Bitcoin: An overview

Author: Pullinger, Jonathan

Version: 3.01a

Contents

Bitcoin Executive Summary

Bitcoin (BTC) smashed an all-time high of $20,000 USD and subsequently dropped to as low as $5,795 within a few months. It's incredibly important to not get lost in the pandemonium and to stay informed about how Bitcoin is progressing technologically. Anybody considering buying Bitcoin should at the very least learn two things:

The history of the technology behind Bitcoin
What lies ahead in Bitcoin's future.

As Bitcoin expert Andreas M. Antonopoulos says, "Invest in education instead of speculation."

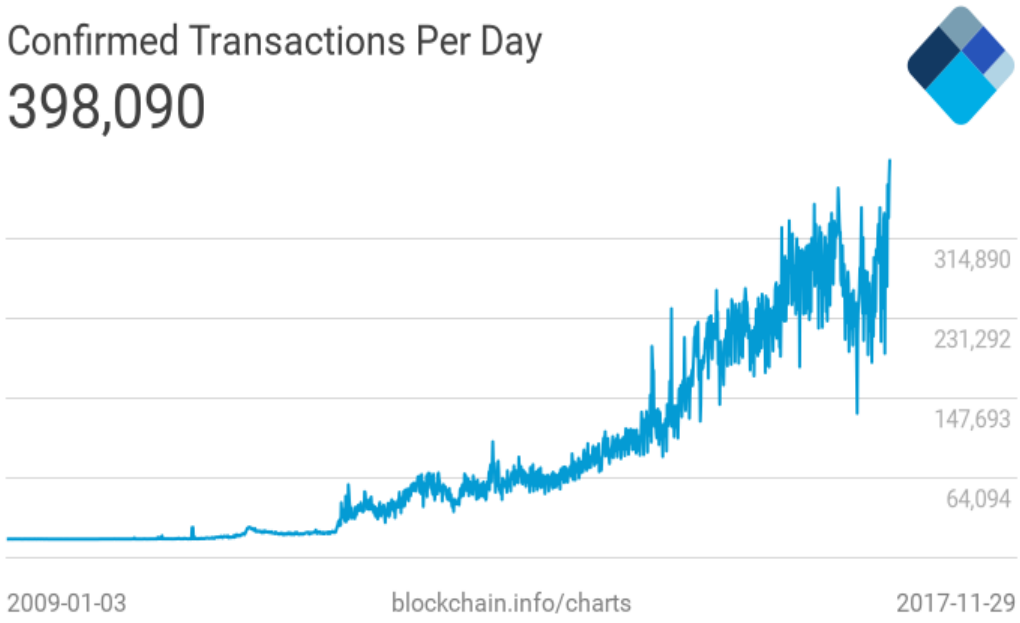
With Bitcoin and its underlying blockchain being such incredibly new technological concepts, it can seem daunting at times to try and research and understand its underlying technical details. This article is written in an effort to highlight the scalability problem Bitcoin faces, and what expected or proposed solutions to that problem are. There are some really exciting ones out there that this article discusses! Let's begin with the scalability problem Bitcoin faces.

The Transaction Throughput Problem

When Bitcoin was first introduced to the world, its creator Satoshi Nakomoto described Bitcoin in the Bitcoin whitepaper as “A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.”

One of Bitcoin’s fundamental values was instant and secure peer-to-peer payment transactions. Now, more than ever, Bitcoin is emerging as the prevailing cryptocurrency in the global market, with a 1,200%+ increase in value over the last year alone.

Because of this unprecedented growth, the number of transactions on the Bitcoin blockchain has also increased, with up to 400,000 transactions per day being conducted. This rapid increase in transactions is posing to be a serious scalability problem for the blockchain, with over 90,000 transactions being backlogged as unconfirmed at the moment.

