How the Bitcoin protocol actually works

by Michael Nielsen on December 6, 2013

Many thousands of articles have been written purporting to explain Bitcoin peer-to-peer currency. Most of those articles give a hand-wavy account of underlying cryptographic protocol, omitting many details. Even those articl delve deeper often gloss over crucial points. My aim in this post is to expla ideas behind the Bitcoin protocol in a clear, easily comprehensible way. W from first principles, build up to a broad theoretical understanding of how tl works, and then dig down into the nitty-gritty, examining the raw data in a l transaction.

Understanding the protocol in this detailed way is hard work. It is tempting take Bitcoin as given, and to engage in speculation about how to get rich v whether Bitcoin is a bubble, whether Bitcoin might one day mean the end and so on. That's fun, but severely limits your understanding. Understandi details of the Bitcoin protocol opens up otherwise inaccessible vistas. In pa the basis for understanding Bitcoin's built-in scripting language, which mak possible to use Bitcoin to create new types of financial instruments, such a **contracts**. New financial instruments can, in turn, be used to create new r to enable new forms of collective human behaviour. Talk about fun!

I'll describe Bitcoin scripting and concepts such as smart contracts in futur
This post concentrates on explaining the nuts-and-bolts of the Bitcoin prot
understand the post, you need to be comfortable with **public key cryptog**
with the closely related idea of **digital signatures**. I'll also assume you're
cryptographic hashing. None of this is especially difficult. The basic idea
taught in freshman university mathematics or computer science classes. T
are beautiful, so if you're not familiar with them, I recommend taking a few
familiar.

It may seem surprising that Bitcoin's basis is cryptography. Isn't Bitcoin a c
a way of sending secret messages? In fact, the problems Bitcoin needs to
largely about securing transactions — making sure people can't steal from
another, or impersonate one another, and so on. In the world of atoms we
security with devices such as locks, safes, signatures, and bank vaults. In
bits we achieve this kind of security with cryptography. And that's why Bitc
heart a cryptographic protocol.

My strategy in the post is to build Bitcoin up in stages. I'll begin by explaini
simple digital currency, based on ideas that are almost obvious. We'll call t
currency *Infocoin*, to distinguish it from Bitcoin. Of course, our first version
will have many deficiencies, and so we'll go through several iterations of In
each iteration introducing just one or two simple new ideas. After several s
iterations, we'll arrive at the full Bitcoin protocol. We will have reinvented B

This strategy is slower than if I explained the entire Bitcoin protocol in one
while you can understand the mechanics of Bitcoin through such a one-sh
explanation, it would be difficult to understand *why* Bitcoin is designed the
The advantage of the slower iterative explanation is that it gives us a much
understanding of each element of Bitcoin.

Finally, I should mention that I'm a relative newcomer to Bitcoin. I've been
loosely since 2011 (and cryptocurrencies since the late 1990s), but only go
into the details of the Bitcoin protocol earlier this year. So I'd certainly appr
corrections of any misapprehensions on my part. Also in the post I've inclu
number of "problems for the author" – notes to myself about questions tha

during the writing. You may find these interesting, but you can also skip th
without losing track of the main text.

First steps: a signed letter of intent

So how can we design a digital currency?

On the face of it, a digital currency sounds impossible. Suppose some per
call her Alice – has some digital money which she wants to spend. If Alice
string of bits as money, how can we prevent her from using the same bit st
and over, thus minting an infinite supply of money? Or, if we can somehow
problem, how can we prevent someone else forging such a string of bits, a
that to steal from Alice?

These are just two of the many problems that must be overcome in order t
information as money.

As a first version of Infocoin, let's find a way that Alice can use a string of I
(very primitive and incomplete) form of money, in a way that gives her at le
protection against forgery. Suppose Alice wants to give another person, Bo
infocoin. To do this, Alice writes down the message "I, Alice, am giving Bol
infocoin". She then digitally signs the message using a private cryptograph
announces the signed string of bits to the entire world.

(By the way, I'm using capitalized "Infocoin" to refer to the protocol and ge
concept, and lowercase "infocoin" to refer to specific denominations of the
similar useage is common, though not universal, in the Bitcoin world.)

This isn't terribly impressive as a prototype digital currency! But it does ha
virtues. Anyone in the world (including Bob) can use Alice's public key to v
Alice really was the person who signed the message "I, Alice, am giving B
infocoin". No-one else could have created that bit string, and so Alice can't
around and say "No, I didn't mean to give Bob an infocoin". So the protocc
establishes that Alice truly intends to give Bob one infocoin. The same fac
else could compose such a signed message – also gives Alice some limite
from forgery. Of course, *after* Alice has published her message it's possibl

people to duplicate the message, so in that sense forgery is possible. But possible from scratch. These two properties – establishment of intent on A and the limited protection from forgery – are genuinely notable features of protocol.

I haven't (quite) said exactly what digital money *is* in this protocol. To make explicit: it's just the message itself, i.e., the string of bits representing the c signed message "I, Alice, am giving Bob one infocoin". Later protocols will in that all our forms of digital money will be just more and more elaborate r [1].

Using serial numbers to make coins uniquely identifiable

A problem with the first version of Infocoin is that Alice could keep sending same signed message over and over. Suppose Bob receives ten copies of message "I, Alice, am giving Bob one infocoin". Does that mean Alice sent *different* infocoins? Was her message accidentally duplicated? Perhaps sh trying to trick Bob into believing that she had given him ten different infoco the message only proves to the world that she intends to transfer one infoc

What we'd like is a way of making infocoins unique. They need a label or s number. Alice would sign the message "I, Alice, am giving Bob one infocoi serial number 8740348". Then, later, Alice could sign the message "I, Alice Bob one infocoin, with serial number 8770431", and Bob (and everyone el know that a different infocoin was being transferred.

To make this scheme work we need a trusted source of serial numbers for infocoins. One way to create such a source is to introduce a *bank*. This ba provide serial numbers for infocoins, keep track of who has which infocoin that transactions really are legitimate,

In more detail, let's suppose Alice goes into the bank, and says "I want to one infocoin from my account". The bank reduces her account balance by infocoin, and assigns her a new, never-before used serial number, let's say Then, when Alice wants to transfer her infocoin to Bob, she signs the mess Alice, am giving Bob one infocoin, with serial number 1234567". But Bob c

accept the infocoin. Instead, he contacts the bank, and verifies that: (a) the with that serial number belongs to Alice; and (b) Alice hasn't already spent infocoin. If both those things are true, then Bob tells the bank he wants to infocoin, and the bank updates their records to show that the infocoin with number is now in Bob's possession, and no longer belongs to Alice.

Making everyone collectively the bank

This last solution looks pretty promising. However, it turns out that we can something much more ambitious. We can eliminate the bank entirely from protocol. This changes the nature of the currency considerably. It means th no longer any single organization in charge of the currency. And when you the enormous power a central bank has – control over the money supply – pretty huge change.

The idea is to make it so *everyone* (collectively) is the bank. In particular, v that everyone using Infocoin keeps a complete record of which infocoins b which person. You can think of this as a shared public ledger showing all I transactions. We'll call this ledger the *block chain*, since that's what the co record will be called in Bitcoin, once we get to it.

Now, suppose Alice wants to transfer an infocoin to Bob. She signs the me Alice, am giving Bob one infocoin, with serial number 1234567", and gives message to Bob. Bob can use his copy of the block chain to check that, in infocoin is Alice's to give. If that checks out then he broadcasts both Alice' and his acceptance of the transaction to the entire network, and everyone their copy of the block chain.

We still have the "where do serial number come from" problem, but that tu pretty easy to solve, and so I will defer it to later, in the discussion of Bitco challenging problem is that this protocol allows Alice to cheat by double sp infocoin. She sends the signed message "I, Alice, am giving Bob one infoc serial number 1234567″ to Bob, and the message"I, Alice, am giving Char infocoin, with [the same] serial number 1234567" to Charlie. Both Bob and their copy of the block chain to verify that the infocoin is Alice's to spend. F

they do this verification at nearly the same time (before they've had a char from one another), both will find that, yes, the block chain shows the coin b Alice. And so they will both accept the transaction, and also broadcast the acceptance of the transaction. Now there's a problem. How should other p update their block chains? There may be no easy way to achieve a consis ledger of transactions. And even if everyone can agree on a consistent wa their block chains, there is still the problem that either Bob or Charlie will b

At first glance double spending seems difficult for Alice to pull off. After all, sends the message first to Bob, then Bob can verify the message, and tell else in the network (including Charlie) to update their block chain. Once th happened, Charlie would no longer be fooled by Alice. So there is most lik brief period of time in which Alice can double spend. However, it's obvious undesirable to have any such a period of time. Worse, there are technique could use to make that period longer. She could, for example, use network analysis to find times when Bob and Charlie are likely to have a lot of later communication. Or perhaps she could do something to deliberately disrup communications. If she can slow communication even a little that makes h double spending much easier.

How can we address the problem of double spending? The obvious soluti when Alice sends Bob an infocoin, Bob shouldn't try to verify the transacti Rather, he should broadcast the possible transaction to the entire network users, and ask them to help determine whether the transaction is legitimat collectively decide that the transaction is okay, then Bob can accept the in everyone will update their block chain. This type of protocol can help preve spending, since if Alice tries to spend her infocoin with both Bob and Char people on the network will notice, and network users will tell both Bob and there is a problem with the transaction, and the transaction shouldn't go th

In more detail, let's suppose Alice wants to give Bob an infocoin. As before the message "I, Alice, am giving Bob one infocoin, with serial number 123 gives the signed message to Bob. Also as before, Bob does a sanity chec copy of the block chain to check that, indeed, the coin currently belongs to at that point the protocol is modified. Bob doesn't just go ahead and accep

transaction. Instead, he broadcasts Alice's message to the entire network. members of the network check to see whether Alice owns that infocoin. If broadcast the message "Yes, Alice owns infocoin 1234567, it can now be to Bob." Once enough people have broadcast that message, everyone up block chain to show that infocoin 1234567 now belongs to Bob, and the tra complete.

This protocol has many imprecise elements at present. For instance, what mean to say "once enough people have broadcast that message"? What e "enough" mean here? It can't mean everyone in the network, since we dor know who is on the Infocoin network. For the same reason, it can't mean s fraction of users in the network. We won't try to make these ideas precise Instead, in the next section I'll point out a serious problem with the approa described. Fixing that problem will at the same time have the pleasant side making the ideas above much more precise.

Proof-of-work

Suppose Alice wants to double spend in the network-based protocol I just She could do this by taking over the Infocoin network. Let's suppose she u automated system to set up a large number of separate identities, let's say on the Infocoin network. As before, she tries to double spend the same inf both Bob and Charlie. But when Bob and Charlie ask the network to valida respective transactions, Alice's sock puppet identities swamp the network, to Bob that they've validated his transaction, and to Charlie that they've va transaction, possibly fooling one or both into accepting the transaction.

There's a clever way of avoiding this problem, using an idea known as *pro* The idea is counterintuitive and involves a combination of two ideas: (1) to make it *computationally costly* for network users to validate transactions; a *reward* them for trying to help validate transactions. The reward is used sc on the network will try to help validate transactions, even though that's nov made a computationally costly process. The benefit of making it costly to v transactions is that validation can no longer be influenced by the number c identities someone controls, but only by the total computational power they

to bear on validation. As we'll see, with some clever design we can make i cheater would need enormous computational resources to cheat, making i impractical.

That's the gist of proof-of-work. But to really understand proof-of-work, we through the details.

Suppose Alice broadcasts to the network the news that "I, Alice, am giving infocoin, with serial number 1234567".

As other people on the network hear that message, each adds it to a queu pending transactions that they've been told about, but which haven't yet b approved by the network. For instance, another network user named Davi have the following queue of pending transactions:

I, Tom, am giving Sue one infocoin, with serial number 1201174.

I, Sydney, am giving Cynthia one infocoin, with serial number 1295618.

I, Alice, am giving Bob one infocoin, with serial number 1234567.

David checks his copy of the block chain, and can see that each transactio He would like to help out by broadcasting news of that validity to the entire

However, before doing that, as part of the validation protocol David is requ a hard computational puzzle – the proof-of-work. Without the solution to th the rest of the network won't accept his validation of the transaction.

