The Bitcoin Universe: An Architectural Overview of the Bitcoin Blockchain

Paul Mueller[1,2], Sonja Bergsträßer[2], Amr Rizk[2] and Ralf Steinmetz[2]

Abstract: On January 2009, the emergence of Bitcoin surprised the world with a new idea involving decentralized secure money transfers outside the ecosystem of FIAT currencies. The concepts behind the Bitcoin architecture the blockchain can, however, be extended to a much wider range of economic assets than just digital currencies. In general, a blockchain is a distributed, verifiable database, which operates through a confluence of public-key cryptography, the concept of *proof of work* and P2P-systems.

In the following article, a detailed overview of the concepts underlying the Bitcoin architecture is given. The ecosystem as a whole is discussed, starting with some historical aspects. We consider the cryptographic background as given here, as this article discusses mainly the basic concepts for key generation. Here, we focus on the concept of transactions in the scope of Bitcoin which we break down into single attributes such as the locking and unlocking scripts. Both, the mining process and the consensus mechanism are examined. Furthermore, the drawbacks of the whole system and the proposed remedies to these via Bitcoin Improvement Proposals (BIP) are outlined. Finally, we address several applications based on the blockchain as well as the question of anonymity.

Keywords: Bitcoin, blockchain, proof-of-work (PoW), Bitcoin keys, Transactions scripts, Mining, Consensus, Bitcoin drawbacks, Bitcoin applications

1 Introduction

The "big bang" of the Bitcoin blockchain took place on January 3rd, 2009, when the first Bitcoin block, the genesis block, was established at 18:15:05 GMT. The genesis block is the only block in the blockchain which is hard-coded within the source code of Bitcoin, rather than the result of the mining process. However, the story of Bitcoin started in 2008, when the domain name "bitcoin.org" was registered anonymously. The basic ideas of the blockchain were then published[3] by "Satoshi Nakamoto" on Friday, October 31st,[4] 2008 at 18:10:00 UTC under the title "Bitcoin: A Peer-to-Peer Electronic Cash System" [SN08]. The author(s) of this document is/are still unknown.

The complexity of the Bitcoin ecosystem comes from it aims, i.e., that is that anyone should be able to write to the Bitcoin blockchain, and that there should be no centralized

[1] University Kaiserslautern, Integrated Communication Systems Lab (ICSY), Paul Ehrlichstraße 34, 67663 Kaiserslautern, pmueller@informatik.uni-kl.de
[2] TU Darmstadt, KOM Multimedia Communications Lab, Rundeturmstraße 10, 64283 Darmstadt, {Vorname.Nachname}@KOM.TU-Darmstadt.de
[3] http://article.gmane.org/gmane.comp.encryption.general/12588/
[4] Halloween

control. The Bitcoin ecosystem can be viewed as a network of replicated databases, where each database contains the same list of previous Bitcoin transactions. Full nodes (nodes who runs the full stack of the Bitcoin protocol) of the network are called “miners”, and these propagate “transaction data” (payments) and “block data” (additions to the ledger). Each miner independently checks the transaction and block data passed to it. There are rules in place (the Bitcoin protocol) to make the network operates as intended. The complexity of the Bitcoin architecture arises from its aims, which are to be decentralized, that is, to have no single point of control, and to be highly secure and anonymous. This has influenced how Bitcoin has developed. All blockchain ecosystems need not have the same mechanisms, especially if participants can be identified and trusted to behave (e.g. in a private blockchain).

The outline of this paper is as follows: Firstly, several historical remarks are made; these are followed by a description of the architectural design principles involved. This description starts with Bitcoin keys, followed by a detailed description of transactions and the mining process, and finishes with a discussion of the consensus approach used in Bitcoin and the vulnerability of this architecture. Next, the drawbacks of Bitcoin architecture are described, followed by countermeasures to these drawbacks as put forward in Bitcoin Improvement Proposals (BIP). The last chapter concerns Bitcoin applications, where some concrete applications are introduced. In the conclusion of the paper, we discuss the question of anonymity.

2 The Architecture of Bitcoin

With respect to the architecture (see Fig. 1) of the Bitcoin blockchain, several important design aspects must be taken into account:

1. The Bitcoin application itself
2. The role of nodes constituting the overall blockchain network, and the node discovery process
3. Transactions, which make up the blocks running in the nodes
4. The security implementation that generates the blocks
5. The process of adding new blocks to the chain.

The blockchain itself runs on a network of distributed servers. The core application is a transaction database modeled as a secure ledger. This is shared by all nodes (servers) that run the full stack of the Bitcoin protocol. It is thus a decentralized transaction system acting as a highly transparent ledger. Any full node running the blockchain protocol runs the entire blockchain locally.

After installing the full stack of the software, the blockchain client syncs up with the other nodes in the network, in a peer-to-peer fashion [SW05]. Hence, that node maintains all Bitcoin transactions (or any other application running on the blockchain). The integrity and chronological order of transactions (and the addresses owning the currency) are enforced by cryptographic rules.

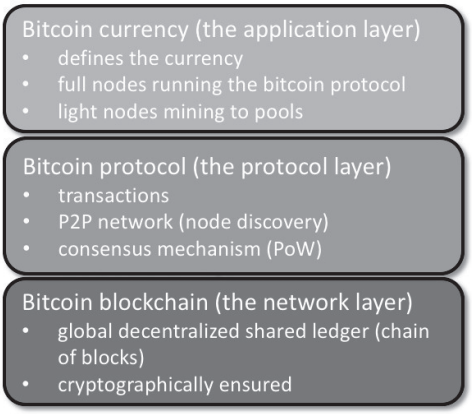


Fig. 1: The Bitcoin Architecture

The nodes in the overall network use a peer-to-peer IP network to process and verify transactions. Nodes that have the same blocks in their individual databases are considered to be in consensus [DW13].

2.1 The Design of the Bitcoin Blockchain

We now take a closer look at the Bitcoin blockchain. As mentioned above, the first block of the Bitcoin blockchain was not a result of the Bitcoin consensus mechanism (mining) but was hard-coded into the source code. It is a special case, in the sense that it does not reference a previous block, and for Bitcoin and almost all of its spinoffs, it produces an unspendable subsidy (for a detailed description of the genesis block, see https://en.bitcoin.it/wiki/Genesis_block).