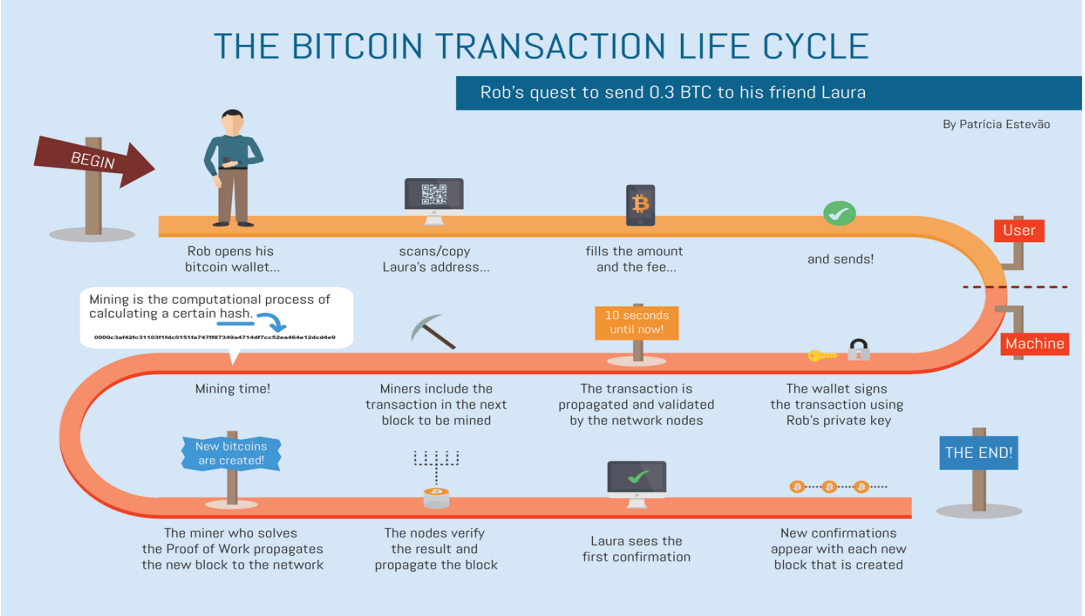


In order to understand why transactions are being backlogged, Bitcoin transactions must first be explained.

Every time a user sends a Bitcoin transaction from his or her wallet to another, the transaction is added into the memory pool (mempool), which is essentially a pool of all unconfirmed transactions in the Bitcoin network. This pool is upheld by individual memory pools on machines that also hold a copy of the blockchain ledger, called nodes.

From the mempool, miners select transactions that they want to verify. Once miners validate a transaction (i.e. confirm that the sender actually has enough bitcoins to send to the receiver), they add it to a new block, which is eventually published to the

blockchain. Other nodes then iterate through this newly published block's transaction to ensure the block is valid, before accepting the block as a part of its ledger.



Let's calculate the throughput of transactions:

The median transaction size approximately 250 bytes
A block's size is limited to 1MB (1000000 bytes)
Thus, a block holds around 4000 transactions (1MB divided by 250 bytes)
A block can only be published to the blockchain once every 10 minutes on average (600 seconds).
4000 transactions (at most) are published every 600 seconds, at a rate of 6.66 transactions / second

With over 90,000 unconfirmed transactions in the mempool, how does a miner select which transactions to verify? Transaction fees! The sender of a transaction has the option of adding a custom transaction fee to its transaction intended for the miner, incentivizing a miner to select the transaction and have it verified faster. Miners will select the transactions that have the highest fee attached to them to maximize profits.

Theoretically, you can send a transaction with no fee. But if there are transactions that have fees higher than yours in the pool, why would yours ever get picked?

As Bitcoin's user base grows, so does the average transaction fee. At most, there are only 7 transactions that are processed every second and everyone wants to get their transaction verified first. At the moment, the average transaction fee is approximately $3.58 USD. This fee is certainly not ideal — if you want to send your friend a couple of dollars worth of bitcoin, you may end up spending more in transaction fees than the transaction value itself! Therein lies the problem, and if all else remains equal, transaction fees can be expected to rise due to the transaction bottleneck.

Solving the Throughput Problem

A proposed solution to this bottleneck that has brought great controversy to the Bitcoin community is to simply raise the block size from the original 1MB limit, thus allowing more transactions per block. Every time the block size is increased in the chain, a hard fork is required, meaning an entirely new copy of the chain must be created, therefore requiring consensus from the Bitcoin community.

