Blockchain Technology, Bitcoin, and Ethereum: A Brief Overview

Dejan Vujičić, Dijana Jagodić, Siniša Ranđić
Faculty of Technical Sciences in Čačak
University of Kragujevac
Čačak, Serbia
dejan.vujicic@ftn.kg.ac.rs, dijana.jagodic@ftn.kg.ac.rs, sinisa.randjic@ftn.kg.ac.rs

***Abstract*—The blockchain technology is a relatively new approach in the field of information technologies. As one of its first implementations, bitcoin as a cryptocurrency has gained a lot of attention. Together with Ethereum, blockchain implementation with focus on smart contracts, they represent the very core of modern cryptocurrency development. This paper is meant to give a brief introduction to these topics.**

Keywords – Bitcoin; blockchain; cryptocurrency; Ethereum; smart contracts

I. INTRODUCTION

Bitcoin and blockchain technology have begun to shape and define new aspects in the computer science and information technology. The need for a decentralized money has been exploited more as a theoretical concept, but in the past decade, it became viable, all thanks to the famous paper of Satoshi Nakamoto in 2008, introducing Bitcoin and blockchain technology.

While there are controversies about Nakamoto's true identity, one is for sure: he brought something revolutionary to the world, and it is up to the users to decide what they want to do with it. Some will take this opportunity and develop their own application for solving various problems in the society, others will invest money in those ideas or simply trade with ups and downs of the cryptocurrencies' values at the market.

In this paper, we thought of bringing a small introduction to the matter of blockchain and cryptocurrencies. We begin with a quick retrospective of some of the most famous solutions for decentralized digital money before Bitcoin, and then we go into the very core of its function, together with Ethereum. These two cryptocurrencies hold majority of the cryptocurrency market capitalization. Of course, as it happens with new technologies, some limitations and problems emerged, and we described them as well.

II. SOME EARLY IDEAS ON DECENTRALIZED DIGITAL CURRENCIES

The idea of digital currency is not a relatively new one, but not until recently has it been successfully implemented. In his paper, Chaum presented an idea of untraceable electronic mail, return addresses, and digital pseudonyms, based on public key cryptography [1]. His technique didn't require a trusted authority and the correspondents could be anonymous. Law et al. presented with an idea of electronic cash also with public key cryptography, but their approach was intended for use with banks as central trust authorities [2].

Dwork and Naor [3] proposed a system for usage in combat against junk mail, by demanding the user to provide a computation of a relatively hard pricing function. This was one of the first ideas of providing a proof-of-work as a system for exchanging digital commodities. In similar manners, authors of b-money [4], reusable proof-of-work [5], and bit gold [6] represented ideas of using computational power as an asset with actual and usable value, comparing it with a precious metal or a minted coin [7].

Vishnumurthy et al. proposed a system for secure peer-to-peer (P2P) resource sharing, KARMA [8]. They dealt with the problem of having nodes in P2P networks that use more network resources than they contribute. With each contribution, a node's karma is increased, and with each consummation, it is decreased. A set of nodes is responsible for keeping records of each node's karma.

However, these approaches either required a trusted party in the form of banks or didn't quite solve the double-spending problem. In the centralized solution, banks or other trusted authorities can prevent the attempt of parallel issuance of two transactions, but in decentralized system, as in cryptocurrency, this problem carries great importance [7]. Also, since the central authority doesn't exist, the users have to maintain a consistent state of the P2P network, thus disabling the possible attackers to compromise the system with false data.

One of the possible solutions to these problems was the introduction of quorum systems [9], [10]. In these systems, the possibility of having incorrect information and malicious entities in the network is assumed as true, but the concept of voting is supposed to surpass them [7]. If the majority of nodes in the network is consent about some information, they have the control of the network. However, this approach is prone to Sybil attack [11], where hostile node(s) could manipulate many peers with incorrect information, thus overcoming the election and injecting false information.

978-1-5386-4907-7/18/$31.00 ©2018 IEEE

III. BITCOIN AND BLOCKCHAIN TECHNOLOGY

A. Bitcoin essentials

In his now famous work, Satoshi Nakamoto showed a solution to the problems that the implementation and usability of digital currency faced, especially the double-spending problem [12]. While the true identity of Nakamoto is a point of speculations, what is known is that until 2010 he remained active on the Bitcoin project, and then he stepped back and gave the project to the community for further development [7].