At least one parameter of the Genesis block is worth a deeper look. The "coinbase parameter" (coded in hex) contains, along with normal data, the following text:

"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

This may be intended as proof that the block was created on or after January 3rd, 2009, as mentioned above; it could also form an argument for a new currency, due to the instability caused by traditional banking.

Another architectural design decision was the limited total number of Bitcoins, resulting from the speed of mining (evaluating) a new block and the reward that a miner can earn from the mining process. On average, a new block is mined every 10 minutes (regardless of the technology used, as discussed later), and the mining reward at the beginning of the system was 50 Bitcoins per block. The block reward is halved every four years (or every 210,000 blocks on average). A simple calculation shows that there will be a total of 21 million Bitcoins available as reward for miners.

Another design decision was the block size. Currently, the block size is restricted to 1MB (on average), and one block can therefore cover around 4000 transactions with an average size of 250 bytes. This results in an overall rate of about seven transactions per seconds (tps). Compared to PayPal (around 200 tps) and VISA (4,000–40,000 tps), this is a fairly low rate. An examination of https://blockchain.info/ gives information on how many transactions are currently included in each block, although there are some blocks with only one transaction, meaning that the miner included no transactions except for their own reward transaction[5].

2.2 The Steps of a Bitcoin Transaction

Transactions can be broadcast by any node in the system at any time. The decision on which transactions of those broadcasted to be included in a new block is dependent on the node (the miner) running the ***proof-of-work*** **(PoW)** algorithm, since the miners are responsible for picking a transaction from the so-called *mem-pool* where all validated transactions[6] are stored, grouping them and including them in the block. The selection of transactions by the miner depends on the transaction fee (the standard fee is 1.000 Satoshi = 10 μBTC = 0.01 mBTC = 0.0001 BTC per kB), which forms a reward for the miner's efforts in addition to the coinbase reward. The priority of picking up a transaction depends on the miner; in general, miners prefer larger fees and smaller transactions, and often prioritize in this way. Transaction portions that is not spent towards recipient of back to the sender are included as a fee. Fees are paid to miners and can be used to increase the speed of transaction confirmation by incentivizing miners to prioritize the transaction(s).

To initiate a transaction, a user must generate the necessary Bitcoin keys (see Fig. 2), starting with a random 256-bit private key (see Step 1). This **private key** is needed to sign a transaction and thus transfer (spend) Bitcoins. The elliptic curve DSA algorithm is then used to generate a 512-bit **public key** from the private key (see Step 2). This public key is used to **verify** the signature for a transaction, although the public key is not revealed until a transaction is signed. Since the 512-bit public key is inconveniently large, it is hashed down to 160 bits using the SHA-256 and RIPEMD hash algorithms (see Step 3).

[5] https://blockchain.info/block-height/315076

[6] validated transaction: all transactions (payments) which follows the Bitcoin protocol rules

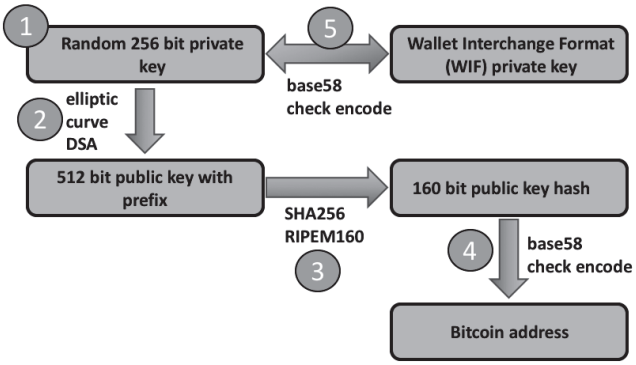


Fig. 2: Bitcoin keys and their relationships[7]

The key is then encoded in ASCII using Bitcoin's custom Base58Check encoding (see Step 4). The resulting address is the **Bitcoin address**, which is published in order for a user to receive Bitcoins. Note that neither the public key nor the private key can be determined from the Bitcoin address. If the private key is lost (for instance, by throwing a hard drive[8] away), the Bitcoins are lost forever.

When all keys have been generated, a **transaction** can be carried out. A transaction enables a transfer of Bitcoins and is broadcast to the network[9] nodes. Transactions are the main building blocks of the Bitcoin system. The Bitcoin architecture is designed to make sure that transactions can be transparently added to the global ledger of transactions. Here, there is even no need to trust the nodes used to broadcast the transaction.

Transactions are data structures that encode a transfer of value between participants in the Bitcoin system. Each transaction is a public entry in the Bitcoin blockchain, and a transaction chain is formed (see Fig. 3), meaning that all users can see every Bitcoin transaction from the genesis block onwards.

[7] Adopted from http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html
[8] https://www.theguardian.com/technology/2013/nov/27/hard-drive-bitcoin-landfill-site
[9] There are several scripts available in Bitcoin to handle the value transfer from the source to the destination. These scripts can handle very simple to relatively complex transactions, depending upon requirements.

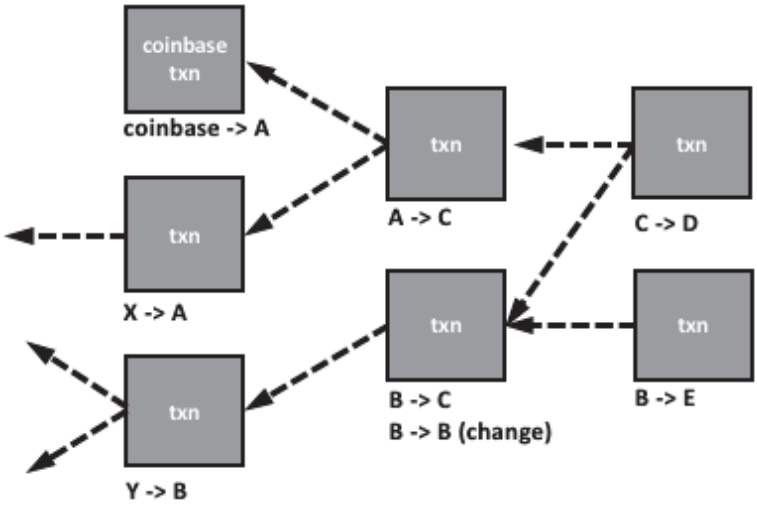


Fig. 3: The Bitcoin transaction chain[10]