What puzzle does David need to solve? To explain that, let h be a fixed ha known by everyone in the network – it's built into the protocol. Bitcoin uses known **SHA-256** hash function, but any cryptographically secure hash fun Let's give David's queue of pending transactions a label, l, just so it's got a can refer to. Suppose David appends a number x (called the *nonce*) to l a the combination. For example, if we use l = "Hello, world!" (obviously this of transactions, just a string used for illustrative purposes) and the nonce x (output is in hexadecimal)

```
h("Hello, world!0") =
  1312af178c253f84028d480a6adc1e25e81caa44c749ec81976192e2ec934c6
```

The puzzle David has to solve – the proof-of-work – is to find a nonce x su when we append x to l and hash the combination the output hash begins run of zeroes. The puzzle can be made more or less difficult by varying the zeroes required to solve the puzzle. A relatively simple proof-of-work puzz require just three or four zeroes at the start of the hash, while a more diffic work puzzle might require a much longer run of zeros, say 15 consecutive either case, the above attempt to find a suitable nonce, with $x = 0$, is a fa the output doesn't begin with any zeroes at all. Trying $x = 1$ doesn't work

```
h("Hello, world!1") =
  e9afc424b79e4f6ab42d99c81156d3a17228d6e1eef4139be78e948a9332a7d8
```

We can keep trying different values for the nonce, $x = 2, 3, \ldots$. Finally, a $x = 4250$ we obtain:

```
h("Hello, world!4250") =
  0000c3af42fc31103f1fdc0151fa747ff87349a4714df7cc52ea464e12dcd4e9
```

This nonce gives us a string of four zeroes at the beginning of the output c This will be enough to solve a simple proof-of-work puzzle, but not enough more difficult proof-of-work puzzle.

What makes this puzzle hard to solve is the fact that the output from a cryp hash function behaves like a random number: change the input even a tiny output from the hash function changes completely, in a way that's hard to p we want the output hash value to begin with 10 zeroes, say, then David wi average, to try $16^{10} \approx 10^{12}$ different values for x before he finds a suitab That's a pretty challenging task, requiring lots of computational power.

Obviously, it's possible to make this puzzle more or less difficult to solve by more or fewer zeroes in the output from the hash function. In fact, the Bitc

gets quite a fine level of control over the difficulty of the puzzle, by using a variation on the proof-of-work puzzle described above. Instead of requiring zeroes, the Bitcoin proof-of-work puzzle requires the hash of a block's hea lower than or equal to a number known as the **target**. This target is autom adjusted to ensure that a Bitcoin block takes, on average, about ten minut validate.

(In practice there is a sizeable randomness in how long it takes to validate sometimes a new block is validated in just a minute or two, other times it n minutes or even longer. It's straightforward to modify the Bitcoin protocol s time to validation is much more sharply peaked around ten minutes. Instea a single puzzle, we can require that multiple puzzles be solved; with some design it is possible to considerably reduce the variance in the time to vali of transactions.)

Alright, let's suppose David is lucky and finds a suitable nonce, x. Celebra be rewarded for finding the nonce, as described below). He broadcasts the transactions he's approving to the network, together with the value for x. C participants in the Infocoin network can verify that x is a valid solution to th work puzzle. And they then update their block chains to include the new bl transactions.

For the proof-of-work idea to have any chance of succeeding, network use incentive to help validate transactions. Without such an incentive, they hav to expend valuable computational power, merely to help validate other pec transactions. And if network users are not willing to expend that power, the system won't work. The solution to this problem is to reward people who h transactions. In particular, suppose we reward whoever successfully valida of transactions by crediting them with some infocoins. Provided the infocoi large enough that will give them an incentive to participate in validation.

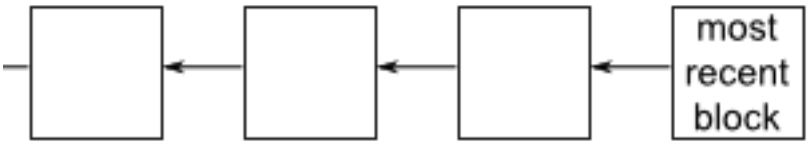
In the Bitcoin protocol, this validation process is called *mining*. For each bl transactions validated, the successful miner receives a bitcoin reward. Init was set to be a 50 bitcoin reward. But for every 210,000 validated blocks (once every four years) the reward halves. This has happened just once, to

so the current reward for mining a block is 25 bitcoins. This halving in the r
continue every four years until the year 2140 CE. At that point, the reward
will drop below 10^{-8} bitcoins per block. 10^{-8} bitcoins is actually the minim
Bitcoin, and is known as a *satoshi*. So in 2140 CE the total supply of bitcoi
cease to increase. However, that won't eliminate the incentive to help valic
transactions. Bitcoin also makes it possible to set aside some currency in
transaction as a *transaction fee*, which goes to the miner who helps valida
early days of Bitcoin transaction fees were mostly set to zero, but as Bitco
gained in popularity, transaction fees have gradually risen, and are now a
additional incentive on top of the 25 bitcoin reward for mining a block.

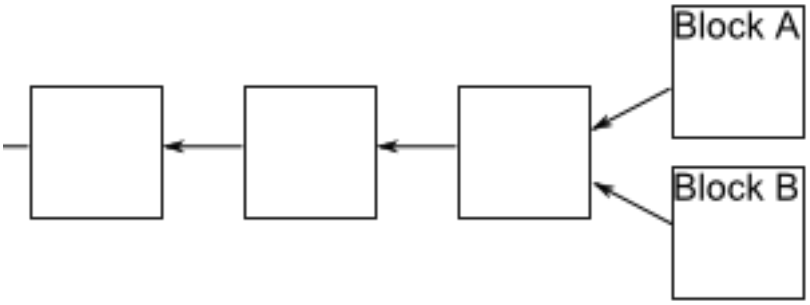
You can think of proof-of-work as a competition to approve transactions. E
the competition costs a little bit of computing power. A miner's chance of w
competition is (roughly, and with some caveats) equal to the proportion of
computing power that they control. So, for instance, if a miner controls one
the computing power being used to validate Bitcoin transactions, then they
roughly a one percent chance of winning the competition. So provided a lo
computing power is being brought to bear on the competition, a dishonest
likely to have only a relatively small chance to corrupt the validation proces
they expend a huge amount of computing resources.

Of course, while it's encouraging that a dishonest party has only a relative
chance to corrupt the block chain, that's not enough to give us confidence
currency. In particular, we haven't yet conclusively addressed the issue of
spending.

I'll analyse double spending shortly. Before doing that, I want to fill in an im
detail in the description of Infocoin. We'd ideally like the Infocoin network t
upon the *order* in which transactions have occurred. If we don't have such
then at any given moment it may not be clear who owns which infocoins. T
this we'll require that new blocks always include a pointer to the last block
the chain, in addition to the list of transactions in the block. (The pointer is
a hash of the previous block). So typically the block chain is just a linear ch
blocks of transactions, one after the other, with later blocks each containin
to the immediately prior block:

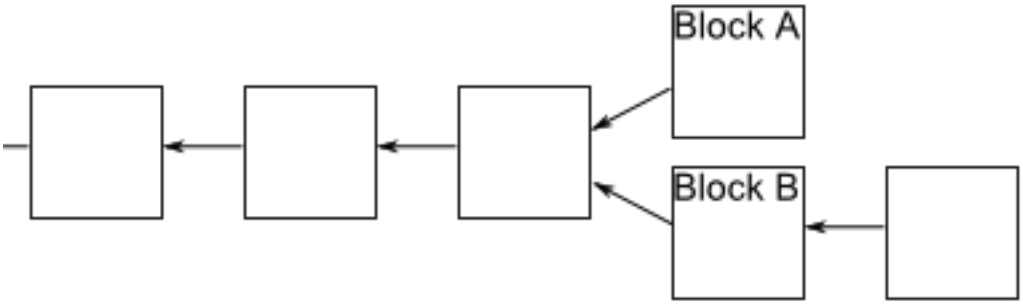


Occasionally, a fork will appear in the block chain. This can happen, for ins
chance two miners happen to validate a block of transactions near-simulta
both broadcast their newly-validated block out to the network, and some p
update their block chain one way, and others update their block chain the



This causes exactly the problem we're trying to avoid – it's no longer clear
order transactions have occurred, and it may not be clear who owns which
Fortunately, there's a simple idea that can be used to remove any forks. Th
this: if a fork occurs, people on the network keep track of both forks. But a
time, miners only work to extend whichever fork is longest in their copy of
chain.

Suppose, for example, that we have a fork in which some miners receive b
and some miners receive block B first. Those miners who receive block A
continue mining along that fork, while the others will mine along fork B. Le
that the miners working on fork B are the next to successfully mine a block



After they receive news that this has happened, the miners working on for
notice that fork B is now longer, and will switch to working on that fork. Pre
order work on fork A will cease, and everyone will be working on the same
chain, and block A can be ignored. Of course, any still-pending transactior
still be pending in the queues of the miners working on fork B, and so all tr
will eventually be validated.

Likewise, it may be that the miners working on fork A are the first to extenc
In that case work on fork B will quickly cease, and again we have a single

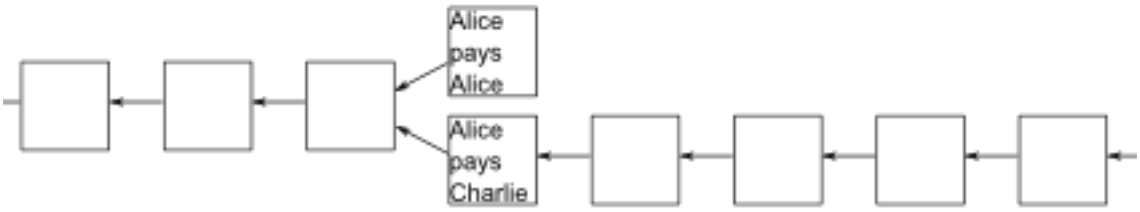
No matter what the outcome, this process ensures that the block chain has
upon time ordering of the blocks. In Bitcoin proper, a transaction is not cor
confirmed until: (1) it is part of a block in the longest fork, and (2) at least 5
follow it in the longest fork. In this case we say that the transaction has “6
confirmations”. This gives the network time to come to an agreed-upon the
the blocks. We’ll also use this strategy for Infocoin.

With the time-ordering now understood, let’s return to think about what hap
dishonest party tries to double spend. Suppose Alice tries to double spenc
and Charlie. One possible approach is for her to try to validate a block that
both transactions. Assuming she has one percent of the computing power,
occasionally get lucky and validate the block by solving the proof-of-work.
Unfortunately for Alice, the double spending will be immediately spotted by
people in the Infocoin network and rejected, despite solving the proof-of-w
problem. So that’s not something we need to worry about.

A more serious problem occurs if she broadcasts two separate transactior
she spends the same infocoin with Bob and Charlie, respectively. She mig
example, broadcast one transaction to a subset of the miners, and the oth
transaction to another set of miners, hoping to get both transactions valida
way. Fortunately, in this case, as we’ve seen, the network will eventually c
of these transactions, but not both. So, for instance, Bob’s transaction mig
be confirmed, in which case Bob can go ahead confidently. Meanwhile, Ch
see that his transaction has not been confirmed, and so will decline Alice’s

this isn't a problem either. In fact, knowing that this will be the case, there reason for Alice to try this in the first place.

An important variant on double spending is if Alice = Bob, i.e., Alice tries tc coin with Charlie which she is also "spending" with herself (i.e., giving bacl This sounds like it ought to be easy to detect and deal with, but, of course, a network to set up multiple identities associated with the same person or organization, so this possibility needs to be considered. In this case, Alice' to wait until Charlie accepts the infocoin, which happens after the transact been confirmed 6 times in the longest chain. She will then attempt to fork t before the transaction with Charlie, adding a block which includes a transa which she pays herself:



Unfortunately for Alice, it's now very difficult for her to catch up with the lor Other miners won't want to help her out, since they'll be working on the lor And unless Alice is able to solve the proof-of-work at least as fast as every the network combined – roughly, that means controlling more than fifty per computing power – then she will just keep falling further and further behinc she might get lucky. We can, for example, imagine a scenario in which Alic one percent of the computing power, but happens to get lucky and finds si blocks in a row, before the rest of the network has found any extra blocks. she might be able to get ahead, and get control of the block chain. But this event will occur with probability $1/100^6 = 10^{-12}$. A more general analys these lines shows that Alice's probability of ever catching up is infinitesima she is able to solve proof-of-work puzzles at a rate approaching all other n combined.

Of course, this is not a rigorous security analysis showing that Alice cannc spend. It's merely an informal plausibility argument. The **original paper** in Bitcoin did not, in fact, contain a rigorous security analysis, only informal a

along the lines I've presented here. The security community is still analysi and trying to understand possible vulnerabilities. You can see some of this **listed here**, and I mention a few related problems in the "Problems for the below. At this point I think it's fair to say that the jury is still out on how sec is.

The proof-of-work and mining ideas give rise to many questions. How muc enough to persuade people to mine? How does the change in supply of in affect the Infocoin economy? Will Infocoin mining end up concentrated in t a few, or many? If it's just a few, doesn't that endanger the security of the s Presumably transaction fees will eventually equilibriate – won't this introdu unwanted source of friction, and make small transactions less desirable? great questions, but beyond the scope of this post. I may come back to the (in the context of Bitcoin) in a future post. For now, we'll stick to our focus understanding how the Bitcoin protocol works.