Because millions of people use Bitcoin, gaining consensus is difficult and efforts should be made to avoid it. Furthermore, although the block size can be increased enough to accommodate the current backlog of transactions, as Bitcoin's user base continues to grow, there will eventually be another backlog of unconfirmed transactions, so another block size increase will be needed, and subsequently another hard fork.

So why don't we just make the block size large enough to ensure the throughput will never be a bottleneck, no matter how many people are using it? First, the mathematics of a block size even remotely large enough to handle mass adoption are impractical and will restrict mining to incredibly powerful machines that only large corporations will be able to maintain, introducing an element of centralization.

Furthermore, recall that once a block is mined, all other nodes must validate the block before accepting it. If the block size was incredibly large and somebody were to publish an invalid block, nodes would waste a large amount of time attempting to validate the block before discarding it as invalid and moving onto the next block. A denial of service attack can essentially be orchestrated by repeatedly publishing insanely large invalid blocks to the network, stopping valid blocks from being processed for a long period of time.

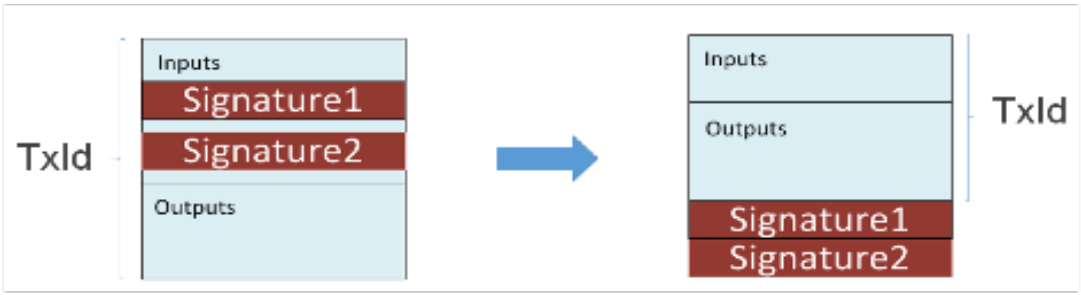
As stated by blockchain pioneer Nick Szabo in this interview, the small block size acts as a

technical security parameter to prevent network flooding. If we can't increase the block size, what can we do? Luckily, there are several solutions in the works that are expected to be deployed in order to solve this issue.

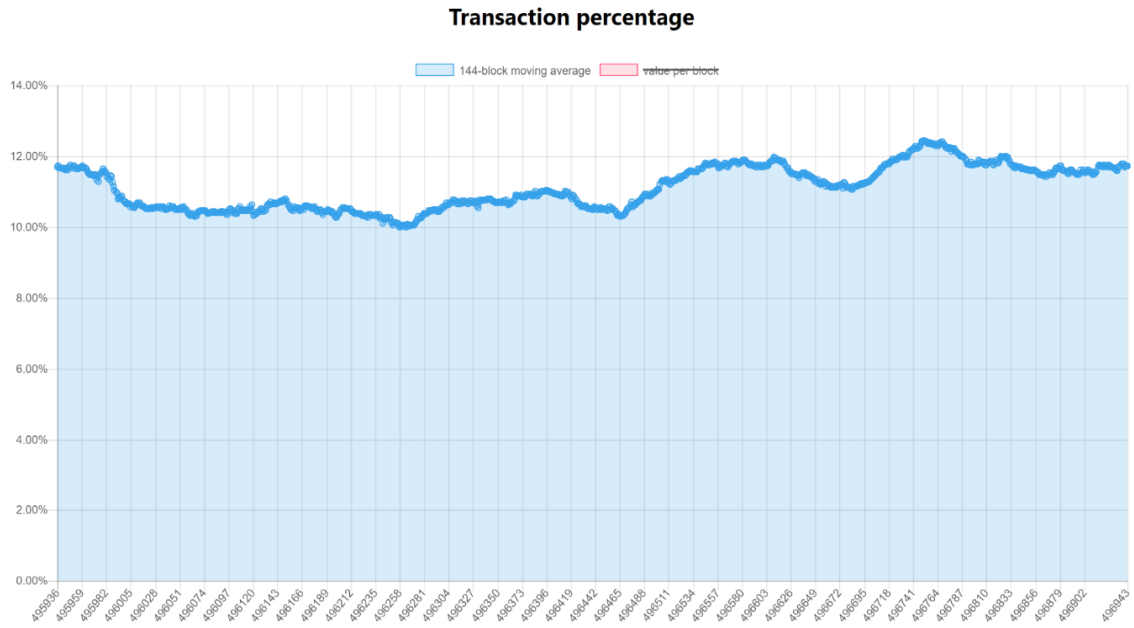
Segregated Witness (SegWit)

