of hard-promises in a highly-decentralized network and how we can build "soft-promises" and centralization in higher layers, I highly recommend this talk by Andreas Antonopoulos from the San Francisco Bitcoin meetup.

Conclusion: "Sovereign-grade" censorship resistance is critical

Ultimately, I think the ongoing trend toward increasing centralization inevitably leads to a situation where a blockchain loses its entire raison d'etre as a permissionless platform with strong assurances and what we're left with is a permissioned network that resembles today's centralized networks but built on less efficient infrastructure. That doesn't sound very interesting or exciting to me.

Instead, I'm most optimistic that highly decentralized networks will provide the robust foundation on top of which we can realize the efficiencies of centralization in higher layers — should it be desired. It's a path that will likely take longer and be more difficult to build, but it might be the only viable route medium- to long-term.

In closing, we're all living through one of the grandest experiments in history and it's playing out in real time, I'd love to hear your feedback about why I'm wrong in the viewpoint presented here. Criticism encouraged!

Disclaimer:

WORDS

Please note that this Journal is provided on the basis that the person who is reading it accepts the following conditions relating to the provision of the same (including on behalf of their respective organization). This Journal does not contain or purport to be, financial promotion(s) of any kind.

This Journal does not contain reference to any of the investment products or services currently offered by the operator of the journal, that means any business I am associated with. Bitcoin, shitcoins, and related technologies can be volatile. Don't buy what you can't afford to lose and please do your own research.

Bitcoin has paved the way for some VERY radical technology AND it's very confusing. Read more. Ask questions. The purpose of this Journal is to provide archive and curate the best commentary and culture in the bitcoin space.

Nothing within this Journal constitutes investment, legal, tax or other advice. This Journal should not be used as the basis for any investment decisions which a reader may be considering. Any potential investor in bitcoin or shitcoins, even if experienced and affluent, is strongly recommended to seek independent financial advice upon the merits of the same in the context of their own unique circumstances.

Share this journal early and often. Engage the authors and tell them what you think. We sharpen our position through discourse and debate.

DYOR | BTFD | HODL

Thanks for your attention and support. I appreciate your feedback and hope you enjoy this publication.

- @_joerodgers