Problems for the author

> I don't understand why double spending can't be prevented in a simpl using **two-phase commit**. Suppose Alice tries to double spend an inf both Bob and Charlie. The idea is that Bob and Charlie would each br their respective messages to the Infocoin network, along with a reque accept this?" They'd then wait some period – perhaps ten minutes – t naysayers who could prove that Alice was trying to double spend. If n are heard (and provided there are no signs of attempts to disrupt the they'd then accept the transaction. This protocol needs to be hardene network attacks, but it seems to me to be the core of a good alternate well does this work? What drawbacks and advantages does it have c the full Bitcoin protocol?
>
> Early in the section I mentioned that there is a natural way of reducing variance in time required to validate a block of transactions. If that var reduced too much, then it creates an interesting attack possibility. Sup tries to fork the chain in such a way that: (a) one fork starts with a bloc Alice pays herself, while the other fork starts with a block in which Alic

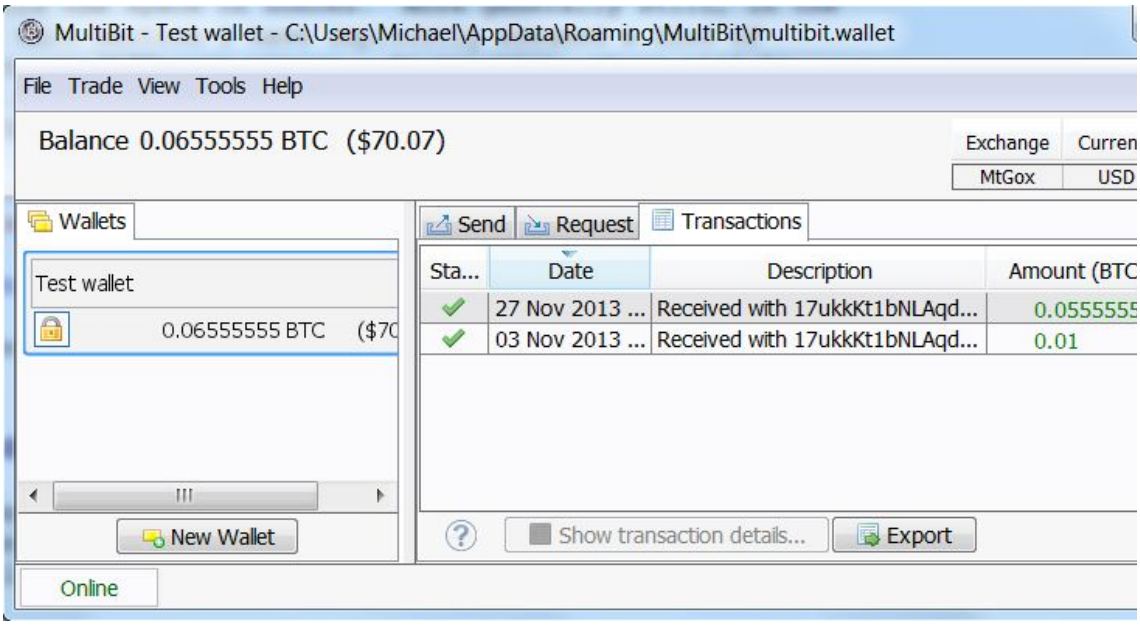
(b) both blocks are announced nearly simultaneously, so roughly half will attempt to mine each fork; (c) Alice uses her mining power to try t forks of roughly equal length, mining whichever fork is shorter – this is hard to pull off, but becomes significantly easier if the standard deviat time-to-validation is much shorter than the network latency; (d) after 5 have been mined on both forks, Alice throws her mining power into m more likely that Charles's transaction is confirmed; and (e) after confi Charles's transaction, she then throws her computational power into t fork, and attempts to regain the lead. This balancing strategy will have small chance of success. But while the probability is small, it will certa much larger than in the standard protocol, with high variance in the tir validate a block. Is there a way of avoiding this problem?

Suppose Bitcoin mining software always explored nonces starting wit then $x = 1, x = 2, \ldots$. If this is done by all (or even just a substant of Bitcoin miners then it creates a vulnerability. Namely, it's possible fo to improve their odds of solving the proof-of-work merely by starting w other (much larger) nonce. More generally, it may be possible for atta exploit any systematic patterns in the way miners explore the space o More generally still, in the analysis of this section I have implicitly ass of symmetry between different miners. In practice, there will be asymm a thorough security analysis will need to account for those asymmetri

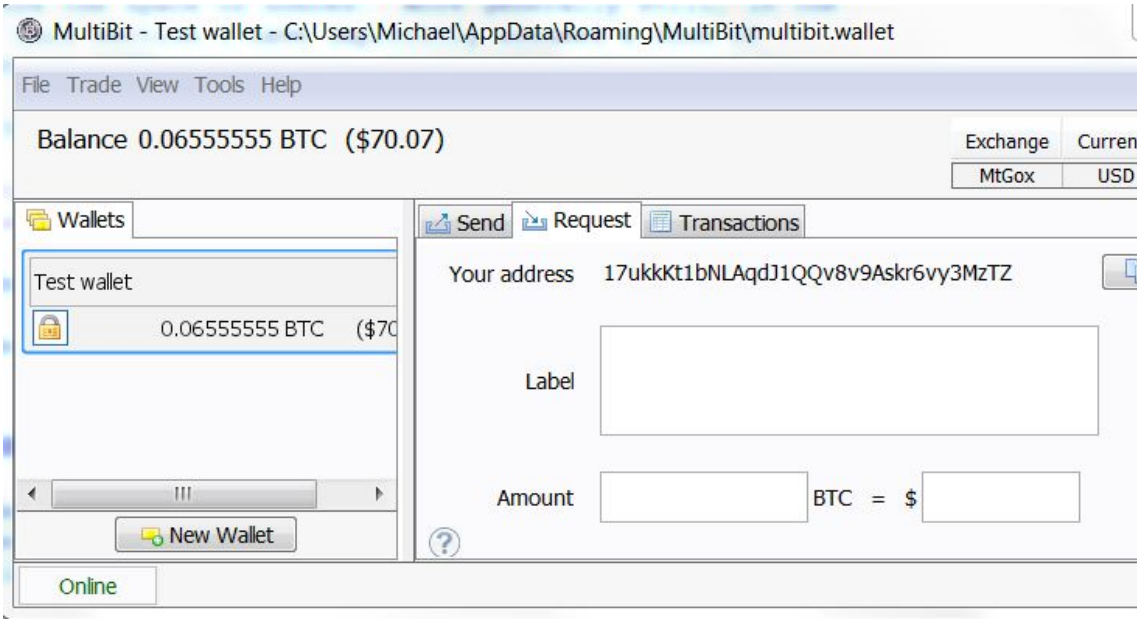
Bitcoin

Let's move away from Infocoin, and describe the actual Bitcoin protocol. T few new ideas here, but with one exception (discussed below) they're mos modifications to Infocoin.

To use Bitcoin in practice, you first install a **wallet** program on your compu you a sense of what that means, here's a screenshot of a wallet called **Mu** can see the Bitcoin balance on the left — 0.06555555 Bitcoins, or about 7 the exchange rate on the day I took this screenshot — and on the right two transactions, which deposited those 0.06555555 Bitcoins:



Suppose you're a merchant who has set up an online store, and you've de
allow people to pay using Bitcoin. What you do is tell your wallet program
a *Bitcoin address*. In response, it will generate a public / private key pair, a
hash the public key to form your Bitcoin address:



You then send your Bitcoin address to the person who wants to buy from y
could do this in email, or even put the address up publicly on a webpage.
since the address is merely a hash of your public key, which can safely be
the world anyway. (I'll return later to the question of why the Bitcoin addres
and not just the public key.)

The person who is going to pay you then generates a *transaction*. Let's tal
the data from an **actual transaction** transferring 0.31900000 bitcoins. W
below is very nearly the raw data. It's changed in three ways: (1) the data
deserialized; (2) line numbers have been added, for ease of reference; an
abbreviated various hashes and public keys, just putting in the first six hex
digits of each, when in reality they are much longer. Here's the data:

```
1.  {"hash":"7c4025...",
2.  "ver":1,
3.  "vin_sz":1,
4.  "vout_sz":1,
5.  "lock_time":0,
6.  "size":224,
7.  "in":[
8.    {"prev_out":
9.      {"hash":"2007ae...",
10.      "n":0},
11.    "scriptSig":"304502... 042b2d..."}],
12. "out":[
13.   {"value":"0.31900000",
14.    "scriptPubKey":"OP_DUP OP_HASH160 a7db6f OP_EQUALVERIFY OP_
```

Let's go through this, line by line.

Line 1 contains the hash of the remainder of the transaction, `7c4025...`, ex
hexadecimal. This is used as an identifier for the transaction.

Line 2 tells us that this is a transaction in version 1 of the Bitcoin protocol.

Lines 3 and 4 tell us that the transaction has one input and one output, res
talk below about transactions with more inputs and outputs, and why that's

Line 5 contains the value for `lock_time`, which can be used to control whe
transaction is finalized. For most Bitcoin transactions being carried out tod

`lock_time` is set to 0, which means the transaction is finalized immediately

Line 6 tells us the size (in bytes) of the transaction. Note that it's not the m amount being transferred! That comes later.

Lines 7 through 11 define the input to the transaction. In particular, lines 8 tell us that the input is to be taken from the output from an earlier transacti given `hash`, which is expressed in hexadecimal as `2007ae...`. The `n=0` tells the first output from that transaction; we'll see soon how multiple outputs (from a transaction work, so don't worry too much about this for now. Line 1 the signature of the person sending the money, `304502...`, followed by a s then the corresponding public key, `04b2d...`. Again, these are both in hexa

One thing to note about the input is that there's nothing explicitly specifyin bitcoins from the previous transaction should be spent in this transaction. the bitcoins from the `n=0`th output of the previous transaction are spent. So example, if the `n=0`th output of the earlier transaction was 2 bitcoins, then 2 will be spent in this transaction. This seems like an inconvenient restriction trying to buy bread with a 20 dollar note, and not being able to break the n The solution, of course, is to have a mechanism for providing change. This done using transactions with multiple inputs and outputs, which we'll discu next section.

Lines 12 through 14 define the output from the transaction. In particular, lin us the value of the output, 0.319 bitcoins. Line 14 is somewhat complicate thing to note is that the string `a7db6f...` is the Bitcoin address of the intend recipient of the funds (written in hexadecimal). In fact, Line 14 is actually a expression in Bitcoin's scripting language. I'm not going to describe that la detail in this post, the important thing to take away now is just that `a7db6f.` Bitcoin address.

You can now see, by the way, how Bitcoin addresses the question I swept rug in the last section: where do Bitcoin serial numbers come from? In fact the serial number is played by transaction hashes. In the transaction abov example, the recipient is receiving 0.319 Bitcoins, which come out of the fi

an earlier transaction with hash `2007ae...` (line 9). If you go and look in the chain for that transaction, you'd see that its output comes from a still earlie transaction. And so on.

There are two clever things about using transaction hashes instead of seri First, in Bitcoin there's not really any separate, persistent "coins" at all, jus series of transactions in the block chain. It's a clever idea to realize that yc need persistent coins, and can just get by with a ledger of transactions. Se operating in this way we remove the need for any central authority issuing numbers. Instead, the serial numbers can be self-generated, merely by ha transaction.

In fact, it's possible to keep following the chain of transactions further back Ultimately, this process must terminate. This can happen in one of two wa possibilitty is that you'll arrive at the very first Bitcoin transaction, containe called **Genesis block**. This is a special transaction, having no inputs, but output. In other words, this transaction establishes an initial money supply Genesis block is treated separately by Bitcoin clients, and I won't get into t here, although it's along similar lines to the transaction above. You can se deserialized raw data **here**, and read about the Genesis block **here**.

The second possibility when you follow a chain of transactions back in tim eventually you'll arrive at a so-called *coinbase transaction*. With the excep Genesis block, every block of transactions in the block chain starts with a coinbase transaction. This is the transaction rewarding the miner who vali block of transactions. It uses a similar but not identical format to the transa above. I won't go through the format in detail, but if you want to see an exa **here**. You can read a little more about coinbase transactions **here**.