Segregated Witness (SegWit) has actually already been implemented into the Bitcoin network, as of August 2017. It's a fundamental network change that modifies the format of transactions, essentially slimming them down in size, and allowing more transactions to be fit into a block which increases throughput. SegWit is considered a soft fork, meaning it is completely backwards compatible with existing Bitcoin protocol, although nodes and wallets must upgrade to take advantage of all SegWit features.
Each transaction has a signature from the sender, or in other words, witness data; this is usually the largest part of the transaction. This data is not actually necessary to verify the transaction, and so SegWit moves this data to the end of the transaction, segregating it. If this transaction is sent to a legacy node (a node that has not upgraded to SegWit), the node strips the witness data off the end of the transaction before inserting it into a block, thus reducing the overall transaction size and saving space.

The added benefit of this is that nodes can no longer modify the witness data, changing who the transaction was from, an ability nodes previously had. This makes way for the implementation of multi-layer solutions that we'll discuss soon. Users also save on transaction fees, as they're usually calculated per transaction byte, and SegWit reduces total transaction size.



Furthermore, SegWit changes the definition of a block: instead of a block being defined in terms of bytes, its now defined in terms of "weights"; a block can have a maximum weight of 4,000. Legacy transactions have a weight of 4, while SegWit transactions have a weight of 0.25, thus enabling a block to contain many more SegWit transactions and have a slightly higher size (approximately 2 megabytes at most). Nodes must upgrade to SegWit to follow this definition, and wallets must incorporate SegWit in order to send SegWit transactions. As a result, SegWit adoption has been slow, accounting for only 12% of current traffic.



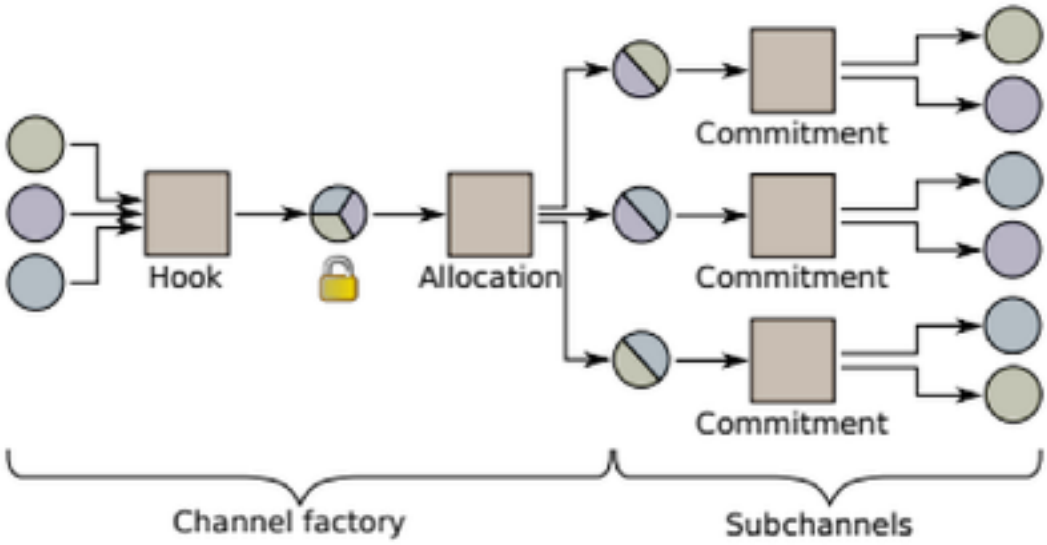
Multi-Layered Solutions

As it stands, the Bitcoin blockchain isn't very feasible for micropayments. If you want to buy a $2 cup of coffee, you're probably going to pay more than $2 equivalent of BTC in transaction fees, and the transaction won't be confirmed instantly — you must wait for the transaction to be published in a verified block on the chain, which will appear within 10 minutes at best.
Second and third layer solutions are networks layered on top of the Bitcoin blockchain that enable users to send several transactions of small amounts of Bitcoin almost instantly, with no transaction fees.