A transaction usually references previous transaction outputs as new transaction inputs and uses all input Bitcoin values. Thus, one key element of a Bitcoin transaction is an unspent transaction output (UTXO). UTXOs are indivisible pieces of Bitcoin currency locked to a specific owner that are recorded in the blockchain. Although a UTXO can have any arbitrary value, it is indivisible once created, just like a coin that cannot be cut in pieces. When a user receives a Bitcoin value, the amount is recorded within the blockchain as a UTXO. Thus, one Bitcoin may be scattered in the form of UTXOs across hundreds of transactions and hundreds of blocks. There are no accounts or balances in Bitcoin; there are only dispersed UTXOs, locked to specific owners in the transaction chain. This means that each input used must be entirely spent in a transaction. If an address received 12.5 Bitcoins and wants to spend only 12.0 Bitcoins, the transaction must spend all 12.5. Hence, one uses a second output (the change address) for ***the difference***, which returns the 0.5 leftover Bitcoins back to the sender (see Fig.4).

Transactions in the blockchain are not encrypted, and it is therefore possible to browse and view every transaction ever made, each of which is stored in a block related to the Bitcoin address rather than the name of the user. Once transactions receive sufficient confirmations, they can be considered irreversible. All transactions are visible (the transaction chain which gives the history of ownership) in the blockchain (which gives the

[10] Adopted from http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html

transaction ordering in time to avoid double spending) and can be viewed with a hex editor. The transaction itself has the general format shown in Fig. 5.[11]

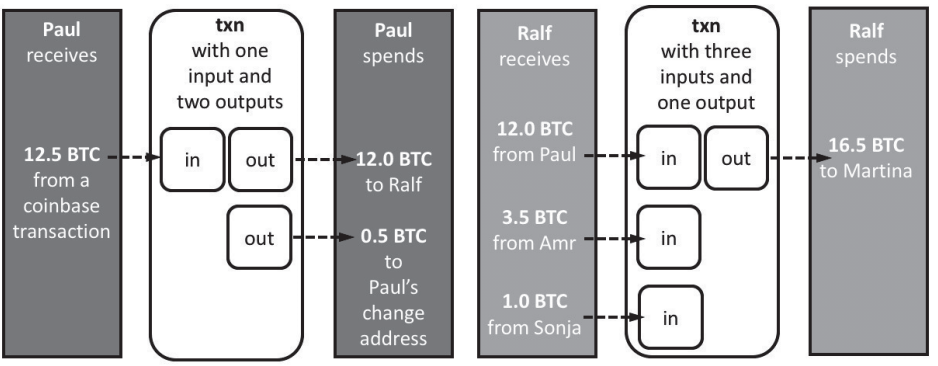


Fig. 4: Input and outputs of a transaction

In general, a Bitcoin transaction is composed of three parts:

- **An input:** This is a record of the Bitcoin address from which the Bitcoins to be spent were initially received.
- **An amount:** This is the specific amount of Bitcoin to be sent.
- **An output:** This is the receiver's public key, also known as a 'Bitcoin address.'

The transaction **input** (or list of inputs) is a pointer to a UTXO using a reference to the transaction hash and sequence number of the record in the blockchain. To spend a UTXO, a transaction input also includes an **unlocking script** that satisfies the spending conditions set by the UTXO. The unlocking script is usually a signature proving the ownership of the Bitcoin address in the locking script. The unlocking script is called *scriptSig*, because it usually contains a digital signature.

A transaction **output** (or list of outputs) comprises, first, an amount of Bitcoins, denominated in satoshis (the smallest Bitcoin unit), and secondly, a **locking script** to "lock" this amount by specifying the conditions that must be met to spend the output. The locking script is called *scriptPubKey*, because it usually contains a public key or Bitcoin address.

[11] From https://en.bitcoin.it/wiki/Transaction#general_format_.28inside_a_block.29_of_each_input_of_a_transaction_-_Txin

field	description	size
Version no	currently 2	4 bytes
In-counter	positive integer VI = VarInt	1 – 9 bytes
List of inputs	the first input of the first transaction is also called "coinbase"	<in-counter> many inputs
Out-counter	positive integer VI = VarInt	1 – 9 bytes
List of outputs	the outputs of the first transaction spend the mined bitcoins for the block	<out-counter> many outputs
Lock_time	if non-zero and sequence numbers are < 0xFFFFFFFF: block height or timestamp when transaction is final	4 bytes

Fig. 5: General transaction format (from: https://en.bitcoin.it/wiki/Transaction)

Breaking down a concrete transaction script (with only a single input and output) and deserializing it, we obtain the following structure, where values are displayed in hexadecimal.

```
01  {"hash":"8ac3455…",
02  "ver":1,
03  "vin_sz":1,
04  "vout_sz":1,
05  "lock_time":0,
06  "size":224,
07  "in":[
08    {"prev_out":
09      {"hash":"aee433…",
10      "n":0},
11    "scriptSig":<sig> <pubKey>}],
12  "out":[
13    {"value":"0.50000000",
14     "scriptPubKey":"OP_DUP OP_HASH160 <pubKeyHash?>
                      OP_EQUALVERIFY OP_CHECKSIG"}]}
```

Algorithm 1: Inside a simple transaction (one input and one output)

In the following, algorithm 1 is explained line by line. In Line 1, we see the remainder of a transaction, `8ac3455...`. This is used as an identifier for the transaction, followed by the version number of the Bitcoin protocol in Line 2 and the number of inputs and outputs (one of each in this case) in Lines 3 and 4. The `lock_time` parameter in Line 5 controls when a transaction should be finalized. For most Bitcoin transactions carried out today, the `lock_time` is set to 0, which means the transaction is finalized immediately. The size of the transaction in bytes is described in Line 6.

