The Original Bitcoin Protocol: What is it and why does it matter?

Prepared by MNP

August 25th, 2021

About MNP

MNP is a leading national accounting, tax, and business consulting firm in Canada. We proudly serve and respond to the needs of our clients in the public, private and not-for-profit sectors. Through partner-led engagements, we provide a collaborative, cost-effective approach to doing business and personalized strategies to help organizations succeed across the country and around the world.

Disclaimer

The opinions and conclusions identified in this independent review of the original Bitcoin protocol do not constitute a legal opinion, nor do they constitute a third-party attestation or statement of audit opinion as defined by the American Institute of Certified Public Accountants / Chartered Professional Accountants (AICPA/CPA Canada) rules and definitions. This assessment makes no warranty regarding the claims made about the current or future state of the Bitcoin ecosystem and its various implementations. This assessment was performed at a point in time based on resources available. Someone else with different access and different resources may have different results. If a third-party is to rely on this report, it is important to understand context, limitations of our analysis and our research approach, and therefore written consent from MNP must be arranged.

Purpose of this document

The purpose of this report is to determine by comparison which current Bitcoin implementation is the most accurate representation of the vision proposed by Satoshi Nakamoto ("Satoshi"), the pseudonymous creator of the original Bitcoin protocol.

Satoshi's original intention and vision for Bitcoin were established by examining source material related to the development of the original Bitcoin protocol and the Bitcoin network, either authored by or directly involving Satoshi himself. The original source material includes publicly available content such as online forum posts, emails, original source code for the first version of the Bitcoin software, and the Bitcoin whitepaper.

This report summarizes the context, methodologies used, and conclusions in the following sequence:

1. Setting the stage
2. Establishing Satoshi's original vision
3. Defining Satoshi's Bitcoin
4. Aligning current implementations to Satoshi's vision
5. Impact of Satoshi's vision once fully realized

Scope and approach

To perform the independent review of the Bitcoin ecosystem, the following areas were assessed for the Bitcoin Core ("BTC") and Bitcoin Satoshi Vision ("BSV") implementations in comparison to the original Bitcoin protocol:

- Capabilities
- Functional requirements
- Non-functional requirements
- Implementation attributes

The review was performed between February 8 and June 4, 2021. Accordingly, our results are based upon information available to us during that timeframe. All conclusions and key findings were based on the following assessment procedures:

- Defining key criteria of the "original Bitcoin protocol" (ruleset for the Bitcoin network) released by Satoshi.
- Reviewing publicly available content and documentation to assess each protocol implementation against assessment criteria.
- Comparing source code of Satoshi's original Bitcoin software version 0.1.0 to BTC version 0.21.0 and BSV version 1.0.7
- Conducting interviews with key stakeholders and subject matter experts on the current state of examined Bitcoin implementations

Limitations, boundaries, and exclusions

Source code for the Bitcoin software is limited to all versions from 0.1.0 until release of version 0.1.5, as there is evidence of larger community involvement in the development of the software thereafter. The software versions 0.1.0 through 0.1.5 are recognized as versions released by Satoshi.

For the purposes of this report, the scope was limited to examining the original Bitcoin protocol (circa 2009-2011) and contrasting it with BTC and BSV implementations as of March 31, 2021.

Due to time and material constraints, other forks of the original Bitcoin protocol were excluded.

The following review areas were not addressed as part of the independent review:

- Valuation - There is no intent to address the mechanisms that inform the market pricing of any digital cash mentioned in this paper. If mentioned, any forward-looking valuations are purely hypothetical and do not constitute investment advice.
- Reputation - There is no intent to address the public perceptions of any digital cash and/or the operators of the digital cash networks mentioned in this report.
- Previous forks - The analysis of previous forks such as Bitcoin Cash, BitcoinABC, and BitcoinXT are not included in this report.

For more information on cryptocurrency or for any questions regarding the information found in this report, please contact Hash Qureshi, Partner, Enterprise Risk Services at Hash.Qureshi@mnp.ca

Summary of findings

This report reviewed Satoshi's writings for the implementations of the Bitcoin network, the associated software and code, and the Bitcoin protocol. Source material for our review included Satoshi's whitepaper, emails, forum posts, and original code, which are all publicly available. These source materials were used to determine Satoshi's original purpose for Bitcoin.

Based on our review, Bitcoin was intended to be a transaction network for digital cash to compete as a global payment system. Current implementations (BSV and BTC) were compared against that original vision. Our findings indicate BSV is most representative of Satoshi's original intention for Bitcoin. We used an assessment framework and resulting criteria — including OpCodes, Bitcoin scripting, and protocol elements — to assess the protocols described in this paper.

Contents

1. Setting the stage

More than a decade has passed since the original Bitcoin whitepaper *Bitcoin: A Peer-to-Peer [P2P] Electronic Cash System*[1] was posted in October 2008 and the Bitcoin blockchain was launched in January 2009. Since then, the concepts of distributed ledger technology and blockchain have become a regular feature in global media and public discourse. Once a fringe idea, blockchain-based digital cash is now an accepted and well-recognized method for handling transactions and has spun out a new industry with many competing blockchains and distributed ledger technologies.

Given the rise in popularity of this new technology, it is important to reinforce the correct understanding of the blockchain movement by returning to its roots and examining the original vision of Satoshi Nakamoto, the pseudonymous creator of Bitcoin. With Satoshi's vision for Bitcoin in mind, the objective of this report is to examine the differences between two current competing blockchains that stem from the original Bitcoin blockchain, determine which blockchain best aligns to Satoshi's original vision, and more importantly, assess why that is important.

The original Bitcoin protocol[2] was proposed and — initially — solely developed by Satoshi. In the years prior to 2008 and 2009, Satoshi quietly developed the original code base of the Bitcoin software as proof of concept. After the protocol was revealed on the Metzdowd cryptography email list, the Bitcoin vision was supported by some early adopters. For a small period after the release of the software, Satoshi continued to play a key role in the development and maintenance of the Bitcoin project.

By 2010, a larger development community started forming around Bitcoin. The project would see many contributors to the Bitcoin software. Consensus is Satoshi last posted on the Bitcoin Forum[3] in December 2010.

For several years after Satoshi's departure from Bitcoin, the project evolved in response to what these early developers felt Satoshi wanted; they made decisions they felt would grow and promote a successful implementation of Bitcoin. The original Bitcoin protocol was modified and continued to evolve without Satoshi's involvement. Development was instead directed by developers who were not involved in Bitcoin's creation but who had their own views, intentions, and visions.

It was not long before early supporters of Bitcoin started to have a larger community supporting and transacting in the new digital cash. Like all communities, there were numerous opinions as to how the protocol should be developed and scaled (or not). With growing popularity and varied ideas on how the protocol should be developed, Bitcoin went through a series of forks[4].

[1] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[2] In computer science, a protocol is a set of rules or procedures for transmitting data between devices, such as a computer. For Bitcoin, these are the rules in which transaction data is transmitted across the Bitcoin network.
[3] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?action=profile;u=3;sa=showPosts
[4] In software engineering, a fork occurs when developers take the source code for an existing project and use it to create new software.

With a bit of history out of the way, and with the rise of alt-chains[5] deriving from the Bitcoin protocol, there are some important questions:

- What was Satoshi's original vision for Bitcoin?
- What attributes of the original Bitcoin protocol are key to realizing the full benefits of this vision?
- Which current implementation of Bitcoin protocol conforms most completely to these attributes and therefore is most likely to realize Satoshi's original vision?

The remainder of this report will focus on answering the questions above. *Section 2* aims to establish the validity of the source material and summarize findings regarding Satoshi's original vision. *Section 3* examines the original Bitcoin protocol and the key attributes for realizing the benefits of this vision. *Section 4* compares the BTC implementation and the BSV implementation to the vision as establish in sections 2 and 3. *Section 5* examines the potential future state if Satoshi's original intentions were met. Finally, *Section 6* provides concluding remarks. Additional detail is supported through Annexes.

2. Establishing Satoshi's original vision

2.1 Sources of information

To understand Satoshi's intentions regarding the Bitcoin project, it is important to establish a set of base truths. Several primary source documents can be reasonably relied upon to define exactly what Satoshi's vision for Bitcoin was. These include:

- The original Bitcoin whitepaper posted in 2008.
- Early versions of the codebase including known Satoshi's versions of the Bitcoin software.
- Known emails from Satoshi, as summarized on the Metzdowd Forum.
- Posts by Satoshi, as summarized on Bitcoin Forum and P2PFoundation.

2.1.1 The Bitcoin whitepaper

The Bitcoin whitepaper was the first published work associated within the Bitcoin project (distributed through the Metzdowd cryptography email list) and is a primary source that Satoshi had complete control over. It is possible to download the original version of the whitepaper from the Bitcoin blockchain itself.

In the whitepaper, Satoshi underlines his proposal that "[a] purely peer-to-peer[6] version of electronic cash would allow online payments to be sent directly from one party to another without going

[5] An alt-chain is the result of a new chain being created from a fork of the original chain.
[6] In computer networking, Peer-to-Peer (P2P) refers to a network in which each computer can act as a server for the others, allowing shared access to files and peripherals without the need for a central server.

through a financial institution… [with] a solution to the double-spending[7] problem using a peer-to-peer network.[8]'' Satoshi details nine key areas of his proposed solution in the publication including:

- How transactions work
- The operations of the timestamp server[9]
- The implementation of a proof of work ("PoW")
- How the network communicates
- The network participation incentives
- The methods to reclaim disk space
- The future state operations
- How transactions are sent through the network
- Privacy considerations

The whitepaper provides an entry point into the functional and operational requirements that make the original Bitcoin protocol work. In addition, Satoshi provides evidence supporting the validity of the solution to the double-spending problem that perplexed previous attempts at creating an electronic cash system. The whitepaper begins to outline what this technology could start to support and how different types of users would use the system . It also establishes a set of ground rules to enable development of the technology.

2.1.2 Other sources

Other important primary sources come from known online forum postings, writings, and emails that were exchanged between Satoshi and early Bitcoin enthusiasts. These digital communications offer a window into Satoshi's mind and paint a picture of how input from the digital cash and related communities was handled. There were mainly three topics of discussion that Satoshi was involved in with early Bitcoin adopters.

First, there were discussions about the validity of the proposed P2P electronic cash transaction system. These types of questions and responses provide a deep insight into how the original proposition was intended to function. These early conversations are mainly from the Metzdowd email list which consisted of the earliest members of Bitcoin's online community.

Second, there were questions around the different use cases of the proposed Bitcoin electronic cash system. There were community discussions about using Bitcoin for paid email services and vending machines. Satoshi often validated how Bitcoin could be used to handle these types of business transactions.

Finally, there were general tech support and administerial questions. These stemmed from users reporting bugs, having problems setting up the software, providing community members access to code so they could help translate the application into different languages, and users providing status

[7] In digital cash there is a unique problem wherein the currency could be spent more than once by an individual who understands how a particular system works at a programming level.
[8] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[9] A timestamp server cryptographically validates a digital signature took place at a specific point in time.

updates on new builds. These threads reveal how hands on Satoshi was in the earliest years of the original Bitcoin protocol.

2.1.3 Bitcoin v0.1.0 ALPHA

> *"I'm better with code than with words, though." – Satoshi Nakamoto, 2008*[10]

While this report relies on publications and written communications as primary source documents, it is just as important to examine the original source code of the Bitcoin software released by Satoshi, which represented his implementation of the original Bitcoin protocol. The earliest versions of the codebase for the Bitcoin software can still be accessed. This provides insight into the software, its potential functionality, and what the Bitcoin project was intended to become.

Several versions of the original Bitcoin software were released while Satoshi was recognized to be involved with the project. Satoshi first shared the pre-release version with a few select users of the Metzdowd group who contacted him personally. Like the whitepaper, this source code is likely[11] the original version created by Satoshi.

As the project took shape and gained acceptance, Satoshi started to add developers from the online community. Notably, Hal Finney[12] was among the first people to start contributing to the project. As the project evolved, the code was increasingly derived from community input rather than from Satoshi.

With that in mind, it can be concluded that the pre-release and version 0.1.0 of the Bitcoin software are the versions where Satoshi had the most influence. A later version could have been included, however, after version 0.1.5, most of the core components had been set. For the purposes of establishing how Satoshi intended the Bitcoin software to work, this report relies on code up to version 0.1.0, as most updates following the version 0.1.0 release are security-related or bug fixes.

Summary of material

The whitepaper, forum posts, emails, source code, and other writings provide the necessary elements to establish a set of criteria for examining the differences between current Bitcoin implementations. There is indication of the proposed system's capabilities, functional requirements, non-functional requirements, and implementation attributes throughout the writings.

The following section will define, as from the source material, what the Bitcoin protocol is and how Bitcoin operates — and finally, summarize what Satoshi's original intentions were for the Bitcoin protocol itself.

[10] Nakamoto, S. (2008). Metzdowd emailing list. https://www.metzdowd.com/pipermail/cryptography/2008-November/014853.html
[11] While there is no way to verify that the code is truly authentic, this is generally accepted to be the case.
[12] Hal Finney was the first person to receive a Bitcoin transaction. He was also the first person to be added to the developers list on Sourceforge to write code other than Satoshi.

3. Defining Satoshi's Bitcoin

Relying on the gathered primary source material, Satoshi's original vision for Bitcoin can be established and defined. The following will outline and summarize the findings from the whitepaper, forum posts, emails, original source code, and other writings.

3.1 The Bitcoin protocol

A protocol is "a set of rules governing the exchange or transmission of data between devices."[13] In the case of cryptocurrencies, a protocol is a set of rules that enables communication between systems and informs the structure of a blockchain.[14] For Bitcoin, Satoshi's original protocol defines the rules which the nodes of the transaction network will follow.

The table on the following page represents the phases of communication through the network and what nodes do during those steps:

Table 1 – Bitcoin protocol phases

	Step 1: Broadcast	Step 2: Block of Transactions	Step 3: Proof of Work ("PoW")	Step 4: Broadcast PoW	Step 5: Accept New Block	Step 6: New Block in Chain
Phase Description	New Transactions are broadcast to all nodes.	Each node collects new transactions into a block	Each node works on finding a difficult PoW for its block	When a node finds its PoW, it broadcasts the block to all nodes	Nodes accept the block only if all transactions in it are valid and not already spent	Nodes Express their accpetance of the block by working on creating the next block in the chain using previous hash.
Electricity Consumption	Low	High	High	Low	Low	Low
Nodes – CPU		●	●		●	●
Network Communication	●			●		
Disk and Memory	●				●	

[13] Definition of "protocol" from https://www.lexico.com/definition/protocol

[14] Each block, and the contents of a block, are written and communicated in an exact manner.