The Lightning Network is this layered network currently in development expected to alleviate Bitcoin's scaling problems. This network consists of two additional layers and enables users to open direct channels between each other to send an effectively unlimited number of payments to each other in an instant manner.

The Second Layer

A user joins the second layer network by conducting a transaction on the blockchain that declares the user is committing a certain number of bitcoin to be used in the layered network. The user then joins a group of nodes that are interconnected with one another, called channel factories. These nodes essentially uphold a lobby of individuals that potentially want to conduct transactions with one another. Channel factories then enable an unlimited number of micropayment channels to be created on the third layer (hence the name factories) between individual parties.



The Third Layer

Micropayment channels are set up to ensure direct payments between two users on the third layer. Because the blockchain is no longer present in this layer, it cannot be used to validate transactions and ensure one party paid the other. Instead, smart-contract technology is employed, such as multisig addresses, meaning addresses that can be signed off by multiple users to enable the movement of funds, and hashed time-lock contracts, which are cryptographically secure automated contracts that lock funds for a certain period of time to ensure one parties that cannot cheat with another. These technologies eliminate the need for trust between users that are connected in micropayment channels.

Here is an example of how a Lightning Network micropayment channel works:

1. Alice wants to dedicate 1 Bitcoin to a micropayment channel between Bob. She declares this 1 Bitcoin to be used in a commitment transaction on the Bitcoin blockchain. This 1 Bitcoin is then locked up in a multisig address that both parties can sign off on if they want to close the channel. This address is secured with a hashed time-lock contract which states, “Alice has 1 BTC and Bob has 0 BTC, to be released in one hour”. This means the 1 Bitcoin Alice has is locked for 1 hour after which it will be returned to Alice and published to the Bitcoin blockchain once more.

2. Alice then decides to give Bob 0.1 BTC. This transaction is logged with a new hashed time-lock contract stating “Alice has 0.9 BTC and Bob has 0.1 BTC, to be expired in 50 minutes”. This contract has an expiry time of 50 minutes, meaning it will be published to the blockchain before the original contract stating Alice has 1 BTC. Therefore Bob instantly knows he has the 0.1 BTC because this new contract will be published to the blockchain before the original contract, essentially making the old contract invalid.

3. Once the full hour passes, the micropayment channel closes and the final balance between Alice and Bob is published to the blockchain. If Alice and Bob want to continue making transactions, they can extend the expiry time of their channel for as long as they want. If one of them wants to close the channel early, one of them needs to sign off on the original multiset address that the Bitcoin is stored in.

The network enables transactions to route itself to its final destination by using other connected users in the channel as intermediaries. This can happen even if a direct connection to the desired user isn’t able to be sought the current micropayment channel. For example, if Alice has a channel open with Bob, and Bob has a channel with Mark, and Alice wants to send Mark some Bitcoin, the network can route the payment to Mark through Bob, all while ensuring no party has to trust another.

The implementation of lightning network transactions and their trust-less nature can get incredibly complex, and is best explained by the Lightning developers in this conference. Ideally, a user will only create a commitment transaction to the secondary layer very rarely because he or she will remain in the layered network for prolonged periods of time to conduct most of their day-to-day transactions. Once a user wants to exit this multi-layered network, a settlement transaction is made on the blockchain declaring the user’s final Bitcoin balance after all of the second- layer activities. This reconciles their total Bitcoin balance on the blockchain after comparison with the original commitment transaction. In total, only two blockchain transactions are made in order to let the user to conduct a limitless number of transactions for free on the second layer.