Lines 7 to 11 define the transaction input, which is taken from the output of an earlier transaction, and uses the hash of this earlier transaction in Line 9. The parameter n in Line 10 denotes that this is the first output from that transaction. The *scriptSig* parameter in Line 11 contains the signature of the sender, followed by a space, and then the corresponding public key (the unlocking script).

Lines 12 to 14 give the output of the transaction: Line 13 gives the value of the output, and Line 14 contains the *scriptPubKey* (the unlocking script) value with the Bitcoin address (<pubKeyHash?>) of the intended recipient of the funds. The other parameters used here will be described later.

To verify a transaction, the Bitcoin validation engine relies on two types of scripts: a *locking script* and an *unlocking script*, as shown in Fig. 7.

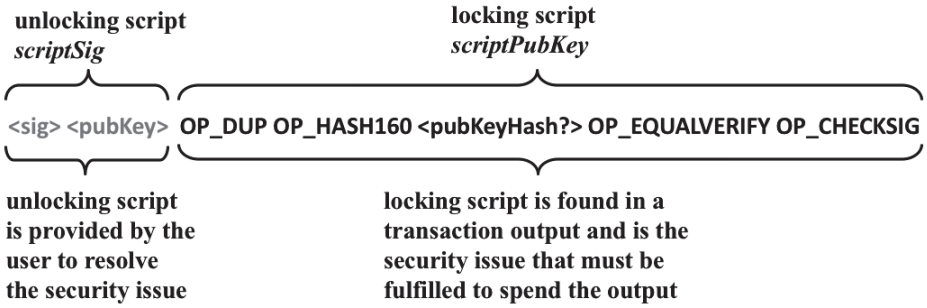


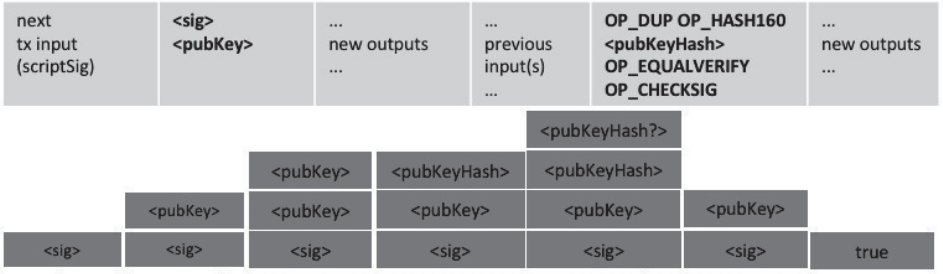
Fig. 6: Bitcoin locking and unlocking scripts (from: Mastering Bitcoin [AA17])

Here, only the standard 'pay to public key hash' (P2PKH) transaction will be discussed. P2PKH is the most commonly used transaction type, and is used to send transactions to Bitcoin addresses. Bitcoin uses a simple stack-based[12] language called a *script* to describe how Bitcoins can be spent and transferred. It is intentionally not Turing complete, and has no loops to avoid a condition where a script runs forever. This scripting language uses a reverse Polish notation in which every operand is followed by its operators. It is evaluated from left to right, using a 'last in first out' (LIFO) stack. A script uses various opcodes (operational codes) to define its operation.

Each node examines a transaction as it arrives, and then runs a series of checks to verify it. These checks[13] are defined in the protocol rules. Every Bitcoin node validates

[12] Note: The stacks hold byte vectors. When used as numbers, byte vectors are interpreted as little-endian variable-length integers, with the most significant bit determining the sign of the integer. Byte vectors are interpreted as Booleans, where False is represented by any representation of zero and True is represented by any representation of non-zero.
[13] https://en.bitcoin.it/wiki/Protocol_rules

transactions by executing the locking and unlocking scripts simultaneously[14]. For each input in the transaction, the validation software first retrieves the UTXO referenced by the input. This UTXO contains the *locking script* that defines the conditions required to spend it. The validation software then takes the *unlocking script* contained in the input attempting to spend this UTXO, and executes these two scripts. First, the UTXO is unlocked, and then it is spent. *scriptSig* is provided by the user who wishes to unlock the transaction, while *scriptPubKey* is part of the transaction output and specifies the conditions that need to be fulfilled in order to spend the output.



	stack	script	description
1	empty	<sig> <pubKey> OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	scriptSig and scriptPubKey are combined.
2	<sig><pubKey>	OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	constants are added to the stack.
3	<sig><pubKey> <pubKey>	OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	top stack item is duplicated.
4	<sig><pubKey> <pubKeyHash>	<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	top stack item is hashed
5	<sig><pubKey> <pubKeyHash> <pubKeyHash?>	OP_EQUALVERIFY OP_CHECKSIG	constant added
6	<sig><pubKey>	OP_CHECKSIG	equality is checked between the top two stack items
7	true	empty	signature is checked for top two stack items

Fig. 7: Bitcoin stack operation

A script is an instruction list paired with each transaction to describe how to gain access to the transferred Bitcoins. Fig. 8 shows a step-by-step execution of the combined script, which will determine whether or not this is a valid transaction. The execution starts (from left to right) by pushing the value <sig> onto the stack. Next, the value <pubKey> is

[14] In the original Bitcoin client, the unlocking and locking scripts were concatenated and executed in sequence. For security reasons, this was changed in 2010.

pushed onto the stack, and this is duplicated in the next step with the operator DUP. The operator HASH160 hashes the top item of the stack using RIPEMET160(SHA256(pubKey)), and pushes the resulting value onto the stack.

The value <pubKeyHash?> from the script is pushed on top of the value <pubKeyHash> previously calculated from the HASH160 of the <pubKey>. The EQUALVERIFY operator compares the <pubKeyHash?> from the script with the <pubKeyHash?> calculated from the user's <pubKey>. If they match, both are removed from the stack and execution is continued. Finally, the CHECKSIG operator checks that the signature <sig> matches the public key <pubKey> and pushes TRUE onto the stack if true. After successful validation, the transaction is copied into the memory pool (mem-pool) of the node and waits to be picked up and included into a block.