Something I haven't been precise about above is what exactly is being sig digital signature in line 11. The obvious thing to do is for the payer to sign t transaction (apart from the transaction hash, which, of course, must be ge later). Currently, this is *not* what is done – some pieces of the transaction a This makes some pieces of the transaction **malleable**, i.e., they can be ch However, this malleability does not include the amounts being paid out, se

recipients, which can't be changed later. I must admit I haven't dug down i
details here. I gather that this malleability is under discussion in the Bitcoir
community, and there are efforts afoot to reduce or eliminate this malleabil

Transactions with multiple inputs and outputs

In the last section I described how a transaction with a single input and a s
works. In practice, it's often extremely convenient to create Bitcoin transac
multiple inputs or multiple outputs. I'll talk below about why this can be use
let's take a look at the data from an **actual transaction**:


```
{"hash":"993830...",
"ver":1,
"vin_sz":3,
 "vout_sz":2,
 "lock_time":0,
 "size":552,
 "in":[
   {"prev_out":{
     "hash":"3beabc...",
        "n":0},
     "scriptSig":"304402... 04c7d2..."},
   {"prev_out":{
        "hash":"fdae9b...",
        "n":0},
     "scriptSig":"304502... 026e15..."},
   {"prev_out":{
        "hash":"20c86b...",
        "n":1},
     "scriptSig":"304402... 038a52..."}],
 "out":[
   {"value":"0.01068000",
     "scriptPubKey":"OP_DUP OP_HASH160 e8c306... OP_EQUALVERI
```

```
23.    {"value":"4.00000000",
24.      "scriptPubKey":"OP_DUP OP_HASH160 d644e3... OP_EQUALVERI
```

Let's go through the data, line by line. It's very similar to the single-input-si transaction, so I'll do this pretty quickly.

Line 1 contains the hash of the remainder of the transaction. This is used a identifier for the transaction.

Line 2 tells us that this is a transaction in version 1 of the Bitcoin protocol.

Lines 3 and 4 tell us that the transaction has three inputs and two outputs, respectively.

Line 5 contains the `lock_time`. As in the single-input-single-output case thi which means the transaction is finalized immediately.

Line 6 tells us the size of the transaction in bytes.

Lines 7 through 19 define a list of the inputs to the transaction. Each corre an output from a previous Bitcoin transaction.

The first input is defined in lines 8 through 11.

In particular, lines 8 through 10 tell us that the input is to be taken from the output from the transaction with `hash 3beabc...`. Line 11 contains the signa followed by a space, and then the public key of the person sending the bit

Lines 12 through 15 define the second input, with a similar format to lines 11. And lines 16 through 19 define the third input.

Lines 20 through 24 define a list containing the two outputs from the transa

The first output is defined in lines 21 and 22. Line 21 tells us the value of t 0.01068000 bitcoins. As before, line 22 is an expression in Bitcoin's scripti language. The main thing to take away here is that the string `e8c30622...` Bitcoin address of the intended recipient of the funds.

The second output is defined lines 23 and 24, with a similar format to the f

One apparent oddity in this description is that although each output has a value associated to it, the inputs do not. Of course, the values of the respe can be found by consulting the corresponding outputs in earlier transactio standard Bitcoin transaction, the sum of all the inputs in the transaction mu least as much as the sum of all the outputs. (The only exception to this pri Genesis block, and in coinbase transactions, both of which add to the ove supply.) If the inputs sum up to more than the outputs, then the excess is u *transaction fee*. This is paid to whichever miner successfully validates the the current transaction is a part of.

That's all there is to multiple-input-multiple-output transactions! They're a p variation on single-input-single-output-transactions.

One nice application of multiple-input-multiple-output transactions is the id *change*. Suppose, for example, that I want to send you 0.15 bitcoins. I can spending money from a previous transaction in which I received 0.2 bitcoi course, I don't want to send you the entire 0.2 bitcoins. The solution is to s 0.15 bitcoins, and to send 0.05 bitcoins to a Bitcoin address which I own. T bitcoins are the change. Of course, it differs a little from the change you m in a store, since change in this case is what you pay yourself. But the broa similar.

Conclusion

That completes a basic description of the main ideas behind Bitcoin. Of co omitted many details – this isn't a formal specification. But I have describe ideas behind the most common use cases for Bitcoin.

While the rules of Bitcoin are simple and easy to understand, that doesn't it's easy to understand all the consequences of the rules. There is vastly m could be said about Bitcoin, and I'll investigate some of these issues in fut

For now, though, I'll wrap up by addressing a few loose ends.

How anonymous is Bitcoin? Many people claim that Bitcoin can be used anonymously. This claim has led to the formation of marketplaces such as

(and various successors), which specialize in illegal goods. However, the c
Bitcoin is anonymous is a myth. The block chain is public, meaning that it's
for anyone to see every Bitcoin transaction ever. Although Bitcoin address
immediately associated to real-world identities, computer scientists have d
deal of work figuring out how to de-anonymize "anonymous" social netwo
block chain is a marvellous target for these techniques. I will be extremely
the great majority of Bitcoin users are not identified with relatively high cor
ease in the near future. The confidence won't be high enough to achieve c
but will be high enough to identify likely targets. Furthermore, identification
retrospective, meaning that someone who bought drugs on Silk Road in 2(
be identifiable on the basis of the block chain in, say, 2020. These de-anor
techniques are well known to computer scientists, and, one presumes, the
NSA. I would not be at all surprised if the NSA and other agencies have al
anonymized many users. It is, in fact, ironic that Bitcoin is often touted as a
It's not. Bitcoin is, instead, perhaps the most open and transparent financia
the world has ever seen.

Can you get rich with Bitcoin? Well, maybe. Tim O'Reilly **once said**: "M
gas in the car – you need to pay attention or you'll end up on the side of th
a well-lived life is not a tour of gas stations!" Much of the interest in Bitcoin
from people whose life mission seems to be to find a *really big* gas station
admit I find this perplexing. What is, I believe, much more interesting and e
to think of Bitcoin and other cryptocurrencies as a way of enabling new for
collective behaviour. That's intellectually fascinating, offers marvellous cre
possibilities, is socially valuable, and may just also put some money in the
money in the bank is your primary concern, then I believe that other strate
much more likely to succeed.

Details I've omitted: Although this post has described the main ideas beh
there are many details I haven't mentioned. One is a nice space-saving tri
the protocol, based on a data structure known as a **Merkle tree**. It's a deta
splendid detail, and worth checking out if fun data structures are your thing
get an overview in the **original Bitcoin paper**. Second, I've said little abou
Bitcoin network – questions like how the network deals with denial of ser

how nodes **join and leave the network**, and so on. This is a fascinating t
also something of a mess of details, and so I've omitted it. You can read m
at some of the links above.

Bitcoin scripting: In this post I've explained Bitcoin as a form of digital, o
But this is only a small part of a much bigger and more interesting story. A
seen, every Bitcoin transaction is associated to a script in the Bitcoin prog
language. The scripts we've seen in this post describe simple transactions
gave Bob 10 bitcoins". But the scripting language can also be used to exp
more complicated transactions. To put it another way, Bitcoin is *programm*
In later posts I will explain the scripting system, and how it is possible to us
scripting as a platform to experiment with all sorts of amazing financial inst

Thanks for reading. Enjoy the essay? You can tip me with Bitcoin (!) at ad
17ukkKt1bNLAqdJ1QQv8v9Askr6vy3MzTZ. You may also enjoy the ***first***
my forthcoming book on neural networks and deep learning, and may wish
me on Twitter*.*

Footnote

[1] In the United States the question "Is money a form of speech?" is an im
legal question, because of the protection afforded speech under the US C
In my (legally uninformed) opinion digital money may make this issue mor
complicated. As we'll see, the Bitcoin protocol is really a way of standing u
rest of the world (or at least the rest of the Bitcoin network) and avowing "I
give such-and-such a number of bitcoins to so-and-so a person" in a way t
extremely difficult to repudiate. At least naively, it looks more like speech t
exchanging copper coins, say.

From

259 Comments

Sumedh permalink

Thanks, I was always too lazy to look up BTC in detail. Your article clearec questions.

I wanted to know one thing what if some smart hacker is able to find some vulnerability in the protocol and he uses that to generate new bitcoins for h Once that happens then whole confidence in bitcoins would be gone and i to chaos.

Is the above scenario possible?

Bobby permalink

Your scenario is possible. Just like any other popular piece of software there are incentives for finding exploits, but there are benevolent hackers examining the code to uncover and fix the

Michael Nielsen permalink

@Bobby: Good point! Yes, that solves much of the problem ne broad point about asymmetries is still true, however. (And is v demonstrated by the rise of large mining pools.)

Edit: This is in response to your comment below. I must have the wrong link when I replied.

achil permalink

each block starts with a coinbase trans the person who solved it. since this trar (each node working on the network has

recipient of that transaction), all block i
should'nt (at least not likely) see two bl

gwern permalink

That bug has actually happened before, but Satoshi/Gavin fix
anyone else managed to exploit it. (There have been 2 major
Bitcoin that I know of: one allowed you to generate billions of
the other allowed you to spend anyone's bitcoins. Neither was
before being patched, and there don't seem to have been any
found since.)

Rena permalink

Interesting. How did those exploits wor

gwern permalink

I don't know the
patches. From t
so bad transacti

Bobby permalink

I believe have the answer to your third question.

The raw block data that each miner is trying to solve contains a generatio
transaction. That transaction is where their coins are sent if they solve tha
Because miners competing against each other want their coins to be sent
addresses, and those addresses are hashed together with their nonce, it c

matter if everyone starts their nonce from zero. The added randomness fr
generation transaction addresses prevents each miner from working in the
space as others.

Michael permalink

Thanks Bobby. I had wondered about the same question as th
Your explanation clears it up for me.

JC permalink

Moreover the nonces need not be enumerable. If randomly pic
large enough pool it is unlikely that the same nonce gets picke

Carles permalink

Very well written!

Only one thing to add (on another post): when you launch Multibit (or bitcc
where does it connect? a list of IPs? DNS? etc. etc.

Pål Driveklepp permalink

Bitcoin has 3 methods for finding peers:
https://en.bitcoin.it/wiki/Network#Bootstrapping

Benoit permalink

Thank you for the great write-up!

You write “I’ll return later to the question of why the Bitcoin address is a ha
just the public key”. Did I miss it? Does it have anything to do with quantur
computing?

Michael Nielsen permalink

Oops – actually, I had an extended discussion of this question
it just before I posted. The reason I deleted it is that the discus
inconclusive. The separation seems to be a fairly arbitrary des
– there are some minor space and security advantages, but n
(in my opinion) to justify making the Bitcoin address the hash
the public key.

Mark Friedenbach permalink

There’s a very serious security advanta
until the moment it is spent. That reduc
private key could be derived and used
minutes. This has significant ramificatic
quantum-proof cryptography, if nothing

And space-wise we’re talking about sav
gigabytes of data from the UTXO set. T

Michael Nielsen

Mark,

You’ve describe
both seem like r
reuse addresses
problem could b

could only ever
soon; (b) it intro
to publish an ad
when the public
confirmed.

On the second p
savings is less t
the type of trans
other similar sav
seem a bit arbitr

(Actually, it occu
Bitcoin transacti
nice example fo

Boldra **permalink**

Could the protoc
YAML? It looks l

Benoit permalink

About your first question:

What would be the incentive for non-miners to answer your question? Why would you trust the answers or lack thereof?

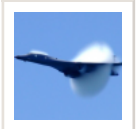
After all, if I understand correctly, when there is no transaction fee set asid miners could very well choose to omit transactions from their blocks?

Michael Nielsen permalink

On incentive: such could be built into the protocol.

On trusting the answers: if someone claims that they see evid double spending, you'd require them to present evidence in th signed transaction. The requirement of a signature makes this forge by a malicious naysayer.

On your last point, yes, this is a very interesting question. At p all seems to be working okay, but over the long run I suspect v use of Bitcoin for small transactions.



Tom permalink

On the last point: I could see the transa the time required to confirm a transfer. quicker, then you have to pay.

On a related note – what happens if blc
order?

MG permalink

I'm curious what happens if/when the underlying crypto is either undermin
broken?

Over the years we've seen flaws that reduce the bits (entropy) in many ma
How would the bitcoin protocol handle, say, a reduction of even 1 bit of dif
reduction == 1/2 as hard to attack) .

Also could someone with very large resources overwhelm the network witl
Eg, if china wanted to use some super computers or a bot net to stop bitcc
operating by adding all sorts of bad data to the block chains?

Michael Nielsen permalink

Denial of service type attacks are a real problem. See, e.g:
https://en.bitcoin.it/wiki/weaknesses#Denial_of_Service_.28D

On the first question, the answer is, I think: "That's really comp
depends on the exact scenario of the break".

cabin permalink

Android had a bug in their random number api that was succe
exploited. Losing a few bits of entropy won't matter, but in this
lost nearly all of them.

http://thegenesisblock.com/security-vulnerability-in-all-android
wallets/

MG permalink

One additional question, what happens to bitcoins that are “lost” . ie What the FBI refuses to sign over the bitcoins seized from Silk road, or wallets tl some coins but were lost due to hard drive failure (bad backups) or lost pa Or maybe someone dies but the next of kin doesnt know the details?

The comparison is If I drop $20 on the ground or my next of kin finds it unc mattress, they can use it.

Michael Nielsen permalink

Lost bitcoins are just that – gone from the money supply for g someone manages to either (a) recover the keypair; or (b) bre underlying crypto.