He proposed a system with P2P distributed timestamp server that serves as a generator of the computational proof of the chronological orders of transactions [12]. An electronic coin is defined as a chain of digital signatures. Each transaction is defined as a set of digitally signed hash of the previous transaction and the public key of the next owner. The private key is used for signing the transaction, and the public key is used for verification of the transaction, as shown in Fig.1 [12]. The public key is kept in the wallet, which can be implemented in software, hardware, or online.

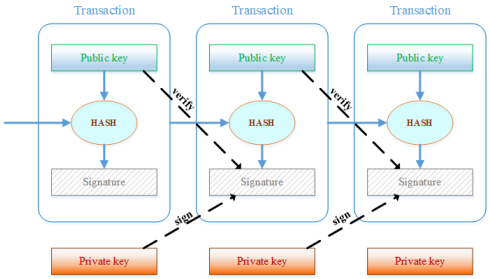

Figure 1. The structure of transaction in a Bitcoin blockchain

The Bitcoin ledger is defined as a state transition system, consisting of a state that shows ownership status of all existing bitcoins and a state transition function, in the form of transaction. The output of the state transition function is a new state [13]. The results of this process are state changes of the sender and recipient if the sender has enough bitcoins to make a transaction, or an error, otherwise.

B. Bitcoin transactions

Each transaction is determined with its hash value representing a transaction identifier and a set of inputs and outputs. Each output of the transaction can only be used once as an input in the entire blockchain [7]. The attempt of referencing the same output twice leads to the double-spending problem and is forbidden in the network. If the output of the transaction hasn't been referenced before, it is called an unspent transaction output (UTXO), and if it has been referenced, it is called a spent transaction output (STXO). A transaction can have multiple inputs and only up to two outputs. Multiple inputs can be used to combine smaller amounts of coins being transferred, and outputs can be either an amount sent to the other party or the change that is sent back to the sender [12].

Bitcoin distributed ledger describes all transactions and ownerships in the network. Every node in this P2P network keeps a copy of the ledger record [13]. If one user wants to send some amount of coins to another, he can do that by publicly announcing this transaction and it is up to the network to verify its correctness. However, a user can try to manipulate the network and issue more than one transaction of the same coin to the different users (double-spending problem). Moreover, the same user can set up several instances to confirm his initial intent and thus perform a Sybil attack.

C. Proof-of-work and blockchain

These situations are prevented (or at least minimized) in Bitcoin network by demanding a proof-of-work from each node that verifies the transaction. The nodes have to do some heavy computations to prove that they are valid members of the network. As long as the total computational power of the honest nodes is greater than the computational power of the attacker, the system will remain consistent and all legit transactions will occur [7], [12].

A set of transactions, together with the hash of the previous block and a nonce, declares a block. A timestamp server makes a hash of a block and publicly announces it, thus proving that the data inside the block must have existed at the time of hashing. The timestamp server has to verify that the timestamp of the block is greater than the timestamp of the previous block in chain and less than two hours into the future. These hashes are linked in a chain and this is called a blockchain, as shown in Fig. 2 [12]. The important property of the blockchain is that the transactions can be traced back at any time in history.

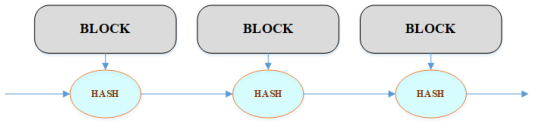

Figure 2. The blockchain scheme

The proof-of-work hashing scheme Bitcoin uses is similar to Hashcash [14] and based on SHA-256 hash function [15]. The proof-of-work is done by incrementing a nonce in the block until the value is produced that has the required number of zero bits at the beginning of the block hash. Once it is done, it cannot be undone without repeating the computations. If it is somehow changed by a malicious attacker, then all the following blocks would have invalid hashes. The rule is that the longest chain that has the majority consensus in the network is the correct one, so if the attacker wishes to change a block, he needs to have enough computational power to overcome the voting of the majority of honest nodes, thus entering the race problem.