Every full node maintains a temporary list of unconfirmed transactions that is called the *mem-pool*. Nodes use this pool to keep track of transactions that are known to the network but have not yet been included in the blockchain. As Transactions that become part of a block and are added to the blockchain are considered "confirmed", allowing the new owners to spend the Bitcoin they received in these transactions. In the following, the process by which a transaction becomes part of a block is described.

2.3 The Bitcoin Blockchain: The Mining Process

The blockchain is a public ledger providing a time stamped, ordered, and immutable list of all transactions on the Bitcoin network. Due to the design decision in the Bitcoin network whereby each block cannot exceed a limit of 1MB, the average number of transactions per second is around seven, i.e., a fairly low number. Since transactions can be launched any time, the number of transactions waiting in the mem-pool[15] tends to increase, and reached about 50 million in January 2018.

After validation, the transaction must be included into a block. To do this, the miner (the node) picks transactions from the mem-pool, recursively hashing pairs of transaction hash values until there is only one remaining; this is called the Merkle root [RM80]. The cryptographic hash algorithm used in Bitcoin's Merkle trees is SHA256, applied twice, also known as double-SHA256.

We now take a closer look at a block respectively the block chain (see Fig. 9). Consider a block as a data structure that aggregates transactions. The block consists of a header with some metadata, mainly followed by the list of transactions. The block header has a size of 80 bytes, while the average transaction is at least 250 bytes and a block can contain up to 4000 transactions.

[15] Mem-pool size: https://blockchain.info/charts/mempool-size?

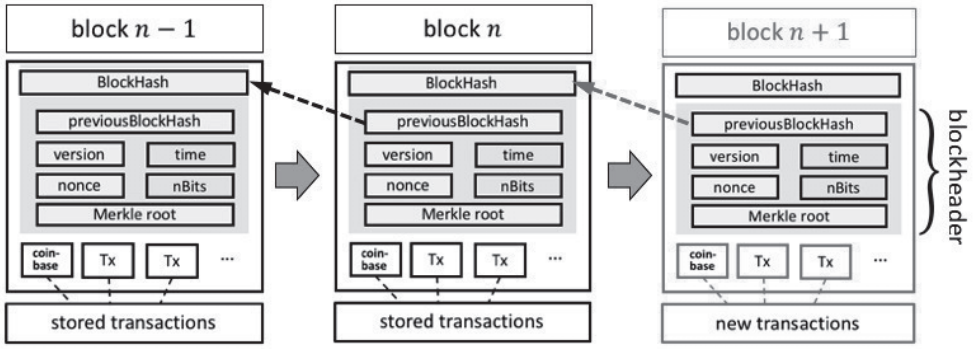


Fig. 8: The blockchain

The block header consists of a reference to a previous block hash; this connects the block to the previous block in the blockchain. The second set of metadata, containing the target, timestamp and nonce, relate to the mining competition. The third part of the metadata is the Merkle tree root, as described above.

When a miner (a node) has put these data together, it can start the process of linking the new block to the blockchain. As an incentive to miners to spend computing power in the mining process, they receive two types of rewards for mining. Firstly, they earn new coins, which are created with each new block in the so called *coinbase* transaction automatically. That grants the mining reward to the miner. Secondly, they receive transaction fees from the remaining transactions included in the block which are proportional to the size (1000 satoshis per kB).

To earn these rewards, all miners compete to solve a difficult mathematical puzzle based on a cryptographic hash algorithm. The solution to this problem, i.e., the proof of work (PoW), is copied into the new block. The PoW is seen as proof that a miner spent significant computing effort. The competition to solve the PoW algorithm to earn rewards and the right to add transactions to the blockchain is the basis of Bitcoin's security model.

To run the PoW, the miner first has to hash the block header, including the block hash of the previous block, and compare it with a predefined target which is stored in the nBits parameter of the header. Also the target is formally defined as $target = \mathbf{2^{224}/d}$ (where $\boldsymbol{d}$ denotes the difficulty). In the concrete implementation it can be retrieved from the *hex* representation of the block header (80 bytes in total) using the 73-76 byte. Because this number, however, is in little-endian the bytes have to be reversed. Now the target can be calculated from a compact scientific notation ($target = c*2^{(8\ (e-3))}$) where the first byte denotes the exponent e and the next 3 bytes the coefficient c. The proof of work is formally defined as:

$$\mathbf{PoW} = F_d(c|x) \rightarrow \mathbf{SHA256(SHA256}(h|x) \leqq \mathbf{2^{224}/d}$$

where $\boldsymbol{h}$ can be seen as a *challenge*, $\boldsymbol{x}$ the *nonce* and $\boldsymbol{d}$ the actual *difficulty* [RW16]. If $\boldsymbol{F_d(h|x)}$ is not smaller than the current target, the miner will modify the nonce (usually just incrementing it by one) and calculate the block hash again. At the current level of difficulty in the Bitcoin network, miners have to try quadrillions of times before finding a nonce that results in a low enough block header hash. The length of time it takes to mine a block can be controlled with the difficulty,[16] $\boldsymbol{d}$. In order to keep the block generation time to 10 minutes on average regardless of the technology (increasing compute power of miners) used, the difficulty of mining must be adjusted. In fact, difficulty is a dynamic parameter that is periodically (every 2016 blocks or every 14 days) adjusted to meet a 10-minute block target. Adjustment of the difficulty occurs automatically and independently on every full node after every 2016 blocks, and all nodes retarget the proof of work difficulty. The equation for retargeting difficulty compares the time to find the last 2,016 blocks to 20,160 minutes, i.e., the time needed based on the 10-minute average block generation time. This can be put in simple terms as follows: if the network is finding blocks faster than every 10 minutes, the difficulty increases; if block creation time is slower than expected, the difficulty decreases.

The last step in Bitcoin's consensus mechanism is the independent validation of each new block by every node in the network. As the newly solved block is propagated across the network, each node performs a series of tests to validate it before propagating it to its peers. This ensures that only valid blocks are propagated on the network. This independent validation also ensures that miners who act honestly have their blocks incorporated in the blockchain, thus earning the reward.