Mango Cat permalink

That brings up an interesting scenario, have to be some allowance made for re sub-division of the satoshi.

James H permalink

With Bitcoin; losing the private key for good is more like accid dropping your coins out of an airplane over the pacific ocean.

The private key is crucial to recovering those coins.

Hasitha N. Liyanage permalink

Looks like we both independently arrived at similar methods of explanatior
http://zen.lk/2013/11/28/how-i-finally-understood-bitcoin/

Amos permalink

In the second paragraph of the Bitcoin section, seems it should be 0.0655
0.6555555

Michael Nielsen permalink

Thanks, fixed.

Vidya permalink

Thank you so much !!!! I had wanted an understandable primer on Bitcoin
and this was a fabulous read !

Boldra permalink

I'm surprised that bitcoins in a transaction are a decimal string. It looks like
floating point approximation errors.

Have you read about coin-join? It follows on very nicely from what you've
here.

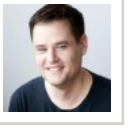
I'm looking forward to the next one

Michael Nielsen permalink

They're not actually a float — as I mention in the article, the m
of Bitcoin is the Satoshi, which is one one hundred millionth of

So it's really specifying an integer number of Satoshis.

(I haven't checked what type is used in the source code; I'd be
to know.)



Michael Nielsen permalink

This page says that it is an integer in th

https://en.bitcoin.it/wiki/Proper_Money_

However, it sounds as though there car
point / rounding issues with code used
the Bitcoin network.

Uri permalink

Just wanted to say thanks for a really great essay — the explanation was
and totally fascinating.

Jim permalink

Can quantum computers mine bitcoin faster? Does this boil down to how c
quantum computer can find a string that has a specified property for SHA-
which we have a quadratic speedup, but probably no more?

I understand that commonly used digital signatures and public-key cryptos
broken by quantum computers, so there's not much to be said about that.

Michael Nielsen permalink

I haven't thought much about it. With that said, I'm pretty sure comments are right – quadratic speedup for finding hash collis the asymmetric crypto stuff is broken.

A.Sallai permalink

This is an incredibly well written article and one that i needed so much. Th Michael!

Oli Rhys permalink

Thanks for this, while I understood the majority of it, the coding element w useful – especially highlighting where the script goes in conjunction with th transaction.

While a lot of people know abot bitcoin, there is such a shortage of good q technical info.

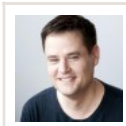
Thanks again 😉

Andrew Jaeger permalink

Great writeup about how bitcoin functions on a technical level, but I had a about it as it's use as a currency.

Why is bitcoin built to be inherently deflationary? This seems to be the go- against why it will ever gain widespread adoption as a currency. Why does for mining bitcoin halve every 210,000 blocks? Could there be a point in th where this is reversed?

Michael Nielsen permalink

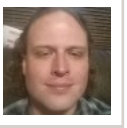


Good questions, and I don't know. I certainly suspect (as do y
these may ultimately turn out to be design flaws.

Assaf permalink

Bitcoin is NOT deflationary. It is inflationary with a known and
rate up until around 2140 at which point it will stop being inflat
only deflation in Bitcoin may happen through coin loss. The sa
way, is true for Fiat. The difference is that Fiat can be arbitrari
and with Bitcoin it is not arbitrary.
As far as why inflation is predetermined, this is so Bitcoin is a
of value which is one of the defining properties of a good "mor
Why is it inflationary at all (as in, why not start with a predeter
amount of bitcoins that never change). The is part of Bitcoin's
decentralized design. Bitcoin designers wanted a way to sprea
around without starting with a central authority that has them a
them out (like, say, ripple). The bitcoin generating part of minir
exactly that.

BTW/ I am yet to see a good argument about why having a m
system that is a good store of value and does not get diluted c
with inflation is bad.



KRG permalink

Inflationary/Deflationary are properties
the supply of real goods. Bitcoin is only
real wealth production will gradually slo
2140 at the same pace as the drop in E
even accounting for the effective tax pr
needing to increase to attract the neces
system of directly paying for that servic

to be paid out in each transaction to co
actual payments will have to fall to mak
revenue translates to lower ability to afi

Bitcoin has already seen hyperdeflatior
but the defined limitations in supply are
speculation over it as a commodity thar
predictable future value (never mind us
measure to reliably store value) rather

What actually needs to be demonstrate
allowing any static, nonproductive accc
as opposed to using the inherent declir
provide the baseline motivation to use i
store anything beyond cash sufficient tc
liquidity. There's no justification to use t
value, because value is a property or re
serves to account for them. Trying to st
future production potential is the ultima
fraud and financial manipulation far out
real assets.

mellyra **permalink**

Assaf is talking a
talking about pri

Both usages are

Jonathan Goldl

There are excell
is the desire to s

Back in 1958 Pa
money as a stor

An Exact Consu
contrivance of m
JPE V66 6 (Dec

Dave permalink

Actually bitcoin is inherently deflationar
bitcoin economy will grow faster than th
quite intuitive, it does make sense upor
reflects the value of the economy it rep
growing faster than the underlying ecor
money supply is growing slower than th

I think all but a few of us expect the bitc
the supply of bitcoins — hence we have

The wisdom of that choice is another m
many different scenarios for the amoun
new currency entering the system. If bi
then my bet is that these will be signific
among variants.

Wanton permalink

I'm confused about the block chain. Does everyone have their own versior
they sync to a master? Does every block chain get updated when validatic
completed? Does this mean every person has a record of everyone else's
transactions for ever? Won't this file get really really big?

Michael Nielsen permalink

In practice, there are thin clients which don't keep a full copy c chain. But the way the protocol is designed at present there is number of people keeping a full copy of the block chain. This i quite a manageable size (about 12 gig). If Bitcoin grows rapidl this may eventually become a problem. There's a nice discuss and related scalability issues here:

https://en.bitcoin.it/wiki/Scalability

The conclusion there, which seems to me believable, is that th many options for scaling Bitcoin at least up to the level at whic cards are used today, and perhaps further.

Sunny permalink

Just about the total amount of bitcoins, if I understand well, new bitcoins a generated each time a transaction is processed? It means the more excha have, the more bitcoins in the market there is ?
So the only way to raise the number of bitcoins is to spend some energy v transaction (that's a little bit wired for me ;-). How were created the first bit there another way of creating bitcoins that checking transactions ?
And thanks a lot for this post because it's really difficult to get a clear pictu it. Regards

Marco permalink

1. Not per transaction but per block (of transactions).
2. Exchanges are a bad example. The transactions within the happen outside the network. Only if you deposit or withdrawal to/from an exchange, it goes over the network and therefore s

the block chain. There are so many trades going on within an
happens internally. And since trades need to happen fast, the
not suited for that.

3. yes

4. Google for the ‘Genesis Block’. That’s how it got started.

5. no

Daniel R. Grayson permalink

Very nice!

Comments: you use the concept of mining before defining it. Change “pos
vulnerabilities” to “possible vulnerabilities”. Fix “spending spending money

---

Michael Nielsen permalink

Thanks, typos fixed!

Greg permalink

Excellent article!

I found two typo’s which you might want to zap:

“Bob doesn’t just go ahead and accept the transaction. Instead, he broadc
message to the entire network.” (Change ‘broadcast’ to ‘broadcasts’).

“Will Infocoin mining end up in concentrated in the hands of a few, or many
(Remove first ‘in’).

Michael Nielsen permalink

Thanks, typos fixed!

Mango Cat permalink

Thanks for the great Bitcoin writeup. I have gleaned most of what you said pieces from articles and message boards, it's nice to see it all described s one place.

What I think is more interesting than the cryptography aspect is the social- aspect of Bitcoin and why it seems to be succeeding. First big mover and seem to be in-play, and the "anonymous cash" myth also was a big factor, that, I think the carrot of "get paid for solving hard problems (often using of computing resources)" has drawn in many participants who have helped g network by promoting their own self interests.

I balk every time I hear the bit about "every transaction for all users for all encoded into the block chain" especially when combined with "the chain is by solving hard problems". Scaling this system to support a billion users tr multiple times per day seems…. unlikely. Your explanation does help to sh problems don't get much harder as transactions scale up – the blocks ther larger, but the hash problem doesn't get significantly harder as block size unless you start talking about transacting the world's monetary business ir system, then those blocks would get uncomfortably large in a very short ti and the forking problem would be much more complex than choosing betv three chains to follow.

I also somewhat disapprove of the concept of encouraging people to "mine space to earn currency" since that creates an artificial demand for energy grow into a significant waste of "real world" resources as such a system sc

I have been playing "trust network" thought games since the 1980s, I'd like peer-to-peer digital currency system that is based in the concept of trustw identities instead of solving hash problems to get paid. You could still get p validating transactions, but it wouldn't have that appeal of "solve more pro

more paid…” plus, a system based on trust would tend to concentrate trus authorities that would be quickly perceived as hopeless to surpass in trust semi-defeating the incentive to participate as a trusted member of the netv some kind of carrot to the underdogs was included – which would be pure motivational instead of a technically required component.

Anyway, all very interesting to watch. As usual, I got in late and out early w (bought around 5, sold around 120, seemed like an awesome profit margi time…) that aspect of Bitcoin is a lot like any other speculative investment certainly fueling interest at this stage.

Michael Nielsen permalink

On scalability, check out https://en.bitcoin.it/wiki/Scalability. Th of useful information there. Like you, though, I wonder about t economics (and impact) of mining.

Michael Newman permalink

Thanks for writing this great explanation of Bitcoin. I noticed in the first Bitc transaction example, you mention 0.39 bitcoins, but the example really de bitcoins, where 0.319 bitcoins goes to one person, and there is a 0.001 bit transaction fee. In other words, did you mean “0.319” instead of “0.39”? Al a need to show “0.31900000” value as an image?

Michael Nielsen permalink

Thanks, fixed.

Joseph Reagle permalink

typo: “trying to understand possibility vulnerabilities”: possible

Michael Nielsen permalink

Thanks, fixed.

Jack L permalink

Thanks for the excellent writeup. I have a question about one item, hopefu
explain it.

It appears the money you send someone is merely chunks of one or more
transactions. Let’s say I receive 1 bitcoin at myaddress_123 and I receive
myaddress_456. I now want to send you 2 bitcoins from myaddress_789.
previous transactions are the inputs for my transaction to you.

How does the transaction message for the 2 bitcoin transaction prove that
recipient of those previous transactions when the addresses are all differe
hash for each input in the new transaction something that can only be gen
whoever was the recipient in the original transactions (myaddress_123 or
myaddress_456)?

If the answer is yes, then it seems like unique addresses can be easily link
case I don’t see any anonymity advantage of using new addresses for eac
transaction. Sorry if I’m missing something obvious here.

Michael Nielsen permalink

The proof is in the digital signature. That signature is generate
public key which must match (when hashed) the address from
to the earlier transaction. That proves that the bitcoins are the
spend.

ajay permalink

But (if I understand correctly) the need
verified means that you are tied to all y
maintain a double life.

If I were a criminal, I might find it very d
Stringer, who sells drugs, and Russell E
and pillar of the community – and use t
bankroll Russell's legitimate businesse
with Bitcoin; I can't transfer Stringer's b
else in the world knowing about it. Anyo
can notice that the flow of money goes
Stringer, to Russell.

Jim Lyon permali

If you really wan
would let any cu
withdraw value,
customers' acco
that were the ba
originally deposi

Such a bank wo
pay interest, ma
traditional currer

Space2001 perm

Extending from
more of your bit
transaction. You
transactions eac
one simple scen
into which all the
xxx address as t
necessary amou
set up specifical
middle-man pro
all the incoming
separate out sub
yyyy addresses
association. Mar
on what I have r
provider are unc

Con Kolivas permalink

Nonce starting at zero is not a vulnerability. Shares are stochastically distr
throughout the 2^32 nonce range and it makes no difference where you st
nonce is simply 32 bits out of the whole 320 bit coinbase that you are hash
there is no way to design a target solution to be distributed anywhere withi
range of those 32 bits. If you start at a higher nonce value you simply will h
possible chances at finding a solution before you'll need to get/create a ne
to hash.

Alex F permalink

I think maybe you're misinterpreting his concern: the danger is
someone can solve blocks *strictly* faster by starting at a diffe
because as you say, the correct nonce could be anywhere in t
0..2^32 so every guess has the same 1/2^32 chance of being

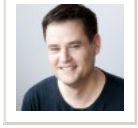
Instead, the danger is that someone could solve blocks faster
everyone else* if they start at a higher nonce and everyone el
0. Specifically, assuming (on average) everyone can calculate
the same rate, then any transaction whose correct nonce is hi
will always be solved first by someone who started at Y, if 0<Y

Of course this creates an obvious incentive for all participants
guess nonces in a different order than everyone else. So it se
reasonable that most client software would use a random seq
nonce guesses rather than guessing sequentially from 0. But s
were to find a vulnerability in the random number generator of
client, then it might be possible to design a competing client w
in practice, almost always find the correct nonce before the ta
client, by virtue of guessing the same sequence a few steps a
would allow the attacker to successfully validate a share of blo

than their actual portion of the collective computational power, of everyone using the vulnerable client and finding the nonce than they should on average.