The transactions within a block are hashed in a Merkle tree [16], [17]. A Merkle tree is a type of binary tree with many leaf nodes, and a root of the leaf nodes is a hash of its children. The Fig. 4 shows a Bitcoin block consisted of a Merkle tree of transaction hashes. Any inconsistency in the tree will reflect somewhere in the chain, so the Merkle tree is vital for long-term maintainability [13]. This is done to free up the storage space needed to store the blockchain on the nodes. The current size of the Bitcoin blockchain is about 144.8 GB [18]. After the transactions are incorporated in a block and this block is verified, the network discards all hashes in a tree except the root hash included in the block header. Bitcoin introduced a Simplified Payment Verification (SPV), which doesn't require the nodes to keep a full record of transactions, but only the copy of the block headers of the longest chain [12].

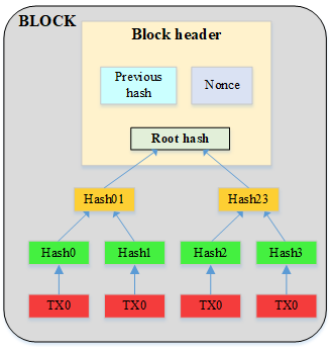

Figure 3. A Bitcoin block with hashed transactions into a Merkle tree

D. *Bitcoin network and mining*

The first transaction in a block creates a new coin which is owned by the creator of the block [12]. This gives stimulus to nodes to verify transactions, and puts coins into circulation, since there is not a central authority that issues them. This transaction is called a coinbase transaction. With this approach, the nodes are incentive to stay honest. The Bitcoin network is intended to produce one block in approximately ten minutes [13]. Since the computational power increases in time, the block time is remained somewhat constant by gradually increasing the difficulty of generating new blocks.

The Bitcoin network starts with new transactions being broadcasted to all nodes. Each node gathers transactions into a block and works on finding proof-of-work, after which it broadcast its block to the network. The nodes in the network accept the block as valid only if all transactions within it are correct and not already spent. If the block is accepted by the network, the chain is being continued by creating the next block and adding the hash of the previously added block to it [12].

Beside the reward based on the block creation, the nodes are rewarded with coins and by verifying transactions. The process of adding new blocks to the blockchain is called mining [7]. The initial block reward was set to 50 coins (50 BTC) and was intended to be gradually split in half with every 210,000 blocks. The first block in the blockchain is the genesis block and is used to supply the initial 50 BTC to the network. The halving of the block creation reward will continue until the reward drops below one satoshi, which is the minimal unit of Bitcoin and is equal to 10^{-8} BTC [7].

Taking the nature of distributed decentralized systems in account, there are situations where several nodes almost simultaneously broadcast the same block, but with probably different set of transactions. This situation is known as fork and leads to the inconsistent state of the network. Basically, there are several chains that originated from different blocks. This situation is resolved in the way that the network always continues with the longest chain. Gradually, there will be a consensus within the network about the correct path of the blockchain, and the chains generated as a result of a fork will be invalid [7].

E. *Bitcoin scalability problem*

With a block size of 1MB, Bitcoin has severe scalability issues. The amount of transactions that can be supported with this block size is less than seven transactions per second (tps) [19]. In comparison, the payment network Visa achieved 47,000 tps during the 2013 holidays, and currently averages with hundreds of millions per day [20]. To achieve such rate on Bitcoin network with 1MB block size, assuming that the transaction is 300 bytes in size, it would require a throughput of 8GB per Bitcoin block every ten minutes, which would lead to over 400TB of data per year [19]. This would highly centralize the Bitcoin network to support only those nodes with such storage capacities, and this is the very opposite of what Bitcoin and blockchain are intended for.

Several solutions were suggested in order to tackle this issue efficiently. As a result, number of soft and hard forks of Bitcoin occurred. A soft fork is any change that is backward compatible, i.e. enabling the old software to recognize newly created blocks as valid. A hard fork, on the other hand, is a software update introducing a new rule to the network, thus rendering the old software unable to recognize new blocks [21].