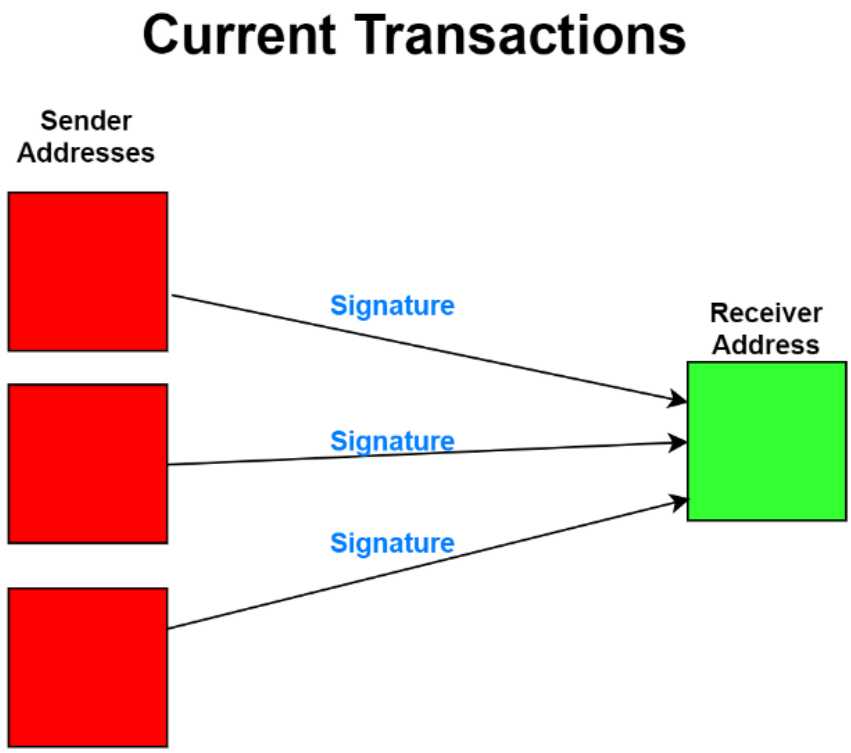
As mentioned previously, SegWit paves the way for the lightning network because it removes nodes’ abilities to modify witness data, which is what is used to identify a user’s entry into the second layer. If the user’s commitment transaction can’t be found because the witness data referring to the user was changed, there is a greater level of difficulty involved when trying to reconcile the user’s settlement transaction.
The second layer of the lightning network involving channel factories was very recently introduced in this whitepaper. It is still under heavy development, so a lot of its concepts are explained abstractly. However, the network is poised to launch in 2018 and will be by far the biggest improvement in transaction scalability thus far.

Schnorr Signatures

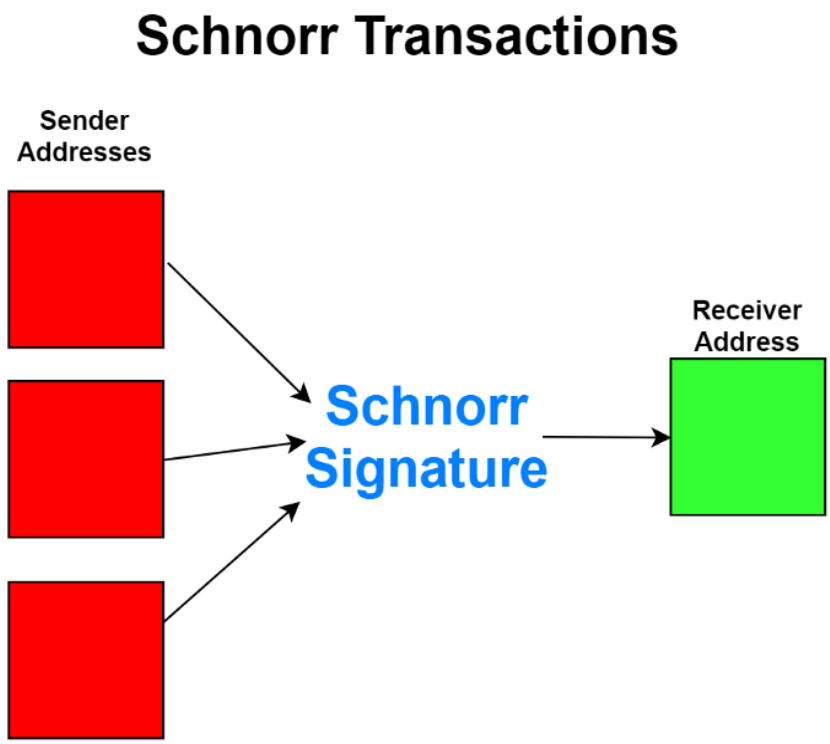
When a user sends a Bitcoin transaction, the inputs of the transaction (the amount you're sending) is calculated simply by retrieving from the blockchain the total unspent amounts of Bitcoin you previously received. So for example:

- Starting with an empty wallet, I receive 1 Bitcoin in transaction #1, and then another 1 Bitcoin in a separate transaction #2
- I now want to send 2 Bitcoins in a transaction. There will be two inputs to this transaction: transaction #1, and transaction #2, summing up to 2 Bitcoin

Under the current algorithm for generating signatures (Elliptic Curve Digital Signature Algorithm), each input requires its own signature. This increases the total transaction size and therefore increases the transaction fee.



Schnorr signatures are an alternative and more efficient way of storing signature data in transactions. All inputs are accumulated and then stored as a single signature by utilizing the Schnorr algorithm, which greatly saves space in a transaction and further helps increase transaction throughput by allowing blocks to store more transactions on average.



Schnorr signatures can be also be used to aid Bitcoin's advancement in privacy by benefiting CoinJoin transactions. CoinJoin is a method of introducing anonymity to Bitcoin transactions. It works by pooling transaction inputs together with other peoples' transactions when making a payment to a receiver. When payments are pooled, it becomes difficult to track which user sent what input, effectively making them anonymous. However, CoinJoin transactions have increased fees due to a higher number of inputs in a single transaction resulting in a higher number of signatures. Utilizing Schnorr signatures would enable all signatures in a transaction to be compressed into one, saving greatly on transaction fees and encouraging the use of CoinJoin.

Furthermore, Schnorr paves the way for complex multisig transactions which require signing off from multiple parties; no matter how many parties' signatures are required for a transaction, all the transaction needs is one Schnorr signature. Schnorr signatures are only now a possibility because of the implementation of SegWit; because signature data can't be modified by third parties, it can now be used to effectively create a Schnorr signature.

MimbleWimble

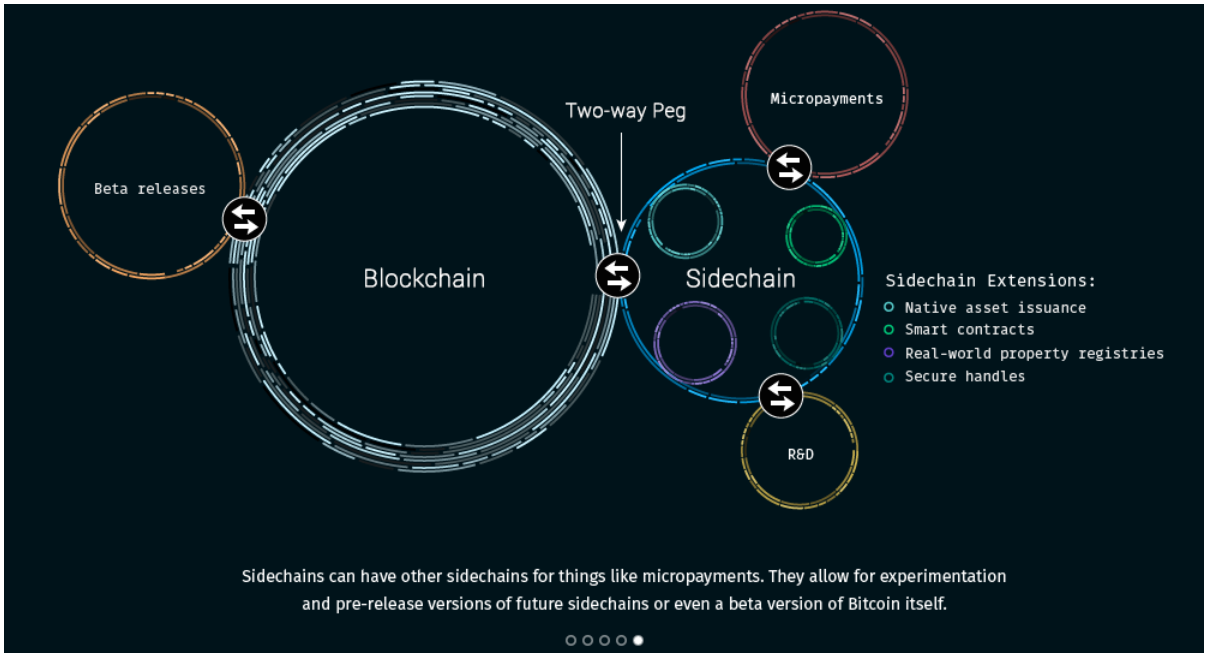
MimbleWimble is a radical but incredibly powerful proposed change to Bitcoin architecture that was anonymously introduced through this whitepaper in July 2016. Named after the tongue-tying curse from the Harry Potter series, its aim is to remove transactions entirely from blocks. Under MimbleWimble, transactions consist of nothing but input amounts, output amounts, and a signature. The signature of the transaction can only be decrypted by the receiver, and so transaction verification is left to the receiver.