2.4 The Bitcoin Consensus

While mining is primarily incentivized by the generation of rewards, the underlying purpose of mining is not the reward or the generation of new coins but the achievement of a decentralized consensus in a trustless network. Mining enables a network-wide consensus without a central authority.

The first practical implementation of a distributed consensus in a trustless network was realized by Bitcoin, which uses public key cryptography with PoW based on hashcash[17] [AB02]. The key innovation here is the idea of an ordered list of blocks composed of transactions and cryptographically secured by the PoW mechanism. Bitcoin's decentralized consensus arises from the interplay of four processes occurring independently on nodes across the network:

[16] The block header also contains the difficulty target in a notation called "difficulty bits" or just "nBits." This is expressed in a coefficient/exponent format, with the first two hexadecimal digits representing the exponent and the next six hex digits the coefficient.

[17] http://www.hashcash.org/

- Independent verification of each transaction by every full node, based on a comprehensive list of criteria (https://en.bitcoin.it/wiki/Protocol_rules).
- Independent aggregation of these transactions into new blocks by mining nodes, coupled with computational efforts using a proof of work algorithm.
- Independent verification of the new blocks by every node, and assembly into a chain.
- Independent selection by every node of the chain with the most cumulative computational effort, demonstrated through proof of work.

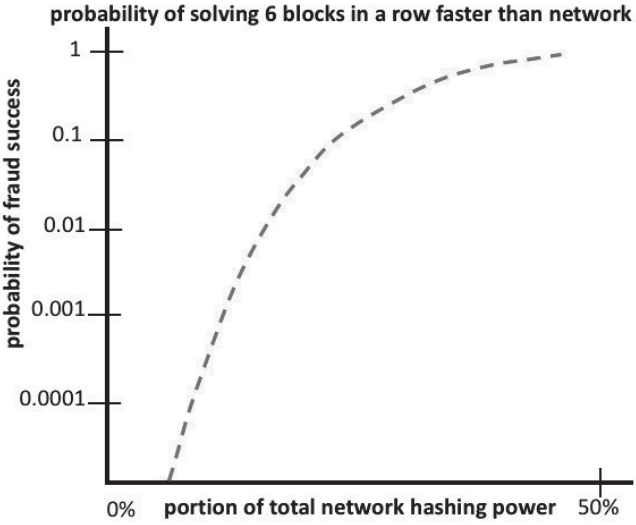


Fig. 9: The 51% attack (calculated from M. Rosenfeld [Ro12])

Because of the distributed nature of Bitcoin, two nodes may announce a valid block simultaneously, meaning that there exist two blockchains containing different transactions. This situation depends on the block creation time relative to the block propagation time, and is addressed by the Bitcoin network accepting only the longest chain. In this case, the shorter chain will be considered orphaned; this is a point at which the Bitcoin network is vulnerable.

This vulnerability is known as the 51% attack (see Fig. 9). In this scenario, a miner or group of miners (mining pools) that control a majority (51%) of the total network's hashing power can attack the Bitcoin blockchain. In this type of attacks, the attackers can leverage/cause forks in the blockchain for their own benefit. For example, one could double-spend transactions or "prohibit" transactions from executing. A double-spend attack is one where the attacker deliberately causes forks at a previous position in the

blockchain. With sufficient power, an attacker can invalidate six or more blocks in a row, causing transactions that were considered immutable to be invalidated.

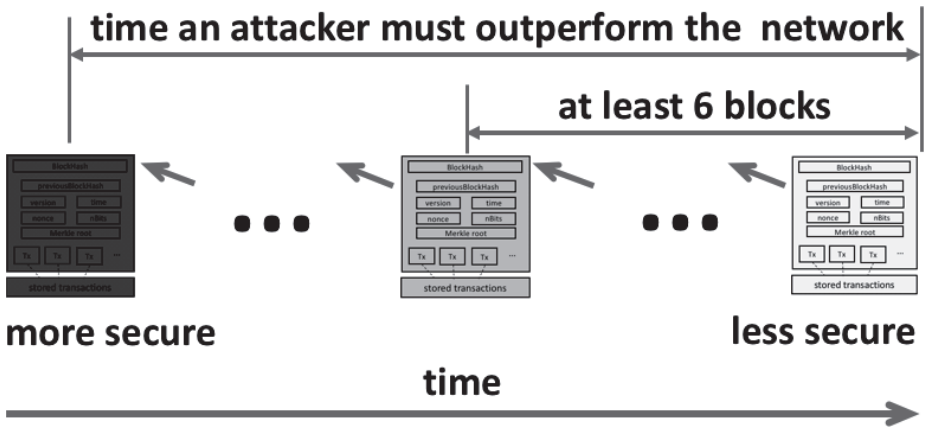


Fig. 10: Security timeline[18]

As explained above, the chance of a single miner solving a new block before the rest of the network, which takes 10 minutes on average, is very low. Even with substantial computing power, the older a transaction/block becomes, the harder it would be for an attacker to change it (Fig. 11).

3 The Drawbacks of Bitcoin

Although the Bitcoin network is the most well-known example of a cryptocurrency, there is currently a discussion about its technical drawbacks.

3.1 Defining the Drawbacks

Firstly, the fairly **small transaction time** of 7 tps, which is a result of the defined block size of 1MB and the average block creation time of 10 minutes, is under discussion. This limitation, together with the high popularity of Bitcoin, has resulted in a dramatic increase in the mem-pool (more than 50 million transactions are currently waiting in the mem-pool[19]) and therefore in a massive delay in transactions.

Another drawback concerns **scalability**. The size of the Bitcoin blockchain reached approximately 154GB in February 2018, and is growing at a rate of 6MB per hour (52GB

[18] Adopted from: http://www.imponderablethings.com/2013/07/how-bitcoin-works-under-hood.html
[19] https://blockchain.info/charts/mempool-size?

per year on average). Although storage capacity is not a real issue today, it is likely to become an issue for small devices, possibly in the IoT arena.