F. *SegWit and Lightning*

SegWit (Segregated Witness) is one of the proposed solutions to the Bitcoin scalability issue, dealing with the problem of transaction malleability. This problem is caused by the fact that the transaction signature doesn't cover all the data in the transaction, so it is possible for some malicious node on the network to change a transaction and invalidate its hash [22].

SegWit enables increasing the block size to a maximum of 4MB and adding the second layer on top of the existing network [23]. It separates the signature data from other transaction data and facilitates the introduction of the Lightning network as a second-layer protocol. It was activated on the 24th of August 2017 on the block 481,824 [24].

The Lightning network is "the next big thing" on the Bitcoin network. It is supposed to facilitate micropayments by using a network of micropayment channels. A micropayment channel is a consensus between two parties to delay with announcing of the transaction to the network, while actually making the transaction. The both parties can guarantee their current balance on the blockchain, but they choose to defer sending information about transaction to the network [19].

The successful trial of the Lightning network was conducted by Blockstream in January 2018 [25].

G. *Bitcoin Cash and Bitcoin Gold*

With Bitcoin becoming more and more popular, the network was unable to handle so many transactions, so the confirmations took even days [26]. The hard fork of Bitcoin Cash (BCH) occurred on the block 478,558 on the 1st of August, 2017. The owners of Bitcoin at the time of the fork became also the owners of the Bitcoin Cash. The block size was upgraded to 8MB, leading to faster confirmation times.

Bitcoin Gold (BTG) is a hard fork of Bitcoin which happened on the block 491,407, on the 24th of October, 2017 [27]. Like with Bitcoin Cash, any holders of Bitcoin before the fork became also the holders of Bitcoin Gold. The reason behind this fork is that instead of CPUs, the Bitcoin was primarily mined with ASIC machines. The creators of Bitcoin Gold sought to change that by changing the proof-of-work algorithm from

SHA-256 to Equihash [28]. This algorithm is memory heavy and mostly suitable for mining on graphic processors. The mining difficulty is adjusted for every new block created, instead of adjusting after every 2016 blocks in Bitcoin [27].

IV. ETHEREUM

A. Overcoming Bitcoin's limitations

Ethereum was introduced in Vitalik Buterin's paper [29] and addressed several limitations of the Bitcoin's scripting language. The main contributions are full Turing-completeness, meaning that Ethereum supports all types of computations, including loops. Then Ethereum supports the state of the transaction, as well as several other improvements over the blockchain structure.

Ethereum represents a blockchain with a built-in Turing-complete programming language. It provides an abstract layer enabling anyone to create their own rules for ownership, formats of transactions, and state transition functions. This is done by involving smart contracts, a set of cryptographic rules that are executed only if certain conditions are met [29].

The consensus in the Ethereum network is based on modified GHOST protocol (Greedy Heaviest Observed Subtree) [30]. It is created to tackle the issue of stale blocks in the network. The stale blocks can occur if one group of miners combined in a mining pool has more computing power than the others, meaning that the blocks from the first pool will contribute more to the network, thus creating the centralization issue. GHOST protocol includes those stale blocks into calculations of the longest chain.

The centralization problem is removed through providing block rewards to stales, where the stale block receives 87.5% of the reward, and the nephew of that stale block receives the remaining 12.5% of the reward. In this way, the miners are still rewarded even if their block didn't become the part of the main blockchain (those blocks are called uncles). Ethereum uses the modification of the GHOST protocol which includes uncles up to seven generations [13].

B. Ethereum accounts

The Ethereum state is consisted of accounts, where each account has a 20-byte address and state transitions. The world state is a mapping between addresses and account states [31].

Ethereum supports two types of accounts: externally owned (controlled by private keys) and contract accounts (controlled by their contract code) [13]. An Ethereum account is made of four fields: nonce, ether balance, contract code hash, and storage root [30], [31].

Nonce represents the number of transactions sent from particular address or the number of contract creations made by an account and is used as a guarantee that each transaction can only be processed once. Ether balance is the number of Wei owned by this address (Wei represents the smallest fraction of Ether, one Ether – ETH, Đ, being equal to 10^{18} Wei). Ether is used for paying transaction fees. Contract code hash is the Keccak-256 hash of Ethereum Virtual Machine (EVM) code of the account, which is executed if an address receives a message call. Storage root is the 256-bit hash of the