There are several immutable rules common among all Bitcoin implementations:[15,16]

- The sum of the value of the inputs of a transaction must be greater than or equal to the sum of the values of the outputs.
- The block reward subsidy will drop by half at a scheduled rate of every 210,000 blocks, starting with a subsidy of 5 billion satoshis[17] per block from the Genesis block[18].
- The network will adjust the target for the difficulty of the PoW needed for a valid block to maintain an approximate 10-minute block discovery rate.
- Only blocks that add to the blockchain formed by building upon the Genesis block are valid.
- The structure of the Bitcoin database as a distributed timestamp server validating chains of transaction outputs.
- Transaction data formatting, including sizes of certain fields and their encoding schema.
- Block structure and header information including sizes of certain fields and their encoding schema.
- The Bitcoin scripting language and its specification including lists of opcodes that are usable in script and the exact outcome of their execution.
- Bitcoin source code should always be open for anyone to read, modify, copy, and/or share.
- All coins are equal and should be equally spendable.
- Confirmed blocks should be set in stone. Blockchain history should be immutable.

3.2 Capabilities

What does Bitcoin even do? Is it cash? Is it digital gold? To discuss the capabilities of Bitcoin, it is important to first understand the difference between a bitcoin and the Bitcoin network. A bitcoin is a unit of account that allows an individual to transact on the Bitcoin network. It is the capabilities of this network — the rails of a payment system — that make Bitcoin what it is: a network that processes and validates digital transactions.

The capabilities of Bitcoin can be broken down into three areas: transaction validation, identity security, and network access.

3.2.1 Transaction validation

A critical aspect of any merchant's digital payment system is a means to validate that-no amount is double spent. This double-spending problem is particularly important for payment networks that rely on digital transactions. Double spending could potentially cause a conflict whereby it is not known who owns an amount of digital cash. With Bitcoin, this is resolved by a rather elegant solution to the Byzantine General's Problem.[19]

---

[15] *Bitcoinsv Wiki*. (n.d.). Bitcoinsv. Retrieved May 21, 2021, from https://wiki.Bitcoinsv.io/index.php/Protocol

[16] *Bitcoin Wiki.* (n.d.). Bitcoin Community. Retrieved May 21, 2021, from https://en.Bitcoin.it/wiki/Principles_of_Bitcoin

[17] A satoshii is the smallest unit of the Bitcoin token. It is equivalent to 1/100 millionth of a Bitcoin.

[18] The Genesis Block is the first block in the chain of blocks that makes up the blockchain

[19] The Byzantine General's problem is a situation where involved parties must agree on a single strategy in order to avoid failure, but where some of the involved parties are corrupt and are giving false information.

Prior to Satoshi's proposal, there was no appropriate solution to the double-spending problem amongst other proposed digital transaction systems. Anyone who sufficiently understood a digital cash system could theoretically duplicate transactions and potentially defraud merchants or individuals receiving a payment. To avoid this, Satoshi proposed a system that used "a peer-to-peer distributed timestamp server" to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes."[20]

3.2.2 Identity security

The whitepaper describes a unique system which gives users an important option for identity security. The system is a distributed network that was specifically designed so the security and integrity of the solution do not rely on a third-party (requiring trust outside of the distributed network). This trust also often led to requiring the exchange of personal information. By 2009 data breaches were becoming extremely common, substantially increasing the threat of identity fraud. Satoshi deemed a payment system that can provide users an added level of security to prevent identity theft to be essential.[21]

In order to have a system without a trusted third-party, Satoshi's proposal relied on a network of nodes to write new blocks to the chain.[22] For node operators to participate in the transaction network, there must be some sort of economic incentive. With Bitcoin, node operators would have two such mechanisms: First, a number of "fresh" coins are distributed through a gradually decreasing subsidy from the fixed 21 million bitcoin supply for winning the right to create a new block of transactions. Second, transaction fees are paid by senders of coins for all transactions collected into a block, which should continue to increase over time until all 21 million Bitcoin tokens have been distributed into circulation through the block generation process.[23]

3.2.3 Network access

There are early online forum discussions regarding what a Bitcoin token is. Satoshi references the Bitcoin token as something like a store of value (gold) or collectable.[24] However, it is clear through analysis of the source code that a bitcoin is a necessary element of the system for users to access and use the distributed transaction network. It must have some value attached to it which gives node operators an economic incentive to provide computing power to run the network. The long-term value of a bitcoin was left for the markets to decide. However, there needs to be digital cash to run a distributed timestamp server and distributed digital transaction system.

Apart from potential value, issues of scalability were some of the first questions brought up by forum users. In the forums, whitepaper, and emails, Satoshi regularly discussed scalability of the Bitcoin network and provided no reason to believe there should be limits to its potential scalability

[20] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[21] Ibid
[22] A node is a connection point within a network that can send, receive, create, or store data. This could be a single CPU.
[23] A Bitcoin is minted / created when a new block is successfully added to the ledger. Further discussion of this process is in section is below.
[24] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=845.msg11403 - msg11403

and growth. In his response to the questions about scalability dating to 2009[25], it is apparent he believed the protocol and network should be able to handle transactions loads comparable to Visa's credit card processing network.

3.3 Critical components

Through the analysis of Satoshi's writings, several critical components arise which are key to the operations of the Bitcoin network. These include a timestamp server, PoW mechanisms, incentives, policies, independence from third parties, stakeholders, network and blocks, and security.

3.3.1 Timestamp server

The timestamp server is the first step in the process of creating the digital ledger. It provides cryptographic evidence the data existed at a given time. In the case of Bitcoin, this includes any transactions that happened between addresses[26] in a given timeframe.[27] Each timestamp block also contains the hash[28] of the previous timestamp and is chained together.[29] This server takes the hash of a block of items and publishes it for all to see. The process of hashing the previous block and appending it to the new block creates a large chain, which is the basis of Bitcoin. The timestamp server is the cornerstone that builds the Bitcoin transaction ledger.

3.3.2 Proof of work

To have a distributed timestamp server, Satoshi proposed implementing a Hashcash[30] style PoW[31,32] in combination with the timestamp server. CPU effort is used to satisfy the PoW. As new blocks get added, this method makes it increasingly difficult to change previous blocks because work would need to be completed again.

PoW is the system used by Bitcoin to validate all transactions and blocks. The concept begins by scanning for a value that, when hashed with SHA-256, has a beginning with a certain number of zero bits.[33] This process can be made more difficult by increasing the required number of zeros or made less difficult by reducing the number of zeros. This is the basis of work. Using the hash of multiple values and an increasing nonce, a CPU can find a solution with the required amount of beginning zeros after a certain period of effort. Once the required work has been completed, the block is chained to the previous one and this process continues.

For someone to make a change to this block, they would need to redo all the work of the blocks chained after it. Miners attempting this type of attack cannot be anonymous — they would be a

[25] Nakamoto, S. (2008). Metzdowd emailing list. https://www.metzdowd.com/pipermail/cryptography/2008-November/014815.html
[26] In Bitcoin, an address is the only information passed to the network. Addresses can send and receive transactions.
[27] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[28] A hash function is any function that can be used to map data of arbitrary size to fixed-size values.
[29] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[30] Hashcash is a proof-of-work algorithm and was invented by Adam Back in 1997
[31] Proof-of-Work is a form of cryptographic zero-knowledge proof in which one party (the prover) proves to others (the verifiers) that a certain amount of effort has been expended. Verifiers can subsequently confirm this expenditure with minimal effort on their part.
[32] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[33] Ibid

known entity to the network. This confirms the solution to the Byzantine General's Problem, as majority decision-making is not necessary.

PoW is equivalent to one-CPU, one vote. The longest chain has the greatest amount of work invested. In order to rewrite the chain, an entity would need more than 51 percent of the available CPU power to create and maintain a longer chain. Provided majority CPU power is controlled by honest nodes, the chain will grow faster than any other.

3.3.3 Incentives

To incentivize node operators to provide their CPU power to the network, they are rewarded every time they win the right to create a new block and add it to the chain. Specifically, the winning node operator is rewarded with amounts of bitcoin. The total supply of bitcoin is limited to 21 million coins, with "fresh" coins circulated to the winning node operator for every successful block (in the number of the fixed "subsidy" portion of the block reward).

The subsidy portion of the block reward is halved every 210,000 blocks. The original subsidiary amount of the block reward was 50 bitcoin. Approximately every four years, it halves — first to 25, then 12.5, 6.25 (currently), 3.125, and so forth until the full 21 million supply of "fresh" coins is distributed into circulation. As the number of fresh coins from the 21 million supply goes into the network circulation, Satoshi indicated transaction fees (also earned by winning node operators for each block) needed to grow as replacement for the diminishing block rewards, especially once the 21 million supply limits of fresh coins are reached.[34]

3.3.4 Policies

There are policies which differ for each Bitcoin implementation. Most of these changes have to do with block sizes. The original block size was set to 32 megabytes (MB) by Satoshi[35] and later set to 1 MB with intentions to increase it in the future.[36] The major forks of the Bitcoin network represent differences in approach as to whether there should be a developer-controlled default maximum on the block size. A larger block size allows more individual transactions and more overall data to be processed per block.

One of the most debated mutable components of all versions of Bitcoin is the block size. Differences in philosophy about the maximum default block size is a key trigger for hard forks in the history of Bitcoin — although other software differences such as the controversy over BTC's addition of the Segregated Witness[37] function in 2017 have also triggered hard forks — including a division of the Bitcoin network between BTC and Bitcoin Cash ("BCH").

Once an incompatible block (with higher block size or other incompatible rule elements) is mined on a network that has changed its rules, the newer software essentially creates a new branch on

[34] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=994.msg12168 - msg12168

[35] MAX_SIZE is a constant variable defined in "main.h" of BTC v0.1.0 ALPHA. This variable is used to ensure that the vector of transactions is smaller than 32mb. Since all other elements of block header are of negligible size, the overall block size was approximately 32mb.

[36] Nakamoto, S. (2010). Bitcoin Forum https://bitcointalk.org/index.php?topic=1347

[37] Segregated Witness was a soft fork under BIP141 that removed signature data from the BTC blockchain to mitigate a blockchain size limitation problem

the longest chain, then rejects transactions from the older software. The older software can continue by adding blocks compatible with its rule set on the older branch. This creates two different branches of the original chain if a software update is not fully accepted by all users.

3.3.5 Independence from trusted third parties

Bitcoin is meant to be a P2P electronic payment system as described by Satoshi in his original post with the whitepaper.[38] This electronic payment system is built on PoW, rather than trust in a third-party, to validate a transaction has occurred.[39] The Bitcoin blockchain can be compared to a digital ledger. There is a lump sum of Bitcoin, and fresh coins from that supply are distributed based on transactions to different participating parties.

Our stance regarding trusted third parties has been derived from the whitepaper:

"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."[40]

In our analysis, we are including cryptocurrency exchanges as trusted third-party institutions. It's not about avoiding third parties but aligning with the new privacy model proposed in the whitepaper. Using an exchange is a point where private information would need to be disclosed, which does not conform to the privacy model set forth in the whitepaper. An individual would still need to know the third-party whom they are sending / receiving payments.

3.3.6 Stakeholders

The primary stakeholders of Bitcoin are the entities directly related to the P2P electronic cash system. This includes holders / users of Bitcoin, node operators, and developers. Node operators ensure transactions are validated and processed, blocks are broadcast to other nodes, and blocks with valid transactions are accepted and added to the blockchain.

There are secondary stakeholders for the Bitcoin network which are necessary for the system to function. These include electricity providers, hardware providers, and internet service providers. With a shortage of any, the system integrity is potentially at risk.

3.3.7 Network and blocks

Communication within the Bitcoin network is handled by nodes. These nodes are connected to several peers which eventually create a massive network as more nodes join the network. Nodes can freely enter and exit the network. These nodes allow a block's transactions to be broadcast and

---

[38] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://nakamotoinstitute.org/static/docs/bitcoin.pdf

[39] Ibid

[40] Ibid

stored (if the new block is accepted) at the leaves of Merkle trees[41] of other nodes.[42] They are then hashed upwards together, branch by branch, until the root is reached. With the hashed root and the block header of the previous block, a node can begin trying to compete for the next block. Once found, the header of the block is created, and the node broadcasts the solution to other nodes for verification — increasing the length of the chain.

To elaborate, the block header is an 80-byte-long string containing the Bitcoin version number, the previous block hash of the Merkle root, a timestamp of the block, a difficulty target for the block, and a nonce. This structure applies to all blocks except the first in the blockchain (also known as the Genesis Block).[43]

For a node to join the network, it must download the Bitcoin software and run it. The node will first download and validate the entire chain. Once that is done, the node will then be able to communicate with its peers using the Transmission Control Protocol ("TCP"). All communications for Bitcoin are done over TCP.

Satoshi's whitepaper outlines the following base rules for running a node (what many industry observers today call "mining" or "transaction processing") which govern those network participants who create blocks and write transactions to the chain.[44]:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult PoW for its block.
4. When a node finds a PoW, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.
7. Nodes rely on data from the longest chain to enter and exit the network freely.

For the system to remain functional, it is apparent there is no need to have a large number of network nodes — provided there are good actors amongst the node operators.[45] The original Bitcoin implementation and associated discussions on forums indicate the network could run on anywhere from ten computers to a large collection of networked devices; the network difficulty is adjusted accordingly to enable a transaction network of nearly any size or hashpower.

3.3.8 Security

Before a transaction can be validated, it is checked by the other nodes for double spend. In the case a double spend is confirmed, the transaction is deemed invalid. For Bitcoin, the earliest

[41] Please refer to the following link for an explanation of a Merkle Tree if needed: https://www.geeksforgeeks.org/introduction-to-merkle-tree/
[42] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[43] *Bitcoin Wiki.* (n.d.). Bitcoin Community. Retrieved May 21, 2021, from https://en.Bitcoin.it/wiki/Genesis_block
[44] Ibid
[45] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf

transaction is the one that counts. The only way an attacker could validate its own double spend would be to harvest and maintain 51 percent of the network's computing power and build a longer chain. The mathematical probability of this occurrence is explained in the whitepaper. An attack of this type could be subject to prosecution as all information surrounding the attack would be public knowledge.

All public keys used in a transaction visible to anyone. However, it is up to each user to avoid revealing information that can link oneself to a specific transaction and/or key. Users can create a new key pair for each transaction to prevent being linked to a specific public address. However, users do not have to exchange private information (such as name, age, location) to have a transaction validated on the Bitcoin network.

SHA-256[46] is a cryptographic hash function used heavily in Bitcoin. It is computationally infeasible to decrypt a hashed message with computing power currently available. There are no current threats to SHA-256.[47] Even with quantum computing, SHA-256 can be increased to SHA-512 or higher. The only known way to solve the hash is by brute force collisions. Therefore, although there will be more computing power in the future, SHA algorithms can be scaled accordingly.

In the original Bitcoin protocol, the difficulty of solving the hash is changed every 2,016 blocks. This adjustment is used to keep the block creation time at 10 minutes, even with advancing technologies. This leads to an increase or decrease of difficulty approximately every two weeks in the original Bitcoin protocol.[48] The discovery of a solution to SHA hashing functions could be detrimental as Bitcoin could be mined faster based on the computational efficiency of the solution.