Gergely Imreh permalink

I think there's also a "time" field in the p
updated every few seconds. Thus in pr
everyone has the same message and t
everyone has a different message, reg
https://en.bitcoin.it/wiki/Block_hashing_

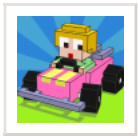


Michael Nielsen permalink

Alex has explained my concern well. H
have pointed out (including Gergely) in
differences in the blocks being hashed
sufficient to make this a non-issue.

Rubberman permalink

A most excellent and well written article! I look forward to more! Thanks.



Bram Stolk permalink

As people make transactions, the public ledger grows.
Will it not grow to an unmanageable size at some time?

Julian permalink

I have a couple questions, possibly a subject for a future article.

1. If the block chain forks, do the miners on both sides of the fork keep the
If so, doesn't it allow someone to continue executing the proof of work eve
known that someone else has solved the proof of work?

2. I am puzzled by transactions in blocks. Is it not possible for two miners t
working on different blocks which contain mostly, although not all, the sam
transactions? Then, the first one to solve the proof of work will have valida
the transactions in the second miner's block. Does the second miner resta
his unverified transactions and putting them in a new block?

Michael Nielsen permalink

On 2, yes: if you're mining and someone else validates some
transactions you are working on, then you remove them from
but continue working with the unvalidated transactions.

On 1, it's true that in different forks, different miners will have
rewarded. However, over time only one of the forks will becom
accepted consensus for confirmed transactions. And so only t
from one fork will be able to redeem their transactions.

Rob permalink

Fascinating read, thank you!

One thing I'm having difficulty with is block chain integrity. What will happe
owner loses his wallet restores a backup from a few weeks back. He may
some coins, and he may have received some. Those transactions are no l
block chain. How would the block chain get back in sync?

On your question-to-yourself about using two phase commit, I think the ma
would be vulnerability to denial-of-service attack. A malicious user could s

swarm of identities to act as nay-sayers and therewith deny some or all otl performing transactions.

Benjamin Marty permalink

In my experience using the bitcoin client, you are not allowed anything on the bitcoin network until your block chain is in syn latest transactions. It somehow recognizes how far behind you chain is and starts downloading blocks and tells you how old y chain is and how much left you have to update as it download BTW, I un-installed the bitcoin client because over the 1 year s had it installed, the block chain went from about 2 GB to abou and the novelty of having my own copy of the block chain wor comparison to its cost. It would be nice if there were some kin block" that could be generated that flattened the tree into a sir enumerating the value stored at each address.

Michael Nielsen permalink

On the naysayer DDoS attack on two-phase commit: if someo that they see evidence of double spending, you'd require them evidence in the form of a signed transaction. The requirement signature makes this hard to forge by a malicious naysayer.

uez permalink

Here is a very entertaining rational explanation
http://www.bringhurst.org/2013/04/03/how-does-bitcoin-crypto-work.html

Benjamin Marty permalink

One thing I still don't fully understand is how the bitcoin reward size is dec
awarded. Who enforces the rules that 25 bitcoins are awarded for validatir
and a few years hence, it'll be 12.5 bitcoins? If we were to decide that the
should be different (remaining at 25 indefinitely, for example), what exactly
to change? Is it the bitcoin mining clients that are hardwired to only validat
transactions that award 25 coins to other miners when they validate their b
the date of the validated block indicates that the award should be 25 BTC?

Michael Nielsen permalink

It's hardcoded, based on the number of blocks in the blockcha
210,000 blocks the rate halves. No need to keep track of the c
count blocks.

Michal permalink

Hi,

And cannot miners just continue to vali
adding 25 bitcoins?

As the chain is just validated list of tran
cap on transactions?
What does hardcoded mean practically

Jozef permalink

You only own th
hardcoded here
supposed) to be
continue giving I
reward to 12.5, I

others who resp

Nick P permalink

I hadn't had time to thoroughly delve into the protocol and your excellent w
exact piece by piece, what/why I needed. The "why's" are extremely impor
people who might want to build on top of the protocol as it helps them und
what they should or shouldn't modify.

pebird permalink

I was thinking about how the blockchain is managed as more transactions
processed, thanks for the link https://en.bitcoin.it/wiki/Scalability

Interesting, one of the potential solutions discussed is the use of dedicatec
instead of lightweight clients to increase transaction rates, reduce latency,
increasing blockchain size (via techniques such as "pruning" the chain), et

What this implies of course is an evolution into "banks", a group of entities
sufficient resources and staying power to dedicate specialized BTC infrast
transaction handling.

In a way, Bitcoin is replicating a history of money evolution in an accelerat
wonder what will take place in the protocol to allow the peer-to-peer nature
while scaling the project to allow the transaction capacity necessary for a t
currency.

Michael Nielsen permalink

Yeah, that is very interesting. I don't run a full client myself, I u
client that doesn't have a full copy of the block chain. And you
see a lot of signs of centralization with the big mining pools:

https://blockchain.info/pools

Jeremy permalink

Great article. Thanks very much for writing it.

Reader permalink

Typo: requring = requiring

[MN: Thanks, fixed.]

Reader permalink

The concept of 'block' (and/or a definition) is not introduced before it is use
makes the concept difficult to grasp.

Reader permalink

Wouldn't three phase commit be more appropriate than two phase commit

---

Michael Nielsen permalink

That would likely be even better, although I haven't thought ab
lot of detail.

Reader permalink

Grammar:

to take account of those asymmetries => to take into account these asymr

[MN: Not sure that was ungrammatical, but in any case I've improved it: "t for those asymmetries."]

Reader permalink

Multiple typos with '…the recipient is receiving 0.39 Bitcoins…' -> 0.319

[MN: As noted earlier, fixed.]

Stuart Quimby permalink

Thanks for such a generous and informative post. There is so much babbl that it often seems to operate socially as more of a rorschach test on curre actual means of exchange. I quite agree that the details are considerably r interesting than yet another pundit's babble about what it all means. The d delight, are in the details.

Stuart Quimby

Terikan permalink

Bitcoin has fascinated me recently. I admit to not being able to fully wrap n around it, but I took what I could and wrote a little here: http://mimictrading.com/viewtopic.php?f=4&t=293

Maybe you can help me out with one part of this I don't quite get. The sign does the block chain know that the address sending the coins is correct? T

sends their sig to go with it, I assume paired up with the hash of the addre the various nodes to validate right?

But if you are sending your sig out then can't any node have access to tha info. They would need to in order to validate. So can a sig only be used on how is it generated and what prevents it from being faked?

cryptograffre permalink

https://en.wikipedia.org/wiki/Digital_signature

Public key cryptography is a remarkable and beautiful thing. E using Bitcoin has keypairs – one key in each pair is public, the private. The nature of asymmetric cryptographic digital signatu can sign any piece of data using my private key, and anyone e only my public key can verify that the person who signed that the private key. There's some fascinating mathematics involve a simple numerical relationship between the public and private

richnormand permalink

Very nicely done write-up.

Makes me wonder about the news at various times about a major "theft" o mostly in exchanges. In order to benefit they would have to be converted c introduced later on.

Some of these were for large amounts and not really easy to hide, unless on them?

Michael Nielsen permalink

I've wondered the same thing. Some observations: if you copy
private key, and then erase their copy, there is no way for ther
that it was ever truly their key. And if two people both have a c
private key, how do you determine who "truly" owns it?

The situation is complicated further by the possibility of launde
quickly spend some stolen bitcoins on, then it becomes very d
later recover those bitcoins, since now they may be in posses
honest parties.

shyam jos permalink

the best explanation ever , thank you Michael 🙂

Austin permalink

This was a fantastic article and answered all my questions about bitcoins.

Thanks!

someone permalink

What about the actual code? How many miners are using same piece of s

---

Reader permalink

Indeed, this is a critical question.
The more implementations there is, the stronger Bitcoin woulc
would not be dependent on the "features"or flaws of one partic
implementation.
The apparent lack of unambiguous protocol documentation m
think that alternative implementations are difficult to achieve.

Certainly, it would greatly help if there was some form of “RFC Standard”, or “W3C spec” for Bitcoin.

Besmir permalink

HI,
Your article was very interesting and detailed, so I learned a lot more from one question or doubt: What is done with all these hashes? are they gonn for cracking/decrypting encoded data?
what is the real benefit behind generating hash tables?

thank you

muthuveerappan permalink

Did you do this video or is this video inspired by this post !! ?
http://youtu.be/Lx9zgZCMqXE – this too is good…

Michael Nielsen permalink

I didn’t make that. I just watched a few minutes – it looks prett certainly much more detailed and accurate than most of what’

Michael Nielsen permalink

Many people have asked about scalability, so let me just leave this here:
https://en.bitcoin.it/wiki/Scalability

It doesn’t address every possible concern, but I think the upshot is that the room for Bitcoin to grow.

Haroun Kola permalink

Thanks. There's so much to learn about this currency and I'm loving all the that its getting.

Diogo permalink

Great article! I have a question: Could miners run a modified version of the choose not to publish a transaction in the blockchain? I mean, like a small powerful miners controlling the entire network?

Mark permalink

If you control half or more of the total mining power in the netv can keep a transaction out of the blockchain by solving blocks average) than the miners who are trying to include that transa control less than half, you can delay the transaction, but soon the rest of the miners will get ahead of you and your version o blockchain will lose out.

jshell permalink

Great explanation — but doesn't solve this problem:

Bitcoins aren't actually backed by anything other than server time.

There was a time in this country when you can go to the bank and trade in dollar bill for an oz of gold. You can't do that anymore, b/c today dollars ar debt not gold.

But bitcoins are backed by server time. That almost makes less sense tha debt.

Thomas Mahoney permalink

I think you're confusing an investment with a medium of excha
investment should be "backed up" by something, in the sense
should give the holder a claim future cash flows or other real a
medium of exchange is just that, something used to facilitate t
accounting device. It should have scarcity value and be resist
counterfeiting. Fiat currencies have scarcity value to the exten
are usually printed in finite amounts. Gold is generally scarce.
is scarce as well.

Gold has been used as a medium of exchange for centuries. \
"backed up" by? Nothing. It's just scarce, and therefore suitab
medium of exchange.

Miksa permalink

BBC had an interesting article called "V
ago.
http://www.bbc.co.uk/news/magazine-2

Marco permalink

If people are willing to pay for something that is rare or unique
value.
A Ford Mustang '65, first issue Marvel Comic books, baseball
Bitcoins all have value because they're scarce and people are
pay for them. The demand for it defines the price.

Marc permalink

Excellent write-up, and I look forward to further installments – which leads are you no longer updating your RSS feed(s)? I came here from Bruce Sc blog, and I like what I see so I subscribed (in goread.io), but neither of you anything newer than the beginning of 2013.

I'd like to clarify: I'm grateful for your posts, and I'm not complaining if you' the whole RSS thing (Google did, why shouldn't you?). But if you _haven't RSS, but it was supposed to be getting updated automatically… it isn't.

Michael Nielsen permalink

Thanks for pointing this out.

I just checked both RSS feeds, and they seem to be fine. I typ longer essays, often in the 3,000-20,000 word range, which is update my blogs a few times a year. You may enjoy looking th of my past articles. This blog carries my more technical stuff, v other blog (http://michaelnielsen.org/blog) is more general.

Your comment did make me notice and fix some mistaken link sidebar, so thanks for that!

Marc permalink

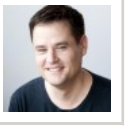
Huh. I clicked through to the Feedburne there. Perhaps the problem is on the g so far, but… Maybe later I'll try again w

Here's what I see for the DDI feed: http and here's what I see for your main fee

I've tried refreshing multiple times, but

I've bookmarked your pages and I can periodically to see if you've got anythin

to know that (at least for some of us) R



Michael Nielse

I checked in my
come through fi

Joseph permalink

Am I to understand that it takes about 60 minutes to pay somebody throug
Bitcoin network? I reached this conclusion based on the 10 minute averag
confirmation and the requirement of it being 6 back in the chain before it is
confirmed.

Thanks,
-Joseph

Michael Nielsen permalink

Full confirmation requires about 60 minutes. Many people are
accept payment on more trust, though, say after just a single
(~10 mins).

Yes, this is a significant disadvantage of the protocol in its pre
architecture.