Bitcoin **transaction fees** are also an issue because they depend on the actual Bitcoin market value. This depends on a variety of factors, and some wallets allow users to manually set transaction fees, but the standard fee is about 0.0001BTC/kB. At the current market value of about $8,700 per BTC, the standard transaction fee is about $8 with a very high volatility ($52 as of December 23rd, 2017). For small transaction amounts, this fee is therefore prohibitively high (for more information, see https://bitinfocharts.com/[20]).

At the outset of the Bitcoin network, it was possible to run the proof of work algorithm on the CPUs of nearly all computers. However, with the emergence of GPU and ASIC hardware, it has become increasingly difficult for single mining nodes to stay in the race. At this time, mining pools or groups of cooperating miners who agree to share block rewards in proportion to their contributed mining hash power have come into play. While these are desirable for the average miner, they unfortunately concentrate power within the mining pools. Today there are about 20 major mining pools. Relative to the hash power controlled by a pool and its location, it can be concluded that Chinese pools control approximately 81% of the network hash rate. This de facto **centralization** contradicts Satoshi Nakamoto's original idea of a real decentralized network of cryptocurrency, and poses a threat to the core concepts underlying Bitcoin.

Last but not least, the current **power wastage** of the Bitcoin network has been widely discussed in the media, since all transactions and the mining of blocks (the PoW effort) are done independently by every miner in the world. When a miner has successfully mined a block, it is propagated to all the others, resulting in the current mining process being terminated and all other miners losing the race for rewards, thus wasting all the invested energy. The current energy consumption due to Bitcoin (as of February 2nd, 2018)[21] is about 46.8TWh; this is the equivalent of the energy consumption of a small country such as Denmark.[22] This should, however, be viewed in comparison with the power consumption due to Google searches, Facebook usage and the data centers of the financial industry, which is also known to be extremely high.

3.2 Countermeasures

These issues reviewed above are well known to the Bitcoin community, which is working hard to overcome these shortcomings. The main approaches are described in the Bitcoin Improvement Proposals (BIP) which can be found on Github.[23] Numerous scaling solutions have been proposed for the above drawbacks.

[20] https://bitinfocharts.com/
[21] https://digiconomist.net/bitcoin-energy-consumption
[22] https://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption
[23] https://github.com/bitcoin/bips/blob/master/README.mediawiki.

The first intuitive approach for increasing the tps is to **expand the block size**. The current block size of 1MB was set in 2010 for security reasons, to prevent miners from creating large spam blocks. Since the transaction volume has increased with the widespread usage of Bitcoin, an increase in the limit of 1MB became the subject of vociferous debate in 2015. This debate is still ongoing, and the main focus is on whether such an expansion should support the old version of 1MB (soft fork) or not (hard fork).

Another option that has been discussed is to push the myriad tiny transactions from gambling or crowd-working sites "off-chain". This approach is known as the **Lightning Network**[24], whereby transactions are sent over a network of micropayment channels. This would allow two users to carry out their transactions in a private room, and to then add their data back on the blockchain at an agreed time. However, even this would require a soft fork of the protocol to get started. The most prominent contender to apply a scaling solution based on side chains such as the Lightning Network is Blockstream,[25] which offers paid side chain services.

Miners are compensated for their costs in terms of CPU, network traffic, disk space and memory through fees that are proportional to the size (in bytes) of each transaction. However, each part of a transaction does not have an equal impact on the cost or the ability of Bitcoin to scale to support more transactions. The most expensive parts of a transaction are the newly created outputs, as they are added to the in-memory UTXO set, while the signatures (witness data) add the least load to the network and the cost of running a node, since witness data are only validated once and then never used again. Therefore, one idea is to discriminate between these two types of data relative to the burden a transaction imposes on the in-memory UTXO set. A proposal called **segregated witness (SegWit)**[26] has been put forward which separates transaction signatures from the transactions themselves. SegWit therefore has two main effects: it is able to both increase the number of transactions and to decrease the cost of transactions (fees).

4 The Bitcoin Application Point of View

Although the Bitcoin architecture was designed primarily for monetary transactions, it is also possible to extend it to smart contracts with some restrictions. The Bitcoin scripting language is also not Turing complete, and does not allow any algorithm running it to be extended beyond a payment infrastructure.

Since the Bitcoin architecture serves to keep internal transactions valid, it can also be used to confirm and validate specific, external, non-Bitcoin transactions. In other words, it enables the application of decentralized public ledgers for purposes other than digital currencies. Many companies such as IBM, SAP, Amazon and others are therefore

[24] http://lightning.network/lightning-network-paper-DRAFT-0.5.pdf
[25] https://www.blockstream.com/
[26] https://segwit.org/

experimenting with blockchain applications. These developments may disrupt the way people do business in the future.

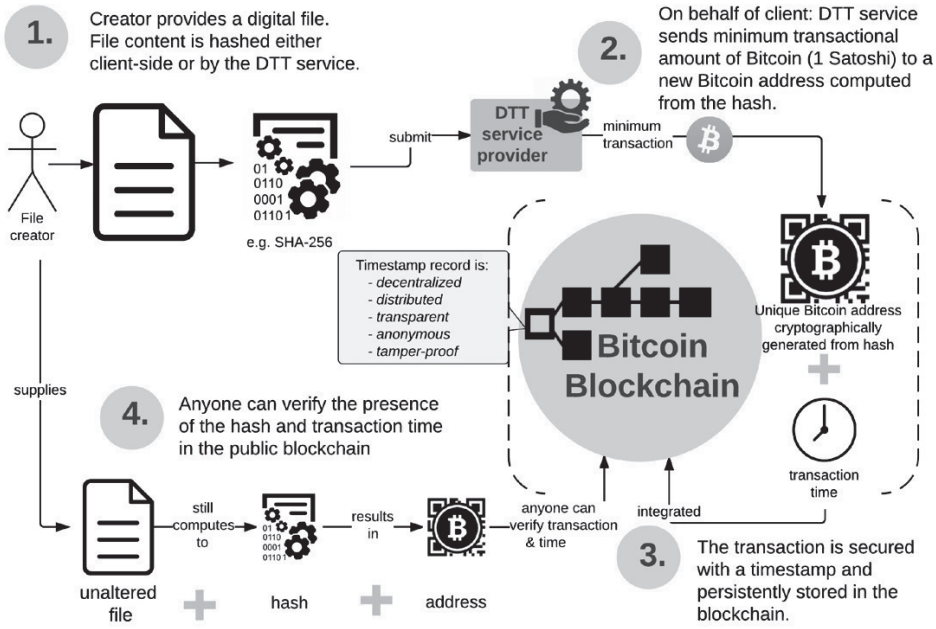


Fig. 11: The timestamping proof (www.originstamp.org)