3.4 Non-functional requirements

3.4.1 Integrity, transparency, and auditability

There are several non-functional requirements that become apparent through the analysis of the whitepaper and forum posts. For the system to work efficiently and effectively it requires:

- Integrity amongst (some) of the node operators.
- Confidentiality and privacy of network participants upheld.
- Transparency and auditability within the ledger itself.
- Network reliability to process transactions.
- Network scaling to handle a large number of transactions.

[46] Secure Hash Algorithms are a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS)

[47] *SHA-256 Cryptographic Hash Algorithm.* (n.d.). Moveable Type Scripts. Retrieved on May 20, 2021, from https://www.movable-type.co.uk/scripts/sha256.html

[48] The difficulty adjustment algorithm period for BSV is a moving window of the last 144 blocks, a change inherited from the Bitcoin Cash (BCH) implementation which resulted from a split of the BTC network in August 2017.

3.4.2 Availability

Provided there are other nodes to connect to, the ledger will remain readily available. Should there be no nodes or only a single node, the ledger will become unavailable. If there is a significant drop-in hash rate of the connected nodes there may be an instance where new transactions are at a standstill, depending on the current hashing difficulty. This situation is highly unlikely due to the profitability of mining Bitcoin.

The time required to confirm a Bitcoin transaction cannot be precisely predicted. Confirmation time usually depends on the fee set by the transacting user, successful block generation time, and propagation of the transactions through the network. If the user sets a fee that is too low, it risks not getting pick up for several blocks, which delays the time for a transaction to be confirmed. Block generation time varies depending on the difficulty of the network but is, on average, 10 minutes per block.

3.4.3 Scalability

Scalability seems to be one of the biggest key performance indicators for measuring blockchain performance. There are several known Satoshi posts that directly or indirectly raise the issue of being able to handle large amounts of transactions via various business channels.[49] Through forum posts and responses, Satoshi regularly asserted the Bitcoin network would be able to scale appropriately to sufficiently handle many types of unique use cases requiring the network and its underlying infrastructure.

3.5 Implementation Attributes

The main attributes for the original Bitcoin protocol and network include block size, economic incentives, and consensus mechanisms. These three attributes provide defining characteristics of any given Bitcoin implementation, how that network will perform, and the functionality the network provides. Ultimately that functionality determines how various applications utilize the power of a particular blockchain network.

3.5.1 Block size

Block size is one of the main elements that allow for scalability of transaction volume and date capacity on the network. A smaller block size means a limited number of transactions can happen. If Bitcoin should be able to compete with the likes of Visa, as suggested by Satoshi, then it clearly requires block size be adjusted higher over time.[50] To supplement this, Satoshi mentions in an email on the Metzdowd list that the potential size of a block could be nearly 100 gigabytes in the future. This implies Satoshi had intentions for the block size to grow to that point.

Moreover, there are several use cases discussed on the Metzdowd emailing list, Bitcoin Forum, and P2P Foundation forums. These include things like escrow services, pay-to-receive email, vending

[49] Nakamoto, S. (2008). Metzdowd emailing list. https://www.metzdowd.com/pipermail/cryptography/2008-November/014815.html
[50] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=1347.msg15366 - msg15366

machines, web hosting, and software as a service ("SaaS") payments. In effect, if there is a reason for people to be transacting, the Bitcoin network should allow users to conduct that transaction. Larger block sizes would be important to enable such additional use cases.

Block size is also important from the perspective of economic incentives that draw node operators into running the distributed network. With a small block size, transaction fees will dramatically increase over time as network use increases due to the limited number of transactions that can be processed per block. Satoshi mentions several times that transactions on the Bitcoin network will operate on low to no fees.[51] Additionally, only after all the Bitcoin tokens have been minted and distributed through block generation will the network rely on fees alone. To allow for this, and to keep fees low, block size must be larger to allow for more transactions.

In addition to this block reward "subsidy" amount for blocks they win, node operators also receive the aggregate of fees paid by all senders of transactions collected into the block they create. Node operators must earn more in transaction fees as the number of fresh coins awarded in each block reward decreases, both to keep fees low and to offset the reducing number of coins awarded in the fixed "subsidy" for each block. This implies block size must be larger to allow for more transactions and thus more transaction fees per block.

3.5.2 Economic incentives

Economic incentives stem from two elements of the protocol itself. First is the fact there will only ever be 21 million bitcoin tokens — which were minted at the moment the Bitcoin system was created — and "fresh" coins will be distributed over time with each new block until the supply of coins is fully circulated. This leverages the well-known economic principle of scarcity. With a finite amount of resources, the value of a resource in the future should be higher than the value of that same resource today. In theory, scarcity should provide enough incentive for the first generation of node operators to generate bitcoin.

To address the fact that the supply of block rewards will eventually end, Satoshi posited there should be enough transactions (and thus transaction fees) within a single block to incentivize mining in the long run.[52] This implies a few things: First, Satoshi's sentiment that transactions should be low cost requires a large number of transactions be included in a single block. Second, it would not be necessary to have a large number of node operators — only a few who are specialized in mining at scale.

An important addition to the economic incentive of node operators relates to the cost of mining.[53] Initially, mining was done on small CPUs running on personal computers. In a forum post, Satoshi mentions trying to slow down the GPU[54] race,[55] as more people were attempting to win hashing

[51] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=994.msg12168 - msg12168
[52] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[53] Please refer to the following article for a brief explanation of Bitcoin Mining if needed: https://www.investopedia.com/tech/how-does-bitcoin-mining-work/
[54] GPU (Graphical Processing Unit) refers to a computer's video card
[55] Nakamoto, S. (2009). Bitcoint Forum. https://Bitcointalk.org/index.php?topic=12.msg54 - msg54

races and block rewards using more powerful computers. Now, most large node operators require specialized hardware, such as ASIC processors that are purpose-built to mine Bitcoin.

Since Bitcoin requires PoW, the rewards from mining must remain profitable for the node operators. In other words, hardware and electricity costs for miners directly affect profitably. As more mining groups and mining pools compete for the block rewards more hash power is required to win the right to mine the next block. Therefore, the average cost of equipment and electricity for a miner to generate a block reward must be at least equal to or greater than the block reward (fixed subsidiary amount + transaction fees).

3.5.3 Consensus mechanisms

A key feature that distinguishes Bitcoin from other blockchains is its consensus mechanisms. Currently, there are blockchain projects that rely on alternate consensus systems such as proof of stake (validation in proportion to the amount of currency owned and staked for the right to validate transactions on the network) and proof of space (allocation of storage to solve a challenge). Different consensus mechanisms affect which operators will join the distributed network.

PoW has proven to be a secure system because of the intrinsic difficulty for someone to attack the blockchain. As the whitepaper explains:

> ***"To modify a past block, an attacker would have to record the proof of work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes."***[56]

An attack on a PoW blockchain is not economically incentivized while the majority decision (consensus) is represented by the longest chain, which has the greatest PoW invested in it. Additionally, any attack would be publicly visible which implies that actions can be taken to permanently stop attacks.

With PoW, the energy requirements can become larger if the network develops sufficient value to attract more computing power willing to compete for the fixed subsidy block rewards and transaction fees on the network. Additionally, node operator consensus allows for the forking of the blockchain should the mutable rules change.

3.5.4 OpCodes and scripting

Bitcoin uses a scripting system for transactions. A script is a set of instructions sent along with each transaction that describe how the next person can gain access to the bitcoin being sent.[57] Scripting provides a framework to change the parameters of what's required to access transferred bitcoins.

[56] [56] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[57] *Bitcoin Wiki.* (n.d.). Bitcoin Community. Retrieved May 21, 2021, from https://en.Bitcoin.it

For example, the scripting system could be used to require multiple private keys, combination of several keys, or even that another transaction be completed first.[58]

OpCodes[59] are the underlying operations that are used to build Bitcoin scripts. They are the building blocks that allow a developer to create tools which allow users to have wallets, send transactions, and essentially manage their accounts. In addition, OpCodes allow more technical users the freedom to create systems that sit on top of the Bitcoin network.

OpCodes, and the scripts created, become a fundamental element in determining how different Bitcoin implementations will function. For example, OP_PUSHDATA4 allowed users to push over four gigabytes of data onto the stack through each Bitcoin transaction. This OpCode indicates Satoshi clearly intended the Bitcoin network to handle large volumes of data — even pushing up to four gigabytes of data per transaction (not even per block). OP_PUSHDATA4 could be called multiple times, indicating a transaction could hypothetically be larger than four gigabytes.

3.6 Analysis of Bitcoin software v0.1.0 ALPHA

The Bitcoin software v0.1.0 ALPHA[60] was developed on Windows and shared as an open-source project on SourceForge. After installing the software, a user was required to download the entire chain for validation. Once that was done, a user was able to connect to other nodes on the network.

For transactions, a user was able to see all crediting and debiting transactions from the wallet along with its description, status, and date. A user was able to send coins via IP for an online transfer or via Bitcoin address if the recipient was not online. The sending via IP address was removed in later versions due to many security risks. It was at the sender's discretion whether to include a transaction fee.

In order to mine (operate as a node), a user had to open the options tab and select the "Generate Coins" option. The software would start to mine in the background of the computer using the CPU.

It was noted in the description above that the software was purely experimental and should not be relied on for "actual financial transactions."

There were several updates made to the software after v0.1.0. However, there were no substantial changes to the Bitcoin software until Bitcoin v0.1.5. Here, a minimum transaction fee was set for transactions of less than one cent to prevent DoS attacks.[61] This was later removed as users found it

[58] Ibid.
[59] A full list of OpCodes for both BTC and BSV is available in Annex 2
[60] See Annexes for timeline of source code changes up to v0.1.5
[61] A Denial-of-service attack (DoS attack) is an attack with the objective to render a network unusable. In the case of Bitcoin, mass spamming of free transactions in the early days of the network could make it unavailable for other users.

confusing. Satoshi mentioned in a reply on the Bitcoin Forum: "[w]e should always allow at least some free transactions."[62]

The Bitcoin v0.1.0 software was rather limiting. There was no command line implementation. If changes were to be made, they needed to be hardcoded in the project files. There was no formal bug tracking software; all bugs had to be reported to Satoshi via email or Bitcoin Forum. However, the software allowed key functionalities as described in the whitepaper such as the sending / receiving of Bitcoin, timestamped blocks, PoW with mining Bitcoin that utilized CPU power, and a network with connected peers that required no personal data.

3.7 Satoshi's original vision

From examining the whitepaper, forum posts, emails, and the source code for original Bitcoin software, an image of Satoshi's system begins to take shape. Satoshi describes, for the time, a novel idea to a problem that had been around since the 1990s: How is the implementation of a digital cash (without any trusted intermediaries) possible?[63] Early digital cash projects had flaws when it came to bad actors being able to double spend.

Satoshi's vision for the Bitcoin project is best summarized by his first post on the Metzdowd Forum[64]:

"I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third-party.

The main properties:
Double-spending is prevented with a peer-to-peer network.
No mint or other trusted parties.
Participants can be anonymous.
New coins are made from Hashcash style proof-of-work.
The proof-of-work for new coin generation also powers the network to prevent double-spending."

Several core elements are required for the functionality of the Bitcoin system, as proposed in Satoshi's whitepaper. These include a distributed timestamp server, a PoW mechanism, and network rules, which lay the foundation the Bitcoin software uses to turn the whitepaper into a proof of

[62] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=994.msg12168 - msg12168
[63] Ibid
[64] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf

concept. Additionally, the Bitcoin software[65] provides the ability to store coins, send and receive transactions, and verify payments.

Satoshi's vision for Bitcoin outlines a distributed transaction network along with protocols for the network to operate in a manner which alleviates the double spending problem. In addition, there are clear indications from the original source materials this system should be able to scale to the level of other large payment processors such as Visa.

This payment system had user data privacy at its forefront. From the whitepaper, forum posts, and other writings, it is clear there is no intention to rely on a trusted third-party for transactions — thus, eliminating the need for the traditional payment processors to collect potentially sensitive data on its clients. It is important to note a trusted third-party refers to a traditional financial institution such as a bank or payment processor. The system does not record an individual's identity, but the individuals will know who they are transacting with. For example, when Alice sends bitcoin to Bob, Alice and Bob will know each other's identity.

4. Comparing current implementations to Satoshi's vision

The Bitcoin implementations that will be compared and referenced in this paper are those which directly evolved from the Bitcoin implementation and subsequent forks of the Bitcoin software code. Specifically, the choice was made to focus on the BTC and BSV implementations for the following reasons:

- These contrast the key attributes in realizing Satoshi's vision.
- BTC is currently the most popular and the most valuable coin.
- Capacity to identify impact of various development decisions of each implementation.

It is important to note there are several other Bitcoin implementations because of forks in the original Bitcoin ledger. However, BTC and BSV provide us with two distinct implementations. BTC is considered the original chain that was worked on by Satoshi until development was taken over by the community.[66] BSV was established with the clear intent to implement Bitcoin according to Satoshi's vision by restoring the original code and adhering to the design principles expressed in the whitepaper, forum posts, and emails.[67] While other implementations have sought to improve the operations of the network, it was noted BSV has taken an approach to increase scalability through larger block size limits and functionality through a restoration and extension of Script functionality within its OpCodes.

Future considerations could include other implementations such as BCH, however, it is out currently of the scope of this report. The level of effort required to fully review the BCH

[65] Bitcoin was original distributed as a GUI based software which allowed users to create a node, and send/receive transactions
[66] Community refers to the group of members of the Bitcoin Forum who participated in discussions, bug fixes etc.
[67] *Bitcoinsv for Developers.* (n.d.). Bitcoinsv. Retrieved May 21, 2021, from https://Bitcoinsv.io/satoshis-vision/

implementation would add considerable time to the completion of this report. For the sake of a comparison, and the fact that BSV is a fork of BCH, BSV provides an adequate counterpoint to the development decisions from the BTC community.

4.1 Description of assessment framework

The assessment framework used within this section was developed for the collection and sorting of research information. This was established to clarify the specification and intents for BTC and BSV and compare these to the specification and identify limitations or opportunities for certain use cases.

In the framework, lines of inquiry were identified in several areas including approach and concept of Bitcoin, components, legality, privacy, design, external factors, resource usage, functionalities, functional and non-functional requirements, associated risks and use cases. Each area was subdivided into assessment procedures that were completed for BTC and BSV as well as the first release of Bitcoin ALPHA v0.1.0. Each subcategory utilized resources available online including, but not limited to, source code, wiki pages, original forum posts by Satoshi Nakamoto, original emails written and received by Satoshi, and direct testing procedures of the different implementations.

4.2 Comparison of implementations

Table 2 below illustrates a summary of the key assessment areas against BTC and BSV current state protocols:

Table 2 – Summary Results of Assessment Procedures

Area of Review	Testing Criteria	BTC 0.21 Review (Met / Partially met / Not met)	BSV 1.07 Review (Met / Partially met / Not met)
Capabilities	Double spend prevention	Met	Met
	Independent of trusted third parties	Partially met	Partially met
	Incentive mechanism	Partially met	Met
	Scalable transactions	Not met	Met
Functional requirements	Timestamp server	Met	Met
	PoW mechanism	Met	Met
	Network rules	Met	Met

Area of Review	Testing Criteria	BTC 0.21 Review (Met / Partially met / Not met)	BSV 1.07 Review (Met / Partially met / Not met)
Non-functional Requirements	Security	Met	Met
	Reliability	Partially met	Met
	Scalability	Not met	Partially met
	Maintainability	Partially met	Partially met
Implementation attributes	Block size	Not met	Met
	PoW mechanism	Met	Met
	Energy requirements	Partially met	Met
	Difficulty	Met	Met
	OpCodes & Scripting	Partially met	Met

In some functions, BTC and BSV are similar. For example, both can allow users to transact without relying on a third-party and prevent double spending leveraging similar timestamping, consensus mechanisms, and encryption methods.

However, the original vision was meant to be an improved way of sending / receiving transactions online with a lower probability of losing personal data while solving the major problem of double spending. It sought to create a more efficient means of internet payments, including "small casual transactions" by eliminating the needs for intermediaries.[68] The original design intended Bitcoin to be used for other functions (beyond mere payments) such as vending machines, paid emails, SaaS products, website activations, etc.[69] It is clear from the forum posts, writings, and community discussions the Bitcoin protocol was intended to scale to allow for many forms of payments. This includes macro-payments (i.e., settling an account at the end of the day) and micropayments (i.e., sending a small fee to access a service or send a text message).

Satoshi's original vision of an electronic cash and payment system, as previously defined, implies the biggest limitation of BTC is the small block size and growing fees to send transactions. In the long-run, transaction fees are the economic incentive for node operators to process and add a new block to the chain. Satoshi discussed a system that would always pick up the free transactions and at least intended Bitcoin to run on a low fee model.[70] If node operators on BTC are to have the

[68] Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf
[69] Nakamoto, S. (2009). Metzdowd email list. https://www.metzdowd.com/pipermail/cryptography/2009-January/015014.html
[70] Nakamoto, S. (2010). Bitcoin Forum. https://Bitcointalk.org/index.php?topic=1347.msg15366 - msg15366

proper economic incentives in the long run, either fees for individual transactions will need to dramatically increase — reducing (or outright preventing) the ability for micropayments — or there must be a change to allow for increased transaction counts in a single block for there to be enough smaller individual fees to justify processing the next block. In the long run, when mining is completed, fees are the only economic incentive for node operators.

Considering these factors, BSV appears more aligned to the original intent of the payment system. With an uncapped block size which can grow with market demand, a significantly largely number of transactions can be included within each block, thereby allowing higher overall network throughput. Miners could wait until a suitable number of transactions and their combined fees (i.e., the economic incentive) arrived until they started processing the next block. This would allow for various payment sizes and the potential for some users to even process transactions with zero fees.

Another key difference between BTC and BSV lies within the details of the embedded development script language implementations, or OpCodes, which allows for functionality such as smart contracts to be developed within each system. BTC has fewer OpCodes implemented than BSV (because many of those OpCodes were disabled or restricted along BTC's history after Satoshi left the project), and the functionality of some of the OpCodes are different than originally designed.[71] In effect, this limits a developer's ability to build complex functionality on top of the BTC protocol. Some examples of OpCodes that are disabled in BTC and enabled in BSV are "OP_CAT," "OP_MU," "OP_DIV," though there are many more.

There are other OpCodes in BTC that have been modified, such as "OP_RETURN," which completely change the functionality of scripting. OP_RETURN was originally created by Satoshi to end a script and skip the remaining instructions. This functionally no longer exists in BTC but is present in BSV, which allows for more complex scripting as a result. Satoshi mentions that people would develop a system suited for their own needs, but also the need for a system that would allow for as many potential outcomes as possible. Smart contracts were envisioned and possible with Bitcoin script and were built in from the start using the FORTH[72]-like script.

However, as mentioned previously, some OpCodes were disabled and changed over time. Coupled with smaller block size, the potential for complex functionality to be developed was limited. OpCodes such as OP_CHECKLOCKTIMEVERIFY and P2SH moved BTC away from the functionality Satoshi had planned for the system.[73] BSV re-enabled original Satoshi OpCodes; removed restrictions (including data limits) on certain OpCodes; removed any developer-set default limit on block sizes to remove limitations imposed by BTC developers; and made more complex and functional tokens, smart contracts, and other advanced functionality possible. A complete breakdown of all OpCodes is available in Annex 2.

[71] Comparison of OpCodes can be seen in Annex 2

[72] FORTH is a procedural, stack-based programming language and interactive environment designed by Charles H. "Chuck" Moore

[73] See Annex 5 for a more detailed primer on the differences that moved BTC away from, and BSV closer to Satoshi's original vision.

Table 2.1, below, represents a more detailed version of the differences in key assessments between BTC and BSV.

Table 2.1 – Detailed results of assessment procedures

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Double spend prevention	Review whether the sampled blockchain follows the design principle of eliminating double spending as described in the Bitcoin whitepaper.	The protocol leverages consensus mechanisms using a timestamp server and longest chains to eliminate the likelihood of double spending.	Partially met The protocol leverages consensus mechanisms using a timestamp server and longest chains to eliminate the likelihood of double spending. There is one instance of a possible double spend on January 21, 2021.	Met The protocol leverages consensus mechanisms using a timestamp server and longest chains to eliminate the likelihood of double spending. There are currently no known instances of double spending.
Independent of trusted third parties	Verify users can independently purchase goods and services without resorting to paper or coin currency. Consider merchant applications and support infrastructure.	An entity can transact without the reliance of any third-party.	Partially met If the entity does not mine the token, or receive them as payment, they would have to rely on a third-party exchange to purchase the specified token. Mining is no longer reasonable for most individuals as it is not possible without specialized equipment. An individual may have to rely on a third-party (such as an exchange) to first receive tokens.	Partially met Same as BTC.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Incentive mechanism	Verify intermediate nodes (i.e.., miners) receive a reward for successfully delivering transactions from the sender to the receiver. Identify the incentive design mechanisms (e.g., miner reward and transaction fee-setting mechanisms).	Node operators who successfully add a new block should receive any new tokens from the minting / reward process along with any transaction fees. Once all-new coins have been generated, there should be an appropriately sized transaction fee to allow for continued node operations while still allowing for transactions fees to be low. This implies a very large number of transactions per successful block. While there is remaining bitcoin to be mined, "[w]e should always allow at least some free transactions."	Partially met As of the current date, miners are rewarded with 6.25 BTC per valid block mined. Miners are also able to collect transaction fees associated with the block. Satoshi envisioned the subsidy portion of the mining reward as the incentive for miners until there is no remaining fresh bitcoin to be distributed with new block rewards. That is when the transaction fees would be mandatory in order to maintain an incentive to mine. In the current state, any transactions without fees can starve in the memory pool.	Met In the current state, based on several blocks, there are free transactions that are processed.
Scalable transactions	Can the sampled blockchain scale accordingly to support significant volumes of transactions in a timely manner if there are significantly more global adoption and usage?	The network and block size should scale dependent on network utilization. The network should be able to compete with any existing large payment network in terms of transaction volumes per day.	Not met BTC is limited due to the fixed block size of 1 MB and can only handle a maximum of approximately seven transactions per second. This imposes a maximum number of transactions that can be processed daily and would not facilitate additional transaction volumes by additional adoption.	Met BSV no longer has any block size limit set as default by protocol developers. BSV can maintain scalable transactions by increasing block size according to network market forces.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Timestamp server	Review the features in place for timestamping blocks. Do these functions match the protocol as detailed in the Bitcoin whitepaper and forum posts (i.e., recording transactions in the wallet.dat)?	The protocol should require all transactions and blocks to be timestamped. This is the basis for providing proof that transactions happened in the past. This record should act as a source of truth as well as provide the evidence of ownership.	Met It was noted the protocol leverages timestamp server mechanisms to serve as proof of existence.	Met Same as BTC.
PoW mechanism	Review whether the sampled blockchain follows the design principle of hash-based PoW as described in the Bitcoin whitepaper.	A consensus mechanism such as PoW must be in place to reduce the chances of tokens being double spent.	Met It was noted the protocol uses the PoW design principal as described in the Bitcoin whitepaper.	Met Same as BTC.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Network rules	Identify the various peer-to-peer networking elements (e.g., node discovery, information propagation and verification). In addition to the core decentralized blockchain functionalities, identify the mechanisms in place for ad hoc message passing and distributed networking.	The network should have a robust protocol in place. This includes how nodes communicate and propagate information across the network. Nodes entering the network should always rely on the longest chain as the source of truth. The following rules should always be applied: 1. New transactions are broadcast to all nodes. 2. Each node collects new transactions into a block. 3. Each node works on finding a difficult PoW for its block. 4. When a node finds a PoW, it broadcasts the block to all nodes. 5. Nodes accept the block only if all transactions in it are valid and not already spent. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.	Met BTC uses P2P networking elements described in the whitepaper.	Met Same as BTC.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Security	Determine which type of secure hash function is currently implemented (e.g., SHA-256, SHA 512, etc.). Furthermore, are there any IT roadmaps or planned upgrades to replace the existing SHA? Identify any potential weaknesses / threats to secure hash function (e.g., quantum computing or other cyber security threats)?	A secure hash function should be in place. There should be opportunities to change the hashing function should there be a potential security flaw in the implemented hashing function to allow for the continued operations of the network.	Met It was noted the protocol used similar hashing algorithms such as SHA256. Biggest security risks are associated with third-party services and not the protocol itself.	Met Same as BTC.
Reliability	Assessed through interviews.	The network should always be available and process all valid transactions including free ones as there is still bitcoin to be mined.	Partially met Free transactions can be ignored by miners and may never get processed. The protocol does not have any history of being unavailable.	Met Free transactions can and do occur within a BSV block. There is no known downtime of the network.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Scalability	Do the current value-added activities align to the benefits as intended by Satoshi (e.g., efficiency, eliminating a third-party, decentralization, reduced errors, increased efficiency, scalability, etc.)?	The network should be able to scale meet any demand in transactions and how users adopt the network.	Not met The network is unable to scale with adoption of the network due to limited block size. There is a hard cap of seven transactions processed per second. This restricts the usage of the network.	Met Theoretical unbounded number of transactions per block. This is restricted by hardware and software limitations but can theoretically grow with improvements in technology over time. This would allow for mass adoption of payment processing. This allows the network to be used in different use cases such as a micropayment system.
Maintainability	How well is the system maintained? Who are the parties responsible for implementing changes and upgrades over time? Will it be reliable in the long run?	Depending on the scale, there should be a group or consortium leading the ongoing development of the protocol. Given this could grow to a global scale, there should be flexibility and clear governance procedures to allow for implementing critical path changes (ex: addressing major security flaws). The project shall always be open source.	Partially met There is a large community of developers contributing to the efforts of maintaining the ongoing development and support of the platform. There are instances of groups of these developers not agreeing on a clear path forward for development decisions (BTC hard forks).	Partially met There is an organized group of developers engaged by a global non-profit association-contributing to the efforts of maintaining the ongoing development and support of the node software and other infrastructure tools for the platform. There are some additional developers and development groups around the world also contributing infrastructure tools to the network. There is a dependency on a third-party group of developers.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Block size	Do the current value-added activities align to the benefits as intended by Satoshi (e.g., efficiency, eliminating a third-party, decentralization, reduced errors, increased efficiency, scalability, etc.)?	Block size should scale based on network demand. There should be systems in place to prevent denial of service attacks by bad actors leveraging large block sizes.	Not met Block size has not scaled with the times as it is still at 1 MB. This was set by Satoshi as a starting point. He envisioned the protocol to scale with the times.	Met The theoretical unbounded block size aligns with Satoshi's scaling vision.
Energy requirements	How much energy will be required when running the distributed system?	Energy consumption should be left to node operators to determine. The overall energy requirements will vary depending on the scale of the network.	Not met The overall energy requirements will change as the system scales and depends on how many unique node operators exist. See Annex 1 for a current estimation of power consumption. The required amount of energy per transaction will go up with difficulty as the block size is capped. Specialized equipment is required to mine block and power consumption has increased with network size.	Met The overall energy requirements will change as the system scales and depend on how many unique node operators exist The required amount of energy per transaction should be significantly lower as the block size increases and more transactions can be processed by the network with each block.

Criteria	Assessment Procedure Description	Satoshi's Original Idea	BTC V 0.21 Review (Met / Partially met / Not met)	BSV 1.0.7 Review (Met / Partially met / Not met)
Difficulty	Verify the projected difficulty adjustment multiplier follows the equation (i.e., blocks since last adjustment divided by time since last adjustment).	Difficulty should adjust based on the hashing power of the participating node operators. It should be adjusted so new blocks of transactions happen approximately every 10 minutes.	Met The protocol uses the same difficulty formula as described by Satoshi in the Bitcoin whitepaper.	Met Same as BTC.
OpCodes & scripting	Verify there is a robust scripting language to enable developers to create any style of transaction.	The scripting language and OpCodes allow developers to create contracts.	Partially met Many OpCodes have been disabled which prevent the creation of detailed and complex smart contracts.	Met BSV has re-enabled OpCodes which allows for complex scripting and smart contracts.

5. Impact of Satoshi's visions once fully realized

The original vision and long-term intentions were far beyond a mere electronic cash and payment system. These elements are primarily provided by the script language and implementation details of the stable, underlying protocol.

5.1 Electronic cash / payment system vision

In the years since its inception, Bitcoin has grown to include millions of currency holders,[74] with hundreds of thousands of active daily network users and approximately 300,000 transactions processed per day. However, these numbers are extremely small compared to the billions of overall daily transactions on other payment systems — for example, daily global credit card transactions were estimated at over 1.01 billion in 2018.[75]

This incredibly large number of transactions processed globally also implies an incredible amount of facilitation and control by financial institutions and intermediaries, as well as government and industry regulatory overhead.

Bitcoin set out to make a global payment system by enabling individuals to transact directly with each other via a secure, P2P and distributed technology solution, and allowing them to transmit value over the network via data transfers. There is still a relatively small global number of people who use Bitcoin as an electronic cash (currency) and payment system; that is in part because BTC has not become the efficient system for electronic cash payments Satoshi envisioned. Bitcoin payments are not widely accepted by merchants and other organizations. Moreover, the overall combined value of the currency and transactions pales in comparison to the established fiat systems. Several issues are standing in the way of Bitcoin becoming the predominant global currency or payment system, including:

1. Volatility – Bitcoin's value (in the form of BTC) is highly volatile and unpredictable. This makes the value exchanged subject to change, sometimes on an hourly or daily basis. Balancing the payment received for the value of the goods and services exchanged back to financial accounting systems can be problematic.
2. Ease of use – Purchasing and using Bitcoin remains poorly understood within the general population because it is more complex than traditional banking and payment systems.
3. General acceptability – In most areas of the world (with the exceptions noted), Bitcoin (whether BSV or BTC) is not readily accepted as payment. Many merchants advertise it is a promotion, but often are not adequately prepared with Bitcoin point of sale ("POS") or accounting systems to balance the currency.

[74] https://www.buyBitcoinworldwide.com/how-many-Bitcoin-users/
[75] https://www.cardrates.com/advice/number-of-credit-card-transactions-per-day-year/#worldwide

4. Scalability – The BTC network handles only approximately seven transactions per second and transactions are not finalized for up to one hour in some instances, which makes high-volume retail POS processing problematic. This pales in comparison other payment systems such as Visa, which can process millions of transactions per second at POS (although arguably final settlement occurs much slower).
5. Reputation / perception – Because of its linkages to criminal transactions, money laundering, and many well-published exchange hacks, Bitcoin has a poor public reputation.

BSV seeks to address these and other issues to more fully realize the vision of Bitcoin as an important electronic cash and payment system.

In developing nations where banking systems and currency are non-existent or unstable, there are many examples of Bitcoin being used, including:

- Person-to-person payments via mobile Bitcoin wallets between people who have no access to banking.
- Large payments from corporations to operations in emerging nations facilitated by Bitcoin where traditional banking methods are slow and expensive.

In its complete form, the Bitcoin vision could have major implications for global financial institutions, consumer behaviour, and investment strategies. Financial institutions which do not recognize and embrace cryptocurrency and include Bitcoin in their strategic planning could face elimination. Many banks, governments, investment houses, and other constituents in the financial system have either already recognized this threat and put cryptocurrency plans and offerings (including Bitcoin) in place or have contemplated adapting to emerging consumer behaviour in the near future to ensure their long-term viability. This is often done within digital transformation initiatives or innovation concept labs.

5.2 Other use case elements in the vision

In addition to the electronic cash vision, the Bitcoin system also included (and realized within the code of its software) from the earliest days necessary elements to enable distributed data applications. Bitcoin (along with its variants and most other digital currencies) use what is now called blockchain technology — what the whitepaper describes as a distributed timestamp server — which, when combined with the original Bitcoin protocol rules, provides the backbone for enabling the vision.

At a high-level, blockchain architectures provide the following benefits / services that can be leveraged within enterprise business applications:

- Cryptographic security
- Immutability
- Provenance
- Distributed data management
- Pseudonymity

- Controlled transparency
- Auditability

The ability to leverage permissioned access to secured information stored in a publicly shared blockchain ledger and the ability to leverage smart contracts and other data functions between participants are the strongest functions that blockchain applications bring to the table for developers of enterprise applications.

Depending on the business requirements and services required, developers have traditionally been left with selecting a blockchain model from one of three choices:

- Public – Like the Bitcoin or Ethereum network for cryptocurrency applications
- Private / permissioned – Internally facing applications on infrastructure such as Hyperledger's Fabric
- Consortium blockchains – Developed and operated by a group of organizations, such as a supply chain like IBM's Food Trust consortium solution.

The overhead (upfront costs, governance challenges, etc.) of operating a private or consortium blockchain puts these solutions beyond the reach of smaller corporations or start-ups and can push project implementation times into many months or years.

From the earliest versions of Bitcoin, the script development language has enabled many of the functions that support enterprise applications. There has been debate around whether Bitcoin Script has sufficient functionality (or is Turing[76] complete) to be considered general purpose for creating these types of applications, and whether block size limitations (1 MB) may prevent the broader use of Bitcoin to enable more functional applications.

The BSV platform has addressed these specific limitations and, in so doing, has provided the capabilities required for enterprise blockchain application functionality, including:

1. Higher transaction throughput enabled by on-chain scaling due to removal of default block size limit.
2. Lower transaction fees enabled by larger block sizes which eliminates congestion.
3. Non-transactional data that can be stored and manipulated on-chain due to the larger block size and script language functionality.
4. BSV transactions are processed very quickly as there is a greater than 99 percent chance these will be verified and included in the following block in the blockchain.
5. BSV Script language is Turing complete, which means the script can theoretically execute any algorithm with limitless possibilities.
6. BSV offers the ability (through the scripting language enhancements) to create more complex smart contracts supporting most conceivable asset types.
7. BSV offers very large data storage capabilities.

[76] Turing complete means the scripting language could, in theory, be used to solve any computational problem.

As a result, there are numerous developers using Bitcoin SV as an enterprise blockchain platform for current applications involving transactional and traceability capabilities. A non-exhaustive list of potential application use cases is provided below:

1. Data integrity and timestamping of documents / records
2. Supply chain applications, including end-to-end traceability applications and transactional applications
3. Data storage and informational database applications
4. Many tokens and token-based applications
5. Micropayments for social media, online content, digital advertising, and online games
6. POS applications

The BSV blockchain has also allowed for a layered approach to enable off-chain components and data to seamlessly integrate with the platform. This will enable easy-to-use interfaces between what businesses and developers want to build and the more complicated Bitcoin network mechanics that lie "under the hood." Programs can leverage the money features, data structures, and even each other with BSV. As the protocol enables a global ledger for anyone to work on the potential for interoperability between applications is unprecedented. Programs no longer need to operate solely on servers and their own databases, they can interact with other applications using the same ledger.

Fully enabling this part of the vision provides a platform(s) that can satisfy enterprises needs with massive scaling, a stable protocol, and a regulation-friendly ecosystem — powering blockchain applications today and into the future.

6. Summary and conclusion

In late 2009 Satoshi proposed a new system to handle the transactions of a digital cash. This concept has since grown to become a new type of standard for digital transactions. Like the early versions of the internet, there have been many improvements and unique systems developed on top of the Bitcoin protocol. There have even been competing protocols developed.

From proof of concept in early 2010 to present, Bitcoin has gone through its stages of evolution as well. Currently, there are several competing Bitcoin implementations. This paper examined two of those implementations: BTC and BSV. BTC and BSV both forge their own path in implementing the Bitcoin transaction network as they see fit.

To summarize, Bitcoin, according to the writings and source material left behind by Satoshi, is type of digital transaction network. At its core, the Bitcoin network provides mechanisms to allow for a distributed transaction network to handle digital payments. Bitcoin has inherent features to greatly reduce the chance of double spending, decrease the reliability on "trusted" third parties and subsequently protect the privacy of those transacting on the network, and allow for the network to scale to any size required by the participants of the network.

After examining BTC and BSV compared to the original vision set forth in the whitepaper, forum posts, emails, and other writings by Satoshi, it is our opinion BSV is the implementation that currently best represents what Satoshi originally intended. BSV has a theoretically unbound block size, which allows payments to scale to the size of a Visa-like network without requiring an increase in fees to meet the economic requirements of the node operators. BSV also provides more functionality in terms of how developers can utilize the network for building their own transaction systems on top of the Bitcoin protocol.

Regardless of which Bitcoin implementation the reader prefers - Bitcoin, the blockchain, and distributed ledger technologies will continue to have measurable impacts on the future of commerce, exchange, and trust in an increasingly decentralized and digitally dominant world. Satoshi's vision laid the foundation, but even he and his early acolytes would be amazed by the potentially limitless ways this new technology can benefit society and enhance the financial ecosystem.

Bibliography

Bitcoin Core integration/staging tree. Bitcoin Core [Source code]. Retrieved March 31, 2021, from https://github.com/Bitcoin/Bitcoin

Bitcoin Difficulty historical chart. (n.d.). BitInfoCharts. Retrieved May 20, 2021, from https://bitinfocharts.com/comparison/Bitcoin-difficulty.html#1y

Bitcoin Fork History. (n.d.). Unisot. Retrieved May 21, 2021, from https://unisot.com/why-Bitcoin-sv/

Bitcoinsv for Developers. (n.d.). Bitcoinsv. Retrieved May 21, 2021, from https://Bitcoinsv.io/

Bitcoin SV (Satoshi Vision). Bitcoinsv [Source code]. Retrieved March 31, 2021, from https://github.com/Bitcoin-sv/Bitcoin-sv

Bitcoinsv Wiki. (n.d.). Bitcoinsv. Retrieved May 21, 2021, from https://wiki.Bitcoinsv.io

Bitcoin Wiki. (n.d.). Bitcoin Community. Retrieved May 21, 2021, from https://en.Bitcoin.it

Frankenfield, J. (2021, March 26). *Mt.Gox.* Investopia. https://www.investopedia.com/terms/m/mt-gox.asp

Frankenfield, J. (2021, May 14). *Silk Road (Website).* Investopia. https://www.investopedia.com/terms/s/silk-road.asp

How Many Bitcoin Users Are There? (n.d.). Buy Bitcoin Worldwide. Retrieved May 27, 2021, from https://www.buyBitcoinworldwide.com/how-many-Bitcoin-users/

Huang, R. (2019, April 26). *How Bitcoin and WikiLeaks Saved Each Other*. Forbes. https://www.forbes.com/sites/rogerhuang/2019/04/26/how-Bitcoin-and-wikileaks-saved-each-other/?sh=6b42079374a5

Munro, A. (2021, April 20). *Exchanges that support BSV*. Finder. https://www.finder.com/ca/how-to-buy-Bitcoin-sv

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.* https://nakamotoinstitute.org/static/docs/bitcoin.pdf

Nguyen, J. (n.d.). Bitcoin Association. *Bitcoin SV: The Regulation-Friendly Bitcoin.* https://www.iadclaw.org/assets/1/7/6_Bitcoin_SV_The_Regulation-Friendly_Bitcoin.pdf

Node Explorer. (2021, May 20). Blockchair. https://blockchair.com/Bitcoin/nodes

Protocol. (n.d.). Lexico [Definition of protocol]. Retrieved May 27, 2021, from https://www.lexico.com/definition/protocol

Sandberg, E. (2020, November 9). *The Average Number of Credit Card Transactions Per Day & Year.* https://www.cardrates.com/advice/number-of-credit-card-transactions-per-day-year/#worldwide

Satoshi Nakamoto Institue. (n.d.). [Satoshi Nakamoto's forum posts, email transcripts and original code]. https://nakamotoinstitute.org/

The cryptography Archives. (n.d.). Metzdowd. [Emailing list which Satoshi was a member of]. https://www.Metzdowd.com/pipermail/cryptography/

Annex

Annex 1: Energy calculations

The following describes the set of assumptions and calculations used to determine the energy consumed by a SHA256 blockchain on a per block basis. This simple model provides an estimate; further considerations are necessary to build a more complete model of a SHA256 blockchain's power consumption.

The following assumptions were made for these energy calculations:

- There are no additional resources required, such as heating, ventilation, and air conditioning ("HVAC"), lighting, or other network devices requiring power. This assumption is strong; however, individual cases will vary based on the size of a particular miner and their geographic location. For example, mining operations in Brazil will require larger HVAC systems than mining operations in northern Canada. To address this in the future, additional research should go into determining the scale of mining operations and their geographic locations.
- Miners are using the best technology available to them. This reduces the size of the set of available mining equipment. For the purposes of this calculation, Bitmain's Antminer S19j that is capable of mining $90^{TH}/s$[77] has been selected.
- This is meant to be a static calculation. It gives us a point in time estimate of the power consumed for a given size of the network in question. An improvement on this would be to examine it as a dynamic system which would allow for changes in network participation, and changes in total network hash rate amongst other contributing factors.

To calculate the energy consumption per block the following variables were used:

Input Variable	Fixed Inputs	Outputs
► Current network hash rate ("HR"), hash/sec ► Mining Equipment Hash Power ("MHP"), hash/sec ► Mining Equipment Energy ("MEE"), watts ► Average Block Time ("ABT"), hours	► Hash adjustment factor ("HAF")	► Total Mining Units required ("TMU") ► Block energy consumption ("BEC"), watts/hour

[77] *Antminer s19j – 90TH/s*. Bitmain. Retrieved on May 25, 2021, from https://m.bitmain.com/product/00020210224195530399kqcF32sc06B9

Then, for any SHA256 blockchain:

$$TMU = \frac{HR}{MHP * HAF}$$

$$BEC = ABT * TMU * MEE$$

Block energy consumption can then be used to examine energy per transaction or scaled to and hourly power consumption of the entire network.

Results[78]:

Bitcoin Alpha (January 1, 2010)	Bitcoin Core (May 1, 2021)	Bitcoin Satoshi's Vision (May 1, 2021)
HR = 7.255 MH/s MHP*= 1 0.00145194 MH/ BEC = 64.96 kWh[79] = 6.496x10^{-5}GWh	HR = 136.27 EH/s MHP = 0.00009 EH/s BEC = 0.47 GWh	HR = 765.31 PH MHP = 0.09 PH BEC = 0.0026 GWh

*assuming CPU mining; intel i7-980X[80]

[78] Hash rate from: https://bitinfocharts.com/comparison/hashrate-btc-bsv.html
[79] Power usage from https://ark.intel.com/content/www/us/en/ark/products/47932/intel-core-i7-980x-processor-extreme-edition-12m-cache-3-33-ghz-6-40-gt-s-intel-qpi.html
[80] *How profitable is mining.* BetterHash. Retrieved on May 25, 2021, from https://www.betterhash.net/Intel(R)-Core(TM)-i7-CPU-X-980-@-3.33GHz-mining-profitability-77699.html

Annex 2: OpCodes

This annex compares the OpCodes between Bitcoin Satoshi's Vision v1.04 and Bitcoin Core V 0.21.0.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
Constants	0	Yes	OP_0, OP_FALSE	An empty array of bytes is pushed onto the stack. (This is not a no-op: an item is added to the stack.)	Yes	OP_0, OP_FALSE	An empty array of bytes is pushed onto the stack. (This is not a no-op: an item is added to the stack.)
	1-75	Yes	Pushdata Bytelength	The next OpCode byte is data to be pushed onto the stack.	No	NA	The next OpCode byte is data to be pushed onto the stack.
	76	Yes	OP_PUSHDATA1	The next byte contains the number of bytes to be pushed onto the stack.	Yes	OP_PUSHDATA1	The next byte contains the number of bytes to be pushed onto the stack.
	77	Yes	OP_PUSHDATA2	The next two bytes contain the number of bytes to be pushed onto the stack in little endian order.	Yes	OP_PUSHDATA2	The next two bytes contain the number of bytes to be pushed onto the stack in little endian order.
	78	Yes	OP_PUSHDATA4	The next four bytes contain the number of bytes to be pushed onto the stack in little endian. One byte has 8 bits. Therefore, 4 bytes have 32 bits. You can represent binary base, octal base, hex base number systems. The number system can represent (2^32) 4294967296 numbers. In the context of data this is about 4 gigabytes.	Yes	OP_PUSHDATA4	The next four bytes contain the number of bytes to be pushed onto the stack in little endian. One byte has 8 bits. Therefore, 4 bytes have 32 bits. You can represent binary base, octal base, hex base number systems. The number system can represent (2^32) 4294967296 numbers. In the context of data this is about 4 gigabytes.
	79	Yes	OP_1NEGATE	The number -1 is pushed onto the stack.	Yes	OP_1NEGATE	The number -1 is pushed onto the stack.

[81] OpCodes for BSV retrieved on May 21, 2021, from https://wiki.bitcoinsv.io/index.php/Opcodes_used_in_Bitcoin_Script
[82] OpCodes for BTC retrieved on May 21, 2021, from https://en.bitcoin.it/wiki/Script

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	81	Yes	OP_1, OP_TRUE	The number 1 is pushed onto the stack.	Yes	OP_1, OP_TRUE	The number 1 is pushed onto the stack.
	82-96	Yes	OP_2-OP_16	The number in the word name (2-16) is pushed onto the stack.	Yes	OP_2-OP_16	The number in the word name (2-16) is pushed onto the stack.
Flow control	97	Yes	OP_NOP	Does nothing.	Yes	OP_NOP	Does nothing.
	98	No	OP_VER DISABLED	Puts the version of the protocol under which this transaction will be evaluated onto the stack. (This OpCode is scheduled to be re-enabled in the Chronicle update.)	Yes	OP_VER THIS IS PART OF RESERVE WORDS	Transaction is invalid unless occurring in an unexecuted OP_IF branch
	99	Yes	OP_IF	If the top stack value is not FALSE, the statements between IF and ELSE are executed. If the top stack value is FALSE, the statements between ELSE and ENDIF are executed. The top stack value is removed.	Yes	OP_IF	If the top stack value is not false, the statements are executed. The top stack value is removed.
	100	Yes	OP_NOTIF	If the top stack value is FALSE, the statements between IF and ELSE are executed. If the top stack value is not FALSE the statements between ELSE and ENDIF are executed. The top stack value is removed.	Yes	OP_NOTIF	If the top stack value is false, the statements are executed. The top stack value is removed.
	101	No	OP_VERIF DISABLED	If the top stack value is EQUAL to the version of the protocol under which this transaction will be evaluated, the statements between IF and ELSE are executed.	Yes	OP_VERIF THIS IS PART OF RESERVE WORDS	Transaction is invalid even when occurring in an unexecuted OP_IF branch.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
				If the top stack value is NOT EQUAL to the version of the protocol under which this transaction will be evaluated, the statements between ELSE and ENDIF are executed. The top stack value is removed. (This opcode is scheduled to be re-enabled in the Chronicle update.)			
	102	No	OP_VERNOTIF DISABLED	If the top stack value is NOT EQUAL to the version of the protocol under which this transaction will be evaluated, the statements between IF and ELSE are executed. If the top stack value is EQUAL to the version of the protocol under which this transaction will be evaluated, the statements between ELSE and ENDIF are executed. The top stack value is removed. (This OpCode is scheduled to be re-enabled in the Chronicle update.)	Yes	OP_VERNOTIF THIS IS PART OF RESERVE WORDS	Transaction is invalid even when occurring in an unexecuted OP_IF branch.
	103	Yes	OP_ELSE	If the preceding OP_IF or OP_NOTIF or OP_ELSE was not executed, then these statements are and if the preceding OP_IF or OP_NOTIF or OP_ELSE was	Yes	OP_ELSE	If the preceding OP_IF or OP_NOTIF or OP_ELSE was not executed, then these statements are and if the preceding OP_IF or OP_NOTIF or OP_ELSE was executed then these statements are not.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
				executed then these statements are not.			
	104	Yes	OP_ENDIF	Ends an IF/ELSE block. All blocks must end, or the transaction is invalid. An OP_ENDIF without a prior matching OP_IF or OP_NOTIF is also invalid.	Yes	OP_ENDIF	Ends an IF/ELSE block. All blocks must end, or the transaction is invalid. An OP_ENDIF without OP_IF earlier is also invalid.
	105	Yes	OP_VERIFY	Marks transaction as invalid if top stack value is not true. The top stack value is removed.	Yes	OP_VERIFY	Marks transaction as invalid if top stack value is not true. The top stack value is removed.
	106	Yes	OP_RETURN	OP_RETURN can also be used to create "False Return" outputs with a scriptPubKey consisting of OP_FALSE OP_RETURN followed by data. Such outputs are probably unspendable and should be given a value of zero Satoshis. These outputs can be pruned from storage in the UTXO set, reducing its size. Currently the BitcoinSV network supports multiple FALSE RETURN outputs in a given transaction with each one capable of holding up to 100 kB of data. After the Genesis upgrade in 2020 miners will be free to mine transactions containing FALSE RETURN outputs of any size.	Yes	OP_RETURN	Marks transaction as invalid. Since Bitcoin 0.9, a standard way of attaching extra data to transactions is to add a zero-value output with a scriptPubKey consisting of OP_RETURN followed by data. Such outputs are provably unspendable and specially discarded from storage in the UTXO set, reducing their cost to the network. Since 0.12, standard relay rules allow a single output with OP_RETURN, that contains any sequence of push statements (or OP_RESERVED[1]) after the OP_RETURN provided the total scriptPubKey length is at most 83 bytes.
Stack	107	Yes	OP_TOALTSTACK	Puts the input onto the top of the alt stack. Removes it from the main stack.	Yes	OP_TOALTSTACK	Puts the input onto the top of the alt stack. Removes it from the main stack.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	108	Yes	OP_FROMALTSTACK	Puts the input onto the top of the main stack. Removes it from the alt stack.	Yes	OP_FROMALTSTACK	Puts the input onto the top of the main stack. Removes it from the alt stack.
	109	Yes	OP_2DROP	Removes the top two stack items.	Yes	OP_2DROP	Removes the top two stack items.
	110	Yes	OP_2DUP	Duplicates the top two stack items.	Yes	OP_2DUP	Duplicates the top two stack items.
	111	Yes	OP_3DUP	Duplicates the top three stack items.	Yes	OP_3DUP	Duplicates the top three stack items.
	112	Yes	OP_2OVER	Copies the pair of items two spaces back in the stack to the front.	Yes	OP_2OVER	Copies the pair of items two spaces back in the stack to the front.
	113	Yes	OP_2ROT	The fifth and sixth items back are moved to the top of the stack.	Yes	OP_2ROT	The fifth and sixth items back are moved to the top of the stack.
	114	Yes	OP_2SWAP	Swaps the top two pairs of items.	Yes	OP_2SWAP	Swaps the top two pairs of items.
	115	Yes	OP_IFDUP	If the top stack value is not 0, duplicate it.	Yes	OP_IFDUP	If the top stack value is not 0, duplicate it.
	116	Yes	OP_DEPTH	Counts the number of stack items onto the stack and places the value on the top.	Yes	OP_DEPTH	Puts the number of stack items onto the stack.
	117	Yes	OP_DROP	Removes the top stack item.	Yes	OP_DROP	Removes the top stack item.
	118	Yes	OP_DUP	Duplicates the top stack item.	Yes	OP_DUP	Duplicates the top stack item.
	119	Yes	OP_NIP	Removes the second-to-top stack item.	Yes	OP_NIP	Removes the second-to-top stack item.
	120	Yes	OP_OVER	Copies the second-to-top stack item to the top.	Yes	OP_OVER	Copies the second-to-top stack item to the top.
	121	Yes	OP_PICK	The item n back in the stack is copied to the top.	Yes	OP_PICK	The item n back in the stack is copied to the top.
	122	Yes	OP_ROLL	The item n back in the stack is moved to the top.	Yes	OP_ROLL	The item n back in the stack is moved to the top.
	123	Yes	OP_ROT	The top three items on the stack are rotated to the left.	Yes	OP_ROT	The 3rd item down the stack is moved to the top.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	124	Yes	OP_SWAP	The top two items on the stack are swapped.	Yes	OP_SWAP	The top two items on the stack are swapped.
	125	Yes	OP_TUCK	The item at the top of the stack is copied and inserted before the second-to-top item.	Yes	OP_TUCK	The item at the top of the stack is copied and inserted before the second-to-top item.
Data manipulation	126	Yes	OP_CAT	Concatenates two strings.	No	OP_CAT DISABLED	Concatenates two strings.
	127	Yes	OP_SPLIT	Split byte sequence x at position n.	No	OP_SUBSTR DISABLED	Returns a section of a string. Disabled.
	128	Yes	OP_NUM2BIN	Converts numeric value a into byte sequence of length b.	No	OP_LEFT DISABLED	Keeps only characters left of the specified point in a string. Disabled.
	129	Yes	OP_BIN2NUM	Converts byte sequence x into a numeric value.	No	OP_RIGHT DISABLED	Keeps only characters right of the specified point in a string. Disabled.
	130	Yes	OP_SIZE	Pushes the string length of the top element of the stack (without popping it).	Yes	OP_SIZE	Pushes the string length of the top element of the stack (without popping it).
Bitwise logic	131	Yes	OP_INVERT	Flips all of the bits in the input.	No	OP_INVERT DISABLED	Flips all of the bits in the input.
	132	Yes	OP_AND	Boolean and between each bit in the inputs.	No	OP_AND DISABLED	Boolean and between each bit in the inputs.
	133	Yes	OP_OR	Boolean or between each bit in the inputs.	No	OP_OR DISABLED	Boolean or between each bit in the inputs.
	134	Yes	OP_XOR	Boolean exclusive or between each bit in the inputs.	No	OP_XOR DISABLED	Boolean exclusive or between each bit in the inputs.
	135	Yes	OP_EQUAL	Returns 1 if the inputs are exactly equal, 0 otherwise.	Yes	OP_EQUAL	Return 1 if the inputs are exactly equal, 0 otherwise.
	136	Yes	OP_EQUALVERIFY	Same as OP_EQUAL, but runs OP_VERIFY afterward.	Yes	OP_EQUALVERIFY	Same as OP_EQUAL, but runs OP_VERIFY afterward.
Arithmetic	139	Yes	OP_1ADD	1 is added to the input.	Yes	OP_1ADD	1 is added to the input.
	140	Yes	OP_1SUB	1 is subtracted from the input.	Yes	OP_1SUB	1 is subtracted from the input.
	141	No	OP_2MUL DISABLED	The input is multiplied by 2. (This opcode is scheduled to be re-enabled in the Chronicle update.)	No	OP_2MUL DISABLED	The input is multiplied by 2. Disabled.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	142	No	OP_2DIV DISABLED	The input is divided by 2. (This opcode is scheduled to be re-enabled in the Chronicle update.)	No	OP_2DIV DISABLED	The input is divided by 2. Disabled.
	143	Yes	OP_NEGATE	The sign of the input is flipped.	Yes	OP_NEGATE	The sign of the input is flipped.
	144	Yes	OP_ABS	The input is made positive.	Yes	OP_ABS	The input is made positive.
	145	Yes	OP_NOT	If the input is 0 or 1, it is flipped. Otherwise, the output will be 0.	Yes	OP_NOT	If the input is 0 or 1, it is flipped. Otherwise, the output will be 0.
	146	Yes	OP_0NOTEQUAL	Returns 0 if the input is 0. Otherwise 1.	Yes	OP_0NOTEQUAL	Returns 0 if the input is 0. Otherwise 1.
	147	Yes	OP_ADD	a is added to b.	Yes	OP_ADD	a is added to b.
	148	Yes	OP_SUB	b is subtracted from a.	Yes	OP_SUB	b is subtracted from a.
	149	Yes	OP_MUL	a is multiplied by b.	No	OP_MUL DISABLED	a is multiplied by b.
	150	Yes	OP_DIV	a is divided by b.	No	OP_DIV DISABLED	a is divided by b.
	151	Yes	OP_MOD	Returns the remainder after dividing a by b.	No	OP_MOD DISABLED	Returns the remainder after dividing a by b.
	152	Yes	OP_LSHIFT	Logical left shift b bits. Sign data is discarded.	No	OP_LSHIFT DISABLED	Logical left shift b bits. Sign data is discarded.
	153	Yes	OP_RSHIFT	Logical right shift b bits. Sign data is discarded.	No	OP_RSHIFT DISABLED	Logical right shift b bits. Sign data is discarded.
	154	Yes	OP_BOOLAND	If both a and b are not 0, the output is 1. Otherwise 0.	Yes	OP_BOOLAND	If both a and b are not 0, the output is 1. Otherwise 0.
	155	Yes	OP_BOOLOR	If a or b is not 0, the output is 1. Otherwise 0.	Yes	OP_BOOLOR	If a or b is not 0, the output is 1. Otherwise 0.
	156	Yes	OP_NUMEQUAL	Returns 1 if the numbers are equal. Otherwise 0.	Yes	OP_NUMEQUAL	Returns 1 if the numbers are equal. Otherwise 0.
	157	Yes	OP_NUMEQUALVERIFY	Same as OP_NUMEQUAL, but runs OP_VERIFY afterward.	Yes	OP_NUMEQUALVERIFY	Same as OP_NUMEQUAL, but runs OP_VERIFY afterward.
	158	Yes	OP_NUMNOTEQUAL	Returns 1 if the numbers are not equal. Otherwise 0.	Yes	OP_NUMNOTEQUAL	Returns 1 if the numbers are not equal. Otherwise 0.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	159	Yes	OP_LESSTHAN	Returns 1 if a is less than b. Otherwise 0.	Yes	OP_LESSTHAN	Returns 1 if a is less than b. Otherwise 0.
	160	Yes	OP_GREATERTHAN	Returns 1 if a is greater than b. Otherwise 0.	Yes	OP_GREATERTHAN	Returns 1 if a is greater than b. Otherwise 0.
	161	Yes	OP_LESSTHANOREQUAL	Returns 1 if a is less than or equal to b. Otherwise 0.	Yes	OP_LESSTHANOREQUAL	Returns 1 if a is less than or equal to b. Otherwise 0.
	162	Yes	OP_GREATERTHANOREQUAL	Returns 1 if a is greater than or equal to b. Otherwise 0.	Yes	OP_GREATERTHANOREQUAL	Returns 1 if a is greater than or equal to b. Otherwise 0.
	163	Yes	OP_MIN	Returns the smaller of a and b.	Yes	OP_MIN	Returns the smaller of a and b.
	164	Yes	OP_MAX	Returns the larger of a and b.	Yes	OP_MAX	Returns the larger of a and b.
	165	Yes	OP_WITHIN	Returns 1 if x is within the specified range (left-inclusive). Otherwise 0.	Yes	OP_WITHIN	Returns 1 if x is within the specified range (left-inclusive). Otherwise 0.
Cryptography	166	Yes	OP_RIPEMD160	The input is hashed using RIPEMD-160.	Yes	OP_RIPEMD160	The input is hashed using RIPEMD-160.
	167	Yes	OP_SHA1	The input is hashed using SHA-1.	Yes	OP_SHA1	The input is hashed using SHA-1.
	168	Yes	OP_SHA256	The input is hashed using SHA-256.	Yes	OP_SHA256	The input is hashed using SHA-256.
	169	Yes	OP_HASH160	The input is hashed twice: first with SHA-256 and then with RIPEMD-160.	Yes	OP_HASH160	The input is hashed twice: first with SHA-256 and then with RIPEMD-160.
	170	Yes	OP_HASH256	The input is hashed two times with SHA-256.	Yes	OP_HASH256	The input is hashed two times with SHA-256.
	171	Yes	OP_CODESEPARATOR	All of the signature checking words will only match signatures to the data after the most recently executed OP_CODESEPARATOR.	Yes	OP_CODESEPARATOR	All of the signature checking words will only match signatures to the data after the most recently executed OP_CODESEPARATOR.
	172	Yes	OP_CHECKSIG	The entire transaction's outputs, inputs, and script (from the most recently executed OP_CODESEPARATOR to the end) are hashed. The signature used by OP_CHECKSIG must be a valid	Yes	OP_CHECKSIG	The entire transaction's outputs, inputs, and script (from the most recently executed OP_CODESEPARATOR to the end) are hashed. The signature used by OP_CHECKSIG must be a valid signature for this hash and public key. If it is, 1 is returned. Otherwise 0.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
				signature for this hash and public key. If it is, 1 is returned. Otherwise 0.			
	173	Yes	OP_CHECKSIGVERIFY	Same as OP_CHECKSIG, but OP_VERIFY is executed afterward.	Yes	OP_CHECKSIGVERIFY	Same as OP_CHECKSIG, but OP_VERIFY is executed afterward.
	174	Yes	OP_CHECKMULTISIG	Compares the first signature against each public key until it finds an ECDSA match. Starting with the subsequent public key, it compares the second signature against each remaining public key until it finds an ECDSA match. The process is repeated until all signatures have been checked or not enough public keys remain to produce a successful result. All signatures need to match a public key. Because public keys are not checked again if they fail any signature comparison, signatures must be placed in the scriptSig using the same order as their corresponding public keys in the scriptPubKey or redeemScript. If all signatures are valid, 1 is returned. Otherwise 0. Due to a bug, an extra unused value (x) is removed from the stack. Script spenders must account for this by adding a junk value (typically zero) to the stack.	Yes	OP_CHECKMULTISIG	Compares the first signature against each public key until it finds an ECDSA match. Starting with the subsequent public key, it compares the second signature against each remaining public key until it finds an ECDSA match. The process is repeated until all signatures have been checked or not enough public keys remain to produce a successful result. All signatures need to match a public key. Because public keys are not checked again if they fail any signature comparison, signatures must be placed in the scriptSig using the same order as their corresponding public keys in the scriptPubKey or redeemScript. If all signatures are valid, 1 is returned. Otherwise 0. Due to a bug, one extra unused value is removed from the stack.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	175	Yes	OP_CHECKMULTISIGVERIFY	Same as OP_CHECKMULTISIG, but OP_VERIFY is executed afterward.	Yes	OP_CHECKMULTISIGVERIFY	Same as OP_CHECKMULTISIG, but OP_VERIFY is executed afterward.
Used NOP opcode identifiers	177	Yes	OP_NOP2 (previously OP_CHECKLOCKTIMEVERIFY)	NO OPERATION evaluation process for UTXOs that predate Genesis: Mark transaction as invalid if the top stack item is greater than the transaction's nLockTime field, otherwise script evaluation continues as though an OP_NOP was executed. Transaction is also invalid if: the stack is empty; the top stack item is negative; the top stack item is greater than or equal to 500,000,000 while the transaction's nLockTime field is less than 500,000,000, or vice versa; or the input's nSequence field is equal to 0xffffffff. The precise semantics are described in BIP 0065.	Yes	OP_CHECKLOCKTIMEVERIFY (previously OP_NOP2)	Marks transaction as invalid if the top stack item is greater than the transaction's nLockTime field, otherwise script evaluation continues as though an OP_NOP was executed. Transaction is also invalid if the stack is empty; the top stack item is negative; the top stack item is greater than or equal to 500.000.000 while the transaction's nLockTime field is less than 500.000.000, or vice versa; or the input's nSequence field is equal to 0xffffffff. The precise semantics are described in BIP 0065.

Group	Op-Codes	Used in BSV Language	BSV Script OpCodes[81]	Descriptions	Used in BTC Language	BTC Script OpCodes[82]	Descriptions
	178	Yes	OP_NOP3 (previously OP_CHECKSEQUENCEVERIFY)	NO OPERATION evaluation process for UTXOs that predate Genesis: Mark transaction as invalid if the relative lock time of the input (enforced by BIP 0068 with nSequence) is not equal to or longer than the value of the top stack item. The precise semantics are described in BIP 0112.	Yes	OP_CHECKSEQUENCEVERIFY (previously OP_NOP3)	Marks transaction as invalid if the relative lock time of the input (enforced by BIP 0068 with nSequence) is not equal to or longer than the value of the top stack item. The precise semantics are described in BIP 0112.
Pseudo-words	253	Yes	OP_PUBKEYHASH	Represents a public key hashed with OP_HASH160.	Yes	OP_PUBKEYHASH	Represents a public key hashed with OP_HASH160.
	254	Yes	OP_PUBKEY	Represents a public key compatible with OP_CHECKSIG.	Yes	OP_PUBKEY	Represents a public key compatible with OP_CHECKSIG.
	255	Yes	OP_INVALIDOPCODE	Matches any OpCode that is not yet assigned.	Yes	OP_INVALIDOPCODE	Matches any OpCode that is not yet assigned.
Reserved words	80	Yes	OP_RESERVED	Transaction is invalid unless occurring in an unexecuted OP_IF branch.	Yes	OP_RESERVED	Transaction is invalid unless occurring in an unexecuted OP_IF branch.
	137	Yes	OP_RESERVED1	Transaction is invalid unless occurring in an unexecuted OP_IF branch.	Yes	OP_RESERVED1	Transaction is invalid unless occurring in an unexecuted OP_IF branch.
	138	Yes	OP_RESERVED2	Transaction is invalid unless occurring in an unexecuted OP_IF branch.	Yes	OP_RESERVED2	Transaction is invalid unless occurring in an unexecuted OP_IF branch.
	176, 179-185	Yes	OP_NOP1, OP_NOP4-OP_NOP10	The word is ignored. Does not mark transaction as invalid.	Yes	OP_NOP1, OP_NOP4-OP_NOP10	The word is ignored. Does not mark transaction as invalid.

Annex 3: Source code timeline

The following table provides a timeline of the development and release of the Bitcoin software from v 0.1.0 to v 0.1.5. Included are key contributors, a context of where these changes came from, and insights on how this relates to Satoshi's Nakamoto's vision for the Bitcoin network.

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
	Satoshi Nakamoto;	Satoshi posted that he had been working on a new electronic cash system that is fully peer-to-peer, with no trusted third-party. He linked his whitepaper for the first time and described some main properties.	Satoshi made his first post with his whitepaper.	There was no source code at this time.	Satoshi stated: "The main properties: Double-spending is prevented with a peer-to-peer network. No mint or other trusted parties. Participants can be anonymous. New coins are made from Hashcash style PoW. The PoW for new coin generation also powers the network to prevent double-spending."	https://www.metzdowd.com/pipermail/cryptography/2008-October/014810.html	2008-10-31
	Satoshi Nakamoto; James A. Donald;	Someone stressed the plausibility of handling immense number of transactions with Satoshi's proposed electronic cash system. Satoshi speaks to the possible number of transactions a network he is proposing could handle. He says: "If the network were to get that big, it would take several years, and by then, sending 2 HD movies over the Internet would probably not seem like a big deal." He refers to his idea of a p2p e-cash system.	James was questioning the scalability of Satoshi's idea with regards to double spending, the number of transactions, and bandwidth.	There was no source code at this time.	Satoshi envisioned a network that could indeed handle as many transactions as Visa.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014815.html	2008-11-02
	Satoshi Nakamoto; James A. Donald; John Levine; Ray Dillinger;	Satoshi is not worried about the risk of zombie farms overpowering the network. He even suggests zombie farms may contribute to the network and generate bitcoin instead. Satoshi also explains how even if there were to be a double spend, someone would only be able to "take money back he himself spent, like bouncing a check." He also suggests someone would make more by generating bitcoin than attacking the system.	Satoshi and several others are discussing th threat of someone having more CPU powr than the rest of the honest nodes.	There was no source code at this time.	Satoshi speaks about the large farms. He does not directly talk about centralization, but does acknowledge large mining farms were a possibility.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014818.html	2008-11-03
	Satoshi Nakamoto; Anonymous;	"Governments are good at cutting off the heads of a centrally controlled networks like Napster, but pure P2P networks like Gnutella and Tor seem to be holding their own. "	No one knows the context of this reply as the conversation was ot made public.	There was no source code at this time.	Satoshi envisioned a pure peer-to-peer network.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014823.html	2008-11-06

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
	Satoshi Nakamoto; Ray Dillinger;	Satoshi talks about how the system will handle increasing hardware speed. He states the difficulty of generating coins will proportionally increase keeping the production constant. And that is why there is a known number of Bitcoins created every year in the future.	Ray had doubts about the scaling of the network from a technological standpoint.	There was no source code at this time.	Satoshi suggests creating a constant rate at which coins are initially distributed "seems like the best formula." Difficulty of generating Bitcoins will move in parallel with the time and advancement of computing power.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014831.html	2008-11-08
	Satoshi Nakamoto; Hal Finney;	Satoshi explains that even if transactions do not get added to a block immediately it will be held in a working set until it is added to a block. He also suggests the receiver of transactions will normally need to wait for "perhaps" an hour or more to allow the verification of a transaction that has been spent on two different branches. He then continued to explain the concept of the longest valid chain and how transactions are dependent on only other valid transactions or transactions in the same block.	Satoshi is answering a lot of Hal Finney's questions regarding the system.	There was no source code at this time.	Satoshi viewed the possibility of Bitcoin being used for buying goods and "immediately" being able to re-spend bitcoin. However, the receiver of said bitcoin "should wait before taking action such as shipping goods." To ensure there is enough time to validate the transaction.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014832.html	2008-11-08
	Satoshi Nakamoto; James A. Donald;	Satoshi explains the PoW chain concept. "Once a transaction is hashed into a link that is a few links back in the chain, it is firmly etched into the global history."	Satoshi answering questions.	There was no source code at this time.	N/A	https://www.metzdowd.com/pipermail/cryptography/2008-November/014833.html	2008-11-08
	Satoshi Nakamoto; James A. Donald;	Satoshi explains to James that Bitcoin does not require inflation.	Satoshi answered questions regarding inflation issue.	There was no source code at this time.	Satoshi suggests adding frees to the blocks as incentives for the PoW concept.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014842.html	2008-11-09
	Satoshi Nakamoto; James A. Donald;	If there are multiple double-spent versions of same transaction, only one will become valid. Target time between blocks will probably be 10 minutes. Every block includes its creation time. If the time is off by more than 36 hours, other nodes won't work on it. If the timespan over the last 6*24*30 blocks is less than 15 days, blocks are being generated too fast and the proof of work difficulty doubles. Everyone does the same calculation with the same chain data, so they all get the same result at the same link in the chain." Satoshi explains that Bitcoin can validate transactions much faster than cheques and credit cards.	Satoshi continuing to explain exact details.	There was no source code at this time.	Target time between blocks of 10 minutes. Every block includes its creation time. If the time is off by more than 36 hours, other nodes won't work on it. If the timespan over the last 6*24*30 blocks is less than 15 days, blocks are being generated too fast and the proof-of-work difficulty doubles. Everyone does the same calculation with the same chain data, so they all get the same result at the same link in the chain. Transactions are irreversible in one to two hours.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014843.html	2008-11-10
	Satoshi Nakamoto; Hal Finney; James A. Donald;	Satoshi gives Byzantine General's Problem explanation to James. Satoshi concludes that the PoW chain is how everything is distributed and synchronized.	Satoshi gives Byzantine General's Problem explanation.	There was no source code at this time.	N/A	https://www.metzdowd.com/pipermail/cryptography/2008-November/014849.html	2008-11-13

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
Bitcoin Pre-Release	Satoshi Nakamoto; Hal Finney; James A. Donald;	Satoshi states it is only important to have a pending transaction pool for the current best branch. When new blocks arrive, they remove transactions from that pool. If a different branch becomes longer there is a re-organization, which he states would be rare. Networks broadcasts are reliable with TCP transmissions and a retry mechanism.	Satoshi confirms / clarifies statements made by everyone else.	There was no source code at this time.	Bitcoin is very attractive to the libertarian viewpoint. Reorganizations of the branches are rare. Bitcoin will use TCP transmissions. Blocks must propagate much faster than it takes to generate them.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014853.html	2008-11-14
	Satoshi Nakamoto; Ray Dillinger; James A. Donald;	Buyers are the only member digitally signing transactions. All ties in chains of equal length are broken by keeping the earliest one received. All double spends are immediately rejected. Chain domination is purely based on proportional share of CPU power.	Satoshi confirms / clarifies statements made by everyone else.	There was no source code at this time.	"The PoW is a Hashcash style SHA-256 collision finding."	https://www.metzdowd.com/pipermail/cryptography/2008-November/014858.html	2008-11-14
	Satoshi Nakamoto; Ray Dillinger;	New key pair for every transaction. Bitcoin is pseudonymous in the sense of the next action on a coin can be identified as being from the owner of that coin. Credentials that establish someone as real is the ability to provide CPU power. Satoshi also clarifies how people could prevent being scammed by double spending (waiting 2 minutes).	Satoshi confirms / clarifies statements made by Ray.	There was no source code at this time.	New key pair for every transaction. Bitcoin is pseudonymous in the sense of the next action on a coin can be identified as being from the owner of that coin. Credentials that establish someone as real is the ability to provide CPU power.	https://www.metzdowd.com/pipermail/cryptography/2008-November/014860.html	2008-11-15

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
	Satoshi Nakamoto; James A. Donald;	Satoshi was clarifying more details about his idea and stated the source code would be coming soon. He also stated he would send the main files per people's request.	Satoshi sent SOME of the source code files to James (main files) and possibly others.	Main header file: • The minimal PoW difficulty was not set yet (it was set to 40 in commented code). • Transactions contain multiple inputs and outputs (vector of input transactions and vector of output transactions) • Presence of other connected nodes — presence of a Merkle branches — each block contains hash of the previous block and hash of the Merkle root • Contains Merkle tree in the memory — "Nodes collect new transactions into a block, hash them into a hash tree, and scan through nonce values to make the block's hash satisfy PoW requirements. When they solve the PoW, they broadcast the block to everyone and the block is added to the time chain. The first transaction in the block is a special one that creates a new coin owned by the creator of the block." • - "The time chain is a tree-shaped structure starting with the genesis block at the root, with each block potentially	N/A	https://www.metzdowd.com/pipermail/cryptography/2008-November/014863.html	2008-11-17
	Satoshi Nakamoto; Nicolas Williams;	Satoshi explains Bitcoin IS NOT anonymous, it is pseudonymous. To detect a double spend, the network DOES NOT need to come to a final consensus, only an approximate consensus.	Satoshi explaining how a double spend is identified.	No source code update.	Bitcoin IS NOT anonymous, it is pseudonymous. To detect a double, spend, the network DOES NOT need to come to a final consensus, only an approximate consensus	https://www.metzdowd.com/pipermail/cryptography/2008-November/014866.html	2008-11-17

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
	Satoshi Nakamoto	The official version is released after more than a year and a half of development. This row is based on the analysis of the original README file	This row is based on the analysis of the original README file.	Bitcoin v0.1.0 ALPHA	"Bitcoin is an electronic cash system that uses a peer-to-peer network to prevent double-spending. It's completely decentralized with no server or central authority." "The time to generate a block varies each time, but may take days or months, depending on the speed of your computer and the competition on the network." This indicates Bitcoin is meant to be run on a standard user's computer.	https://satoshi.nakamotoinstitute.org/emails/cryptography/16/#selection-9.0-9.21	2009-01-08
Bitcoin v0.1.0 ALPHA	Satoshi Nakamoto	After more than a year and a half of development, the official version is released. This row is based on the analysis of the original UI.	Doing a UI analysis of the first version.	Bitcoin v0.1.0 ALPHA	Mining is straightforward. All one must do to mine is clicking generate tokens. A user can see all previous transactions they made with its debits and credits along with a description. A user can create multiple public H18 the comments suggest, "You may want to give a different one (address) to each sender so you can keep track of who is paying you." To send coins you can either enter an IP address if the receiver is online or a Bitcoin address if the recipient is offline. There is an optional text message box to transmit comments. There is a custom select menu that has only ONE transfer option ("standard"). The use of this drop-down box could suggest there would be more options for transferring bitcoin. The SENDER of bitcoin is able to determine the transaction fee. IT IS FULLY OPTIONAL.	https://satoshi.nakamotoinstitute.org/emails/cryptography/16/#selection-9.0-9.21	2009-01-08

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
	Satoshi Nakamoto;	Satoshi released the first public version in his post. He states the basic functionality and the use of the application. He also warns the software is still experimental.	Analyzing the first post of the first version of Bitcoin.	Bitcoin v0.1.0 ALPHA	"Announcing the first release of Bitcoin, a new electronic cash system that uses a peer-to-peer network to prevent double-spending. It's completely decentralized with no server or central authority." "Generated coins must wait 120 blocks to mature before they can be spent." "Total circulation will be 21,000,000 coins. It'll be distributed to network nodes when they make blocks, with the amount cut in half every 4 years." "When that runs out, the system can support transaction fees if needed. It's based on open market competition, and there will probably always be nodes willing to process transactions for free."	https://satoshi.nakamotoinstitute.org/emails/cryptography/16/#selection-9.0-9.21	2009-01-08
	Satoshi Nakamoto; Dustin D. Trammell;	Dustin brings up the biggest challenge for Bitcoin: "to get people to actually value [it]". Satoshi responds that Bitcoin can be used initially for micropayments on sites and games. It's already available for pay-to-send email.	Satoshi is replying to the reply of Dustin from Satoshi's original post.		Satoshi continues to refer to small-scale applications for Bitcoin. He has the idea that someone can effortlessly pay a few cents.	https://www.metzdowd.com/pipermail/cryptography/2009-January/015014.html	2009-01-16
Bitcoin v0.1.2	Satoshi Nakamoto;	Update on Bitcoin version.	Updating bugs.	Bitcoin v0.1.2: Bugs fixed: • Fixed various problems that were making it hard for new nodes to see other nodes to connect to. • - If you're behind a firewall, it could only receive one connection, and the second connection would constantly disconnect and reconnect.	N/A	https://www.Bitcoin.com/satoshi-archive/emails/Bitcoin-list/2/	2009-01-11
Bitcoin v0.1.3	Satoshi Nakamoto;	Update on Bitcoin version.	Updating bugs.	Bitcoin v0.1.3 Fixed a problem where your node's communications could go dead after a while. The network is running much more smoothly now with this version.		https://satoshi.nakamotoinstitute.org/emails/Bitcoin-list/22/#selection-9.15-9.29	2009-01-12

Bitcoin version	Key contributors	Summary	Context	Source code update	Insights on the true Satoshi vision	Link to forum post	Forum post date
Bitcoin v0.1.5	Satoshi Nakamoto; Dustin D. Trammell; Nicholas Bohm;	First major update.	Updating bugs Adding features.	Bitcoin v0.1.5 Changes: • Disk full warning • Fixed a bug that could occur if DNS lookup failed • Prevent entering your own address in the address book, which confusingly changed the label for your own address • Moved change address button to menu under options • Tweaks to make it get connected faster • Close sockets on exit • Created minimum fee for transactions less than 1 cent • Hid the transaction-type selection box that only had one choice • Cleaned up ParseMoney a little • Slightly cleaner reformatting of text message • Changed the font in transaction details dialog • Added some explanation text to transaction details for generated coins • - Reworded the description for transactions received with Bitcoin address	There is now a minimum transaction fee for transactions under one cent. (This does get removed in future updates. It was removed as it caused confusion but limited the risk of DoS attacks.) Removed the transaction type selection that only had one choice. All these updates improve are scaling the software as needed.	https://sourceforge.net/p/Bitcoin/mailman/Bitcoin-list/thread/CHILKAT-MID-0e05a16e-6ede-06d8-6d65-e873c53b3a42%40server123/#msg21500063	2009-02-04
	Satoshi Nakamoto;	Satoshi says next release will take advantage of multiple processors to generate blocks. Will also add interfaces to make it easier to integrate into websites from any server-side language.	Satoshi replying to a question asking what is next for Bitcoin.	No change.	Satoshi wants to enable multiple processors to generate blocks. Satoshi wants to make mining more efficient / faster.	https://sourceforge.net/p/Bitcoin/mailman/message/21646307/	2009-02-22

Annex 4: Risk and control framework based on the whitepaper

The diagram below illustrates a risk and control framework which was used to analyze the various key layers of the Bitcoin protocol. The framework is similar to how one would assess different elements of a network protocol using layer of abstraction similar to the Open System Interconnection (OSI)[83] Model which was also used to compare and contrast similarities and differences between internet protocol (TCP/IP) and the original Bitcoin protocol.

	Step 1 Broadcast	Step 2 Block of Transactions	Step 3 Proof of Work (PoW)	Step 4 Broadcast PoW	Step 5 Accept New Block	Step 6 New Block in Chain	Commentary
Phase Description	New transactions are broadcast to all nodes.	Each node collects new transactions into a block.	Each node works on finding a difficult PoW for its block.	When a node finds a PoW, it broadcasts the block to all nodes.	Nodes accept the block only if all transactions in it are valid and not already spent.	Nodes express their acceptance of the block by working on creating the next block in the chain, using previous hash.	
Incentives	Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.	First transaction in a block is a special transaction that starts a new coin owned by the creator of the block and is an incentive for nodes to support the network.	Never the need to extract a complete standalone copy of a transaction's history.				► Any needed rules and incentives can be enforced with consensus mechanism. ► A predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees.
Risks and Problems	△ Two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first.	△ Node does not receive a block. △ Linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.	△ Greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins.	△ Traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party and the necessity to announce all transactions publicly precludes this method.			► Framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. ► As long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker.
Controls	✓ Nodes always consider the longest chain to be the correct one. ✓ Nodes work on the first one they received, but save the other branch in case it becomes longer. ✓ New transaction broadcasts do not necessarily need to reach all nodes (As long as they reach many nodes, they will get into a block before long). ✓ Strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency.	✓ Block broadcasts are also tolerant of dropped messages. ✓ If a node does not receive a block, it will request it when it receives the next block and realizes it missed one. ✓ Distribute coins into circulation by adding first transaction and starting coin. ✓ Allow value to be split and combined, transactions contain multiple inputs and outputs. ✓ Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts and at most two outputs: one for the payment, and one returning the change back to the sender.	✓ Tie will be broken when the next PoW is found and one branch becomes longer. ✓ System is deigned to be more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth. ✓ Only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in.	✓ Run their own nodes for more independent security and quicker verification. ✓ privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. ✓ A new key pair should be used for each transaction to keep them from being linked to a common owner.	✓ Transactions are hashed in a Merkle Tree , with only the root included in the block's hash. ✓ Old blocks can then be compacted by stubbing off branches of the tree. ✓ The interior hashes do not need to be stored.	✓ The network is robust in its unstructured simplicity. ✓ Nodes work all at once with little coordination. ✓ They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis.	► Proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. ► Output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. ► A predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.
Key Products	New transactions enter here and are broadcast. Initial broadcasts.	Blocks.	PoW.	PoW broadcast.	New Blocks.	New Chain (Longest Chain).	

[83] The Open Systems Interconnection (OSI) model is a conceptual framework that describes the functions of a networking or telecommunication system.
The model uses layers to help give a visual description of what is going on with a particular networking system. This can help narrow down problems and help design system to leverage capabilities and functionalities of the protocol.

Annex 5: Highlights of major Bitcoin and BSV protocol changes

The following describes a set of important changes throughout the evolution of the Bitcoin protocol which, in our opinion, moved the protocol away from Satoshi's original vision. BSV, in its evolution, made changes to align their protocol closer to Satoshi's original vision for Bitcoin. The set of changes described is not exhaustive, but are, in our opinion, some of the more important changes.

As Bitcoin evolved from the first release, there were numerous hard and soft forks to the protocol, described by Bitcoin Improvement Proposals[84] ("BIPs") which altered its underlying code and functionality. Many of these would be considered minor improvements or bug fixes and remaining consistent with Satoshi's vision, but there are several notable changes which altered underlying functionality to such a level that we consider these to be major deviations from Satoshi's original vision for Bitcoin.

BIP 65 – OP_CHECKLOCKTIMEVERIFY ("CLTV")
BIP 65's description on Github is as follows:

"This BIP describes a new opcode (OP_CHECKLOCKTIMEVERIFY or abbreviated to CLTV) for the Bitcoin scripting system that allows a transaction output to be made unspendable until some point in the future".[85]

With CLTV, when the transaction is created, the options (including when it will occur) are specified in the transaction and recorded on-chain. CLTV puts the sender in control of how the recipient receives the transaction. The recipient is not able to decide. Because the transaction is recorded on-chain, it is publicly announced in advance.

Within the original Bitcoin, Satoshi included a field nLockTime in the transaction to allow for open transactions that could be replaced with newer transactions up to the deadline specified. With nLockTime, any number of signed transactions could be prepared and selected by the recipient for release at the later time, with the highest version at the deadline being mined and recorded on-chain.

The nLockTime OpCode allows for flexibility and options for the recipients which would be managed via smart contracts in script. Script could also provide functionality so double spends would be prevented. This functionality was not enabled at the time of the original release, but supported for future use cases that would be made possible through smart contracts, such as escrow-type transactions.[86]

84 https://github.com/bitcoin/bips/blob/master/README.mediawiki
85 https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki
86 https://bitcointalk.org/index.php?topic=1786.msg22119#msg22119

The key differences between nLockTime and CLTV are :

1. Control/Flexibility – With nLockTime, the recipient is in control of which option is released at a future time, whereas with CLTV the options are set and recorded on-chain with the transaction.
2. Visibility – nLockTime transactions are kept off-chain until they are released at the time specified, which keeps them private. CLTV transactions are visible as they are recorded on-chain when they are created, which limits flexibility.

Another major issue with CLTV is that it is not considered backward compatible with previous releases. The functionality of CLTV is in contrast to Satoshi's vision as it limits the users' ability to create different types of transactions.

BIP 16 – Pay to Script Hash - P2PH

BIP 16's description on the bitcoin wiki is provided as follows:

"This BIP describes a new "standard" transaction type for the Bitcoin scripting system, and defines additional validation rules that apply only to the new transactions.

The purpose of pay-to-script-hash is to move the responsibility for supplying the conditions to redeem a transaction from the sender of the funds to the redeemer.

The benefit is allowing a sender to fund any arbitrary transaction, no matter how complicated, using a fixed-length 20-byte hash that is short enough to scan from a QR code or easily copied and pasted."[87]

The P2SH transactions effectively enable the hiding of output scripts by allowing transactions to be sent to a script hash which are then only spendable when the recipient provides the script that matches the script hash. This moves responsibility for satisfying the conditions from the sender to the recipient and, because the scripts are supplied by the recipient, security is essentially unknown to the sender.

There is a very large overall transparency issue created by P2SH in that if the details of the script are not known, especially the payee information, they cannot be third-party audited unless the redeem script is made available.

This violates Satoshi's original vision in two ways:

1. With P2SH, there is information hiding of the recipient (payee) as well as conditions in the recipient's scripts, which goes contrary to the principle of a 'public history of transactions', which includes transaction details.
2. P2SH can allow for robust privacy practices to be bypassed via the recipient scripts, which are normally ensured by a true, publicly visible peer to peer transaction workflow as envisioned by Satoshi.

87 https://en.bitcoin.it/wiki/BIP_0016

Block size – Several BIPs that were not accepted

Early Bitcoin developers recognized the issues involved in having a limited block size, which included (amongst others):

- A limited number of transactions per block (thereby limiting overall throughput of BTC transactions to approximately 7tps).
- Limitations on how much data could be included within each transaction and block.
- Increased fee per transaction due to the limited number of transactions per block.

BIP 101 proposed to replace the fixed 1MB maximum block size with a maximum size that expands over the years at a predictable rate. According to the BIP 101 proposal, the maximum block size would increase to 8MB in January 2016 and double in size every 730 days until January 2036. Though BIP 101 provided a solution to the block size issue, it failed to pull enough support from other Bitcoin developers. Large mining pools were greatly interested in the proposal, but that wasn't enough to convince Bitcoin core developers to support the movement and it was withdrawn.

Subsequent BIPs (including 102,103,104,105, 106, 107, and 109) attempted to increase block size through various mechanisms but were rejected. Bitcoin's block size remains at 1MB and many of the early limitations identified still exist.

BSV's re-alignment with Satoshi's original vision: Genesis protocol

In February 2020, BSV released the Genesis protocol that included several significant improvements to restore BSV to Bitcoin's original functionality, including removing block size limitations, removing P2SH, and restoring functionality of nLockTime.

Block size
Satoshi established Bitcoin with the intent to have block sizes much larger than 1MB. With the Genesis upgrade to BSV, this limit will still exist as a configuration option for miners however the default will be changed to no limit at all. After Genesis it was the miner's responsibility to manage this limit if they choose to impose it at all.

Remove P2SH support
The Pay-to-script-hash (or P2SH) is a mechanism introduced to Bitcoin to enable hiding of output scripts at the time they are created. This change removes the ability to run transactions using P2SH. Any existing P2SH coins will be unaffected, so there is no need to sweep old wallets. This change simply prevents any new P2SH outputs from being made, re-aligning the protocol to Satoshi's original vision.

Restore nLocktime functionality
The nLocktime data field is a key part of the mechanism of payment channels that Satoshi describes as a fundamental mechanism for allowing high speed micropayments on Bitcoin. nLocktime was repurposed by BTC developers by the new op code CLTV (see discussion above). Along with removing this op code, the original usage of nLockTime was restored in the Genesis protocol upgrade, which more closely aligned BSV with Satoshi's original vision.

PRAXITY™
Empowering Business Globally

Wherever business takes you

MNP.ca