Marco permalink

Transactions are instant. Confirmations are not.
A confirmation takes 10 minutes. If you want full confirmation, on average it takes an hour (6 confirmations).
For eCommerce, this will probably work in most cases.
For retail, this can be an issue.
However, there are a few points:
– If you try to double spend, it doesn't mean you will succeed.
– It is not easy to double spend in front of the cash register (u have build some app and are all prepared)
– The merchant (I think BitPay is doing this) can listen on the see if there was a double spend attempt. Those are easily det
– Don't forget that a Credit Card payment can be charged bac later. Just saying. 🙂

Gregory Johnson permalink

Love the article. It is the first article that I have been able to understand on and I have been reading a few on it. And a comment to style, I really appre higher-principled discussion on the topic. I am so annoyed with the interne barrage of get-rich-quick articles on this, or the excitement of the exchang have yet to read before now any intelligent comments to the social value, i your link to http://szabo.best.vwh.net/formalize.html was appreciated.

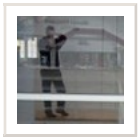
Michael Nielsen permalink

Thanks. All of Szabo's writing is worth reading, incidentally, his treasure trove.

Steffen permalink

You know that there are some hints tha
Nakamoto?

https://likeinamirror.wordpress.com/201
probably-nick-szabo/



Bart permalink

great article.

Ian permalink

Re: why BT doesn't use 2PC, as I understand it, it's because 2PC become
exponentially more complex/unreliable with an increased number of partie

2PC is a collapsed version of the byzantine / paxos protocols (which is 2P
place of 2), and the basic problem is that a lot of nodes have to be online a
lot of messages in order for it to be workable. In any case it's more comple
'longest blockchain wins'.

That's my amateur understanding anyway.

HPublius permalink

Great article and great discussion! This is a very good overview of the tech
aspects around the bitcoin protocol. The fact remains that bitcoins have no
value and the promise of a peer-to-peer payment network (medium of exc
not be fulfilled unless the bitcoin is transformed into a true digital currency.
my thoughts on how to accomplish that: http://tinyurl.com/m57hd2z

OnlyMe permalink

Hi,
first of all great explanation on Bitcoin, I love it!
I guess my question is simple to answer.
How can I verify that a transaction is signed by a certain address if all I got
of the public key? Don´t I need the full public key for that instead of only th
What am I missing guys??

Michael Nielsen permalink

The transaction contains the Bitcoin address of the payee (or
there are multiple outputs) in the output fields, and the public k
signature(s) of the payer(s) in the input fields. So there's no pr
you do have the full public key of the payer.

kgb permalink

Silly question from a non technical person: how will transactions be approv
verified subsequent to 2140 when there are no more rewards for mining?

Michael Nielsen permalink

Transaction fees (which I briefly describe in a couple of places
article).

Tom Hatcher permalink

Thank you. Best explanation I've seen so far! I still don't understand it com it's slowly becoming clearer. One question, though. I hear that it's open so we can look at the source code. I'd like to do that. It's written in C++? Whe the code and look at it? Thank you.

Tom Hatcher permalink

I answer my own question. It's at github.com

Fee Fi Fo Fum permalink

I would love to see you discuss tumblers and the effectiveness and possib anonymizing your bitcoins

Cb permalink

In your anonymous section you speak of debunking a fairly huge myth witl backing it up. You just state the equivalent of "actually it's not anonymous" going into detail.

I don't believe that to be true unless you are implying that various ways of internet anonymously are breakable.

For instance if TOR is compromised versus if it is not, or if other methods traffic surrounding use of bitcoins are insufficient

Your assertion that bitcoin is open and transparent has nothing to do with i use it anonymously, and the claim that it wouldn't be able to 'achieve conv will narrow the pool of suspects down sound closer to a statement of succ anonymity rather than unsuccessful.

Michael Nielsen permalink

"You just state the equivalent of "actually it's not anonymous" v
going into detail."

It's certainly not meant to be a proof! I do, however, go a great
than just saying "it's not anonymous" — I reference a large an
body of academic literature that takes supposedly anonymous
and then de-anonymizes them. I believe techniques similar to
in those papers will be very useful for attacking Bitcoin. There
complications in Bitcoin, notably that some people (though far
routinely use new addresses for each transaction. That makes
interesting challenge, and (I think) is different than in earlier w
anonymization. I'll be most curious to hear what the de-anony
to say after making a sustained attempt at Bitcoin.

Magnus Sorenson permalink

Linking bitcoin addresses to a real iden
somehow associated with an address i
the street with cash – without revealing
public network, and a device with a ma
me, then I can spend that bitcoin with a
personal info to whoever I send that bit
the guy who sends me the drugs would
can be fudged as well. If he does not st
gets linked to me if the drugs arrive saf
on that transaction, not even the NSA c
Now if i do the same thing many times,
other vectors – but explain how anyone
if I buy the bitcoin with cash on the stre

permanent record. I assert that bitcoin
forever- if you do it right.

MerkleTree perm

The system is a
somewhere and
track addresses
during man tran
There are sever

I suggest you lo

You will find that
effort.

To remain anony
This includes th
they will work.

This leads to the
dangerous to sp

Adam Back permalink

You mention using multiple sub-puzzles to reduce variance. This is a bad i
introduces progress. Because bitcoin is a first-past-the-post race, progress
unfair advantage to more powerful miners (>2x reward for 2x power).

Michael Nielsen permalink

You're presuming a particular design (in which people solve a
puzzles privately). There is no necessity to make that presum

why I make the first part of the statement “with some careful d
possible to considerably reduce the variance in the time to val
block of transactions”. I haven’t worked out the best possible c
however, I have found a design whose distribution is quite a b
sharply peaked around 10 minutes than the current puzzle. Ur
the details are more complex than I want to write out right now
come back to it in a future post.

Adam Back permalink

Also you talk about risk of nonce reuse. This wont happen because people
their own reward address, so even if the nonce is reused the work proof w

Further in the case of pool mining the pools hand out work, specifically to a
re-use (which is somewhat insecure as others could guess the work range
users and race them to produce it).

And finally the secure way is pooled miners use getblocktemplate and use
random counter start extranonce. If extranonce is large enough and rando
probability of nonce collision is pratically 0. You can read about this in the
paper http://hashcash.org/papers/hashcash.pdf or https://en.bitcoin.it/wiki/

For decentralization miners should also choose their own blocks by runnin
node and filling in the details into the coinbase provided by getblocktempla

---

Adam Back permalink

two-phase commit: if you are willing to wait 10 minutes, bitcoin already do

I presume the form it would take is the proof of double spend would be (or
double spends.

There have been proposals to forward double-spends with a double-spent
(currently the first only is received).

Maybe just an api to ask if there are any transactions conflicting with a giv transaction a user could ask a few random nodes to gain confidence.

You also have to bear in mind preserving the 0-confirmation spend functio people rely on that for low value point of sale transactions.

SRM permalink

“I’ll return later to the question of why the Bitcoin address is a hash, and n public key.”

I don’t think you ever come back to this topic.

Michael Nielsen permalink

See the discussion above, in reply to Benoit Mason.

JohnT permalink

Thank you for the primer.

You might consider removing the footnote. IMO, Bitcoin cannot be succes defended as free speech. Free speech is not a full blown unlimited right, a in a crowded theater reminds us.

But Article I, Section 8, subparagraph ? does grant Congress full power to money and declare [its] value.” And a subsequent subparagraph grants C power to outlaw any currency it wishes for citizen uses as legal tender.

Thus, IMO, the Supreme Court could never allow free speech to prevail ov Congress’s unfettered Constitutional authority.

I hope this comment does not derail a great discussion of Bitcoin. Please comment if it becomes a red herring.

johnp permalink

a question
re the part about
“”
everyone (collectively) is the bank. In particular, we’ll assume that everyon
Infocoin keeps a complete record of which infocoins belong to which persc
“”
assuming millions and perhaps eventually billions are using bitcoins
so for example billions of transactions could take place daily
times this by the qty of bitcoins (each single one being unique)
times this by the billions using bitcoins
then what affect would that have on the network ?

Michael Nielsen permalink

Already addressed in comments above.

Pavel Masyuk permalink

That question about a nounce… I think that the parametres of the puzzle c
every single miner. Everyone’s desired block contains a unique transactior
other miner has – a transation of giving a reward to himself. So there is no
trying to trick others – parametres of their puzzles are different. Its OK for
just try 0,1,2 etc…

Michael Nielsen permalink

Already addressed in comments above.

Joshua Holden permalink

So if I've got this right, one proof-of-work computation takes about 10 minu you you (currently) get 25 Bitcoins for doing it, and each Bitcoin is (current about 1000 USD. Right? If so, this only makes sense if most proof-of-work computations don't get finished and/or don't get rewarded. Is that usually b someone else got there first? Do you know about what fraction of proof-of- computations get rewarded?

This was really useful; thanks!

Steffen permalink

Joshua,

yes. It's a race. Whoever finds the hash that is smaller than th defined difficulty, they will gain the reward for the block. The d adapted every two weeks or so to reflect the changing (now g power of the network.
The power is growing so fast and so much that some already "arms race".

-st

Chris Crozier permalink

Thanks for the article, very interesting, but I don't see any mechanism for r combining fractional bitcoins.

I'm puzzled by what seems to be an ever-increasing fragmentation. It seer time you would accumulate a large number of coins of varying fractional va to make a payment you would have to lump together a collection of fractio equal or exceed the transaction required, then typically end up with paying

your change. This one-way process of cutting off pieces of a bitcoin would steadily. A holding of one bitcoin would end up being constituted of maybe thousands of differently-sized fractions.

If that's the case, that will make transactions increasingly messy: you may consolidate a large number of inputs for one payment. In turn, that will lea block chain file growing faster and faster.

Did I miss something?

Michael Nielsen permalink

This is not a problem. Yes, some transactions will "fragment" b other transactions undo fragmentation. For example, a 5-input transaction will reduce fragmentation.

This sounds a little complex for the user, but in practice, good software will make this invisible. You simply say "I want to sen such bitcoins to so-and-so address", and all the details of com transactions will be taken care of. In this sense it's actually ea cash, where we deal with the fragmentation / de-fragmentatio the time (i.e., finding the right combination of bills and coins to service, and then dealing with the resulting change).

Amit Prakash permalink

Amazing article. Looking forward for more on similar topics – may b some explain for tor.

Edward permalink

Very informative article; thank you. I was wondering:

1) With regard to transaction fees, I assume it is up to the first miner to suc validate a block to determine if your offered fee is large enough to be inclu next iteration of the block chain. Additionally, I assume you have to determ you're willing to pay well in advance of the next chain being verified. If this and we fast-forward to 2140, won't all miners (assuming the computing po concentrated) be incentivized to take ANY transaction fee no matter how s instance, if I have .001% of the computing power I should, on average, val block once every 2 years (hopefully the math's right, it would be infrequent regard) and, considering (assuming?) there would be no marginal cost to i transaction (or 1,000 transactions) offering me only $.1 equivalent in trans can't think of why I wouldn't validate that transaction. As such, even if ther few big players who controlled say 2/3 of the computing power, on averag blocks would be validated by a smaller player who wouldn't care about pri people already wait nearly an hour to finalize a transaction likely they'd be wait an extra ~30 minutes and, as such, the big miners would likely just lo thresholds pretty substantially; Wouldn't this create an odd prisoners' dilen situation? Maybe I'm missing something but it strikes me that this would e fees forcing people out of business until computing power concentrated ar oligo(mono)poly of miners who could exercise sufficient pricing control wh pointed out, would probably create integrity concerns.

2) I believe I've previously read somewhere that there was a price thresho to incentivize mining at different levels of complexity – for instance, if bitco tomorrow, would miners continue to mine or would the marginal cost of run equipment outweigh the reward (ignoring fees)? If that's the case and a sc that occurs, does bitcoin grind to a halt or will some miners shut down or s less expensive equipment as they did when bitcoin was in that price range this sort of boils down to whether the use of high cost computing equipme function of competition (and price) or problem complexity?

Both are hypothetical but I was curious to know if you (or anyone) had con these questions.

Thanks again for the article.

Edward permalink

(I think) I figured out the answer to #2 – I was unaware of how was calculated.

JimmyWeg permalink

Great article! I'm working on a case and see that the bitcoin user employe a dozen different applications: Anoncoin, Phenixcoin, Primecoin, etc. I tak protocol is the same among the clients, though hash algorithms, proofs of and the like may differ. From what I understand, if I use XPMs and want to something from a vendor who accepts BTCs, I have to go through some b exchange facility to complete the transaction. If that's correct, and conside U.S. in a vacuum, isn't it like we're all carrying around a different brand of (USDs, Yen, Pounds, etc.) and have to exchange them almost every time buy something? It's easy with credit cards, but I don't see a similar approa bitcoins. Thanks!

David permalink

"I don't understand why double spending can't be prevented in a simpler n using two-phase commit. Suppose Alice tries to double spend an infocoin Bob and Charlie. The idea is that Bob and Charlie would each broadcast t respective messages to the Infocoin network, along with a request: "Shoul this?" They'd then wait some period – perhaps ten minutes – to hear any r who could prove that Alice was trying to double spend. If no such nays are provided there are no signs of attempts to disrupt the network), they'd ther transaction. This protocol needs to be hardened against network attacks, b to me to be the core of a good alternate idea. How well does this work? W drawbacks and advantages does it have compared to the full Bitcoin proto

I think something along these lines is planned:

“On top of all that is a long list of new features and improvements I’d like to
into a 0.9 release; the highest priorities on my wish list are:
1.“First double-spend” relay and detection. Detecting attempted double-sp
soon as possible is great for low-value, in-person transactions, and we sho
more to support that use case.”

https://bitcoinfoundation.org/blog/?p=204

David permalink

Great article. I will use it as a self-study tutorial.

In your next instalment, could you give a broad description of where the pr
actually to be found (is it a particular piece of software?), how can it be cha
can change it and indeed to what extent is it capable of being changed? T
important questions because they go to the ability of Bitcoin to evolve and
but it is very hard to find any good general account of these issues.

Best regards

David

Nitan Shalon permalink

If Satoshi Nakamoto already made two patches to Bitcoin, what’s stopping
from making another patch right now that destroys Bitcoin?

Arthur Colle permalink

“Of course, after Alice has published her message it’s possible for other pe
duplicate the message, so in that sense forgery is possible. ”

How can someone forge the message without Alice’s private cryptographi

Dave permalink

Hi. Great article. Thanks for putting it together.

I am still having one big problem — and I feel like I must be missing some obvious. You wrote:

“Suppose Alice tries to double spend with Bob and Charlie. One possible a for her to try to validate a block that includes both transactions. Assuming percent of the computing power, she will occasionally get lucky and valida by solving the proof-of-work. Unfortunately for Alice, the double spending immediately spotted by other people in the Infocoin network and rejected, solving the proof-of-work problem. So that’s not something we need to wo

Who is going to be looking to reject it, and what does that even mean? If a party (Alice) manages to complete a block that contains transactions that a fact, valid then what? Do other miners check them before building on top c block? And, if not, then what does it mean for others to ‘spot’ them.

This has been bugging me for days! Your thoughts would be greatly appre

Michael Nielsen permalink

Anyone with a copy of the block chain is not going to accept a block which has an obvious attempt to double spend in it.

Dave permalink

Thanks!

So that means that miners examine ea choose to build on it…? And, if that’s tr inspection built into the incentive struct become invalidated if down the line sor

a block with a double spend? (That wo
that I have seen described.)

Michael Nielse

Checking for a c

Daniel Park permalink

That's a long detailed process of bitcoin above which was a great read. I d
understand near the end, but I definitely get the gist of it.
I don't perfectly understand the relationship between transactions and bloc
only required when ppl try to cheat the system?) If so, the money earned b
essentially is imaginary and something that only exists within trust that bitc
to continue to work. I guess there are two cases:
1. as miners get bitcoins, the value of bitcoin comparison to other currenci
(This is probably the case)
OR
2 as miners get bitcoins, if it comes to a point where everyone wants to co
bitcoins to real currency, it's not going to equal each other; bitcoins -> conv
bitcoin exchange rate -> not enough real money.

Also, why assume every 210,000 blocks is occurs every 4 years? is this a
assumption based on bitcoin flow so far? Wouldn't every 210,000 blocks c
often if there is more flow?

Steffen permalink

If everybody would like to exit Bitcoin at the same time the pric
collapse. The current speculation is though, that the opposite

Many people try to buy bitcoins for the fiat money. How many?
here http://fiatleak.com.

With regards to why 210,000 blocks are created in roughly fou
The network difficulty is set so that only six blocks per hour ca
created. Roughly every 10 minutes a new block enters the blo

Let's do the math: 4 years ->1461 days * 24 hours -> 35,064 h
blocks -> 210,384

Jozef permalink

Fairly good explanation although some important things missing. One mist

"So if we want the output hash value to begin with 10 zeroes, say, then Da
need, on average, to try 16^10 (approx 10^12) different values for x before
suitable nonce."

I think it should be 16^9 on average. At worst it is 16^10.

Michael Nielsen permalink

No, the average number of trials is 16^10.

(If we repeatedly sample from a Bernoulli random variable witl
p of a success, then the expected number of trials to success
\sum_{n=1}^\infty n p (1-p)^{n-1}, which is easily shown to be
probability p in the proof-of-work is 16^{-10}, so the expected
trials to success is 16^{10}.)

What important things are missing from the explanation?

Lex permalink

i have alot of questions about bitcoin and ive done alot of searching, this e
has to be the one that has got me the closest to the answers ive been look
unfortunately im still falling short of understanding alot of the basic, i do wa
question here which seems to be the most important question for me. In c
science terms what is a “bitcoin” there were points in this article where it s
close to specifying this, although i may have actually missed the answer, s
“bitcoin”?? is it a unique value? if i have 100 bitcoins what is it that i really
unique values ? or is it simply a value that was “said” to be giving to me ar
assigned a value, sort of like the 5th entry in a ledger, in the genesis block
“said” 50 bitcoin was givin to whom ever,50 bitcoin was given to bill and w
that 50 1234567.. ??? overall im still searching for clarity on the fundamen
bitcoin from a computer science view

david lee permalink

Michael,
We are organising World Bitcoin Conference on 24 & 25 March 2014 in Ma
Would like to invite you to speak on Bitcoin protocol. Could we have your e
address to send the invitation.

David permalink

Hi – what a great write-up! As for your first Author’s question – There are p
several less complex methods for confirmation but there is inherent securi
current approach which appears to organically solve the problem without r
on factors and layers outside the network itself.

momingle permalink

There is one part that I am not sure I understand. For example, let say mir
his queue transaction A, B, and C to validated on a new block. Is it possibl
will have transaction B, C and D in his queue (but not A) that he will valida

new block? Assuming both solve the puzzle. Now both transactions B and two different blocks. Will both blocks got accepted?

Nagaraj Hubli permalink

Thanks for the write-up, it helped my a lot in understanding the underlying Bitcoin protocol. Can’t wait for the next in series.

elvin permalink

The protocol rules in the bitcoin wiki are ambiguous
when an incoming block designates as its predecessor a block somewher
the main branch, what happens exactly ?
the wiki can be interpreted in two ways :
1. nothing at all – the incoming block is NOT added as the beginning of a s
2. the incoming block IS added as the possible beginning of a side branch
any verification for the moment. If that is the case, then most blocks in the
branch should have many “brothers”

elvin permalink

I have verified that the correct answer is 2.
I have also found what limits the number of “brothers” for a blc
1. miners stop mining as soon as they receive a valid node an
no new blocks are sent
2. their neighbours do not relay the blocks they might have se
meantime
Sorry for asking

fivebuckchuck permalink

Thanks for the write up ..

How do transactions get organized into blocks ? Are nodes broadcasting transacations or blocks ?

Number of transactions in a block ; is that hardcoded in the protocol ?

in your example of h(“hello world”|nonce) , is “hello world” a unique transa whole block ?

Rich permalink

“Suppose Bitcoin mining software always explored nonces starting with x = 1, x = 2,\ldots. If this is done by all (or even just a substantial fraction) of B miners then it creates a vulnerability. Namely, it’s possible for someone to their odds of solving the proof-of-work merely by starting with some other larger) nonce. More generally, it may be possible for attackers to exploit ar systematic patterns in the way miners explore the space of nonces….”

This is incorrect:
Because the block hash is dependant on the contents of the block. For the possibility of a miner improving his odds through this method the miner mu mining the exact same block as someone else including not using his own coinbase and transaction fees to go to. Removing the entire incentive for r

Trackbacks and Pingbacks

1. How the Bitcoin protocol actually works | Enjoying The Moment
2. Understanding the Bitcoin protocol | Minsheng 民生
3. Science & Technology Roundup: #3 | Great Wall of Numbers
4. mid-day-ish links dump | imagine art -||+ there about?mid-day-ish links dump
5. How the Bitcoin protocol actually works | DDI | geistwc
6. How the Bitcoin protocol actually works | DDI | Mark Solock Blog
7. How the Bitcoin protocol actually works « adafruit industries blog
8. The Bitcoin Protocol - iamronen

9. Links 12/8/13 « naked capitalism
10. Bitcoin for dummies – Author walks users through how Bitcoin actually works | My Blog
11. Want to Spend Bitcoin in France? (Caveat Emptor) | FrenchNewsOnline
12. Robert McGhee » December 8th
13. ¿Cómo funciona realmente el protocolo del Bitcoin? [ENG]
14. BitCoins
15. Bitcoinizzle For Shizzle | BubbleCoin
16. How the Bitcoin protocol actually works | Blog of Leonid Mamchenkov
17. Starting of “Gift Economy (reputation based)” | l'ambizione dell'artigiano
18. Headline News from PaymentsNews.com – December 9, 2013 - ResidualMarketplace.Com Residual Listings for Free
19. How the Bitcoin protocol actually works « The Bitcoin Channel
20. Bitcoin Explanation « Electronic Padlock
21. Bitcoin Explanation
22. Bitcoin | Thoughts on computing
23. Links: Mandela and Great Leaders; WTO deal in Bali; How Money Works
24. How Does Bitcoin Work? ← Clinton Pavlovic
25. Following Bitcoin Believers is still an act of faith | Digital Trends
26. Virtual Mining Bitcoin News » Bitcoin Believers aren’t wrong, but don’t mistake their faith for
27. Open thread: Bitcoin and similar currencies
28. How Bitcoin Actually Works
29. Bitcoin Speculations | Aleph
30. How the Bitcoin protocol actually works | The Freedom Watch
31. How the Bitcoin protocol actually works | DDI - The Ripe Group
32. Explaining Bitcoins | kef's blog
33. A Detailed Explanation of the Bitcoin Protocol – Glimpse From a Height
34. Friday Links | Meta Rabbit
35. Bitcoin
36. Virtual Mining Bitcoin News » Bitcoin: Open source money whose time has come
37. Bitcoins, Cash and Gold | GusBook
38. Plaidoyer pour le Bitcoin | ComplexitésComplexités
39. Occupy BitCoin
40. How to not get rich from bitcoin | Turing's Invisible Hand

41. Bitcoin: The Economic Frontier of a Digital Age | The American Statesmen
42. Why You Need to Own Bitcoin | Winning Bitcoin
43. Bitcoin: a bit risky | Learn About Finance Careers
44. Various well written articles/sites | GSR8 Blog
45. Bitcoin transactions explained; How the Commodore 128 was … | BitCoin Razor
46. Digitopoly | Time for a little Bitcoin discussion
47. How and why BitCoin will plummet in price
48. How the Bitcoin protocol actually works and what that means for us | Helping Public Market
49. | Explaining bitcoin in a bar
50. Cryptocurrency Cat-and-Mouse games in China | Great Wall of Numbers
51. Random thoughts: Bitcoin
52. Info On Bitcoin | Curly's Blog
53. My Favourite Bitcoin Bookmarks | Webs Wonder Design - Webs Wonder Design
54. Facecoin In More Detail | Rob Myers
55. Fed Banker Tries Criticizing Bitcoin, Ends Up Perfectly Describing The Fed – Bitcoin Magaz
56. Bitcoin 2.0: Distributed Corporations, Derivatives, and Information Markets | The Ümlaut
57. The Difference between Bitcoin and OpenTransact | OSCurrency
58. j.r.mchale : The Nuts-and-Bolts of the Bitcoin Protocol
59. Fed Banker Tries Criticizing Bitcoin, Ends Up Perfectly Describing The Fed
60. Sécurité Bitcoin #1 : Introduction | Le Comptoir Sécu
61. De week in 375 woorden « Bits of Freedom
62. Bitcoins the hard way: Using the raw Bitcoin protocol | Bitcoin
63. On Bitcoin | Emergent Chaos
64. bitcoin critical mass/explosion, wheres it going? | Turing Machine
65. Bitcoin Beyond Money | SoshiTech – Social Media Technology – Soshitech.com
66. Bitcoin reading for developers | Coffee Spoons of Code
67. The Homebrew Litecoin Mining Project | Record important events of digital currencies
68. Complete Dogecoin Guide - Introduction into Mining - Everything Doge!
69. How the Bitcoin protocol actually works | DDI | Arca
70. bitcoin | jamesmath
71. Fun economic reading | MuddyHorse Farm and Tech
72. Bitcoin: have some of us been had? | Jim Bennett
73. “The Bitcoin Protocol” Infographic |

74. I don’t get Bitcoin… but I want to… | StartupCafe
75. Daily BTC – PBOC edition : Token Intelligence
76. Enter The Blockchain: How Bitcoin Can Turn The Cloud Inside Out | TechCrunch
77. Quora
78. Bitcoin Family Tree | Randolog

Comments are closed.