By extension, blocks consist of only the list of all transaction input amounts of all transactions, all transaction output amounts, and their corresponding signatures. Blocks can then be merged seamlessly with previous blocks as they're nothing but pairs of input and output amounts. Nodes then have the ability to cryptographically ensure that transactions in blocks do not create extra bitcoins (i.e. their net difference between inputs and outputs in blocks is 0) without having to decrypt transactions. This removal of transaction storage grants complete anonymity to all users by stripping away the ability to generate transaction history. Furthermore, with blocks only containing the unspent transaction outputs (meaning the number of Bitcoins that have been received in an address but not moved out yet), the blockchain size can be reduced by over 60% according to the whitepaper. This reduction in size means that in order to validate a MimbleWimble blockchain, nodes will only need to look at the set of unspent transaction outputs instead of the entire set of transactions, which will exponentially increase performance.

The mathematical details of MimbleWimble are outside of the scope of this article, but are explained in detail in the whitepaper. Although MimbleWimble presents some clear advantages and technical breakthroughs, its implementation requires the removal of Bitcoin's Script system that much of the existing architecture relies on. As a result, MimbleWimble's implementation on the Bitcoin blockchain is not technically feasible.

However, there are proposals for MimbleWimble to exist as a sidechain. A sidechain is a separate blockchain directly attached to the Bitcoin blockchain through the use of a two-way peg. This peg enables assets between the two chains to be exchanged and "pegs" the value of the sidechain asset to the value of Bitcoin. In this setup, users would be able to exchange Bitcoins for MimbleWimble coins, conduct completely private and rapid transactions on the MimbleWimble chain, and then exchange their MimbleWimble coins for Bitcoin whenever they please.

In fact, a group of developers are already in the process of developing MimbleWimble as a separate cryptocurrency called GRIN; it was recently deployed on a test network and may be launched in the near future.



Rootstock

Rootstock is for whatever reason one of the less talked about advancements in Bitcoin technology but is by far one of the coolest. Rootstock is described as “the first open-source smart contract platform with a 2- way peg to Bitcoin that also rewards the Bitcoin miners via merge-mining, allowing them to actively participate in the Smart Contract revolution.”

Much like MimbleWimble, Rootstock is being developed as a sidechain solution to the Bitcoin blockchain. Its fundamental value lies in its focus in smart contracts. Rootstock aims to be a Turing Complete (completely programmable) smart-contracts platform that will be backwards compatible with Ethereum’s virtual machine. This means that Rootstock will be able to execute any smart contracts developed for the Ethereum platform and have smart contracts developed for its own platform.

Rootstock aims to implement this versatile smart-contract functionality all while leveraging Bitcoin’s comparatively dominant userbase and value by acting as a two-way pegged sidechain. It is also being designed to be secured by the existing Bitcoin mining network, therefore not needing to incentivize miners to secure its own blockchain. Rootstock also aims to tackle the transaction scalability problem by implementing its own version of a multi-layered solution called Lumino. With this, it may be able to accomplish up to 20,000 transactions per second. Rootstock is aiming for a launch by the end of 2017. Overall, the platform aims to fit in perfectly alongside Bitcoin and if its claims hold true, it will undoubtedly bring unprecedented utility to the Bitcoin network.