One remarkable property of the blockchain is its **time stamp feature**. The whole network essentially validates the state of a wrapped piece of data (called a hash) at a certain specific time. The blockchain, hence, essentially confirms the existence of a record at a stated time, which is provable in a court of law. Until now, only centralized notary services could serve this purpose.

This proof of existence allows users to upload a file and pay a transaction fee to have a cryptographic proof (the hash of a document) included on the Bitcoin blockchain, although the file itself is not stored online and therefore does not risk unwanted publication (see Fig. 12). The document is hence timestamped using the block timestamp. Another application concerns **decentralized file storage** on the blockchain, which is likely to disrupt data storage and cloud computing within the next few years. Platforms like Sia[27] or Storj[28] use Blockchain technology to decentralize data storage by dividing up files, encrypting and transferring them all around the world. The accompanying

[27] https://sia.tech/
[28] https://storj.io/

cryptocurrencies (Siacoin, Storjcoin) incentivize usage and to create a market for decentralized storage.

Beside the Bitcoin blockchain there are several other platforms available for building distributed applications based on the blockchain technology. BlockApps, for example, is a good way to build decentralized blockchain-based applications. STRATO, BlockApps's[29] most prominent product, is a full-stack technology solution that allows users to build applications on top of their own customized permissioned private blockchain. This is built on the Ethereum blockchain which can be seen as a second generation blockchain with a built in programming language based on virtual machine concept.

Many of these applications are being currently developed, but the future potential of blockchain applications is still unfolding. The next few years will involve experimenting and applying this technology to all aspects of the economy. Regardless of which application comes first on a global scale. The blockchain is here to stay, and is transforming how business functions [DD17].

5 Conclusion

The blockchain technology first described in 1991 by Haber and Stornetta [HS91], in 1996 by Anderson [An96] and in 1998 by Schneier and Kelsey [SK98] got its sheer popularity through the first implementation of this idea as a distributed ledger called Bitcoin by Satoshi Nakamoto in 2008/09. Based on a simple scripting algorithm, the Bitcoin architecture promises secure and anonymous transactions for transferring money. However, the underlying blockchain technology has a much wider potential than simply transferring money.

As shown in this paper, transactions are first analyzed based on Bitcoin's protocol rules, and then validated using a scripting language based on locking and unlocking scripts. If a transaction is successfully validated, full Bitcoin nodes (miners) will select transactions from the memory-pool, hashing pairs of transaction values until only one hash remains (the Merkle root), and then anchoring the transaction tree into the block header. After choosing a nonce, the complete block header will be hashed and the result compared to a defined target with the PoW mechanism, as described above. If the hash value of the block header is less than the target, the block is mined, the rewards are paid out to the successful miner, and the transaction is included in the blockchain. After at least six blocks, a transaction can be considered irreversible.

Although the security model of Bitcoin promises complete privacy, it is not in general anonymous. As described above, the blockchain is public, meaning that anyone can see every Bitcoin transaction from the genesis block onwards. Bitcoin addresses cannot be

[29] https://blockapps.net/

directly associated to real-world identities though. Nevertheless, a great deal of work has been carried out on how to de-anonymize the Bitcoin network.[30] Furthermore, identification will be retrospective, meaning that someone who carried out a transaction in 2018 will still be identifiable on the basis of the block chain in, say, 2040.

Even if one does not believe in the specific cryptocurrency Bitcoin itself, the underlying blockchain technology will certainly survive and gain in relevance. In summary, Bitcoin, and especially the underlying blockchain system, is a very valuable technology from a scientific computer point of view (be it in computer science or other related fields such as information systems or organization studies), a thread for some businesses, and a gold mine for speculators [FH17].

6 References

[AA17] Antonopoulos, A.M.: Mastering Bitcoin, O'Reilly Media, Inc., 2017.

[AB02] Back, A.: Hashcash - A denial of service counter-measure, 2002.

[An96] Anderson, R.J.: The Eternity Service, Pragocrypt, 1996.

[DD17] Drescher, D.: Blockchain Grundlagen, mitp Verlag, 2017.

[DW13] Decker, C. and Wattenhofer, R.: Information propagation in the Bitcoin network, *IEEE P2P 2013 Proceedings*, 2013.

[FH17] Fraunhofer-Gesellschaft: BLOCKCHAIN UND SMART CONTRACTS Technologien, Forschungsfragen und Anwendungen, 2017.

[HS91] Haber, S. and Stornetta, W.S.: How to Time-Stamp A Digital Document, Journal of Cryptology, Vol.3, No.2, pp.99-111, 1991.

[JB04] Buchmann, J.: Introduction to Cryptography, 2004.

[RM80] Merkle, R.C.: Protocols for public key cryptosystems, In Proc. Symposium on Security and Privacy, IEEE Computer Society, 1980.

[Ro12] Rosenfeld, M.: Analysis of hashrate-based double-spending, arXiv:1402.2009, 2009.

[RW16] Wattenhofer, R.: The Science of Blockchain, Inverted Forest Publishing, 2016.

[SK98] Schneier, B. and Kelsey, J.: Cryptographic Support for Secure Logs on Untrusted Machines, in The Seventh USENIX Security Symposium Proceedings, pp. 53–62. USENIX Press, Januar 1998

[SN08] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[SW05] Steinmetz, R. and Wehrle, K.: Peer-to-Peer Systems and Applications, 2005.

[30] https://scholar.google.com/scholar?q=de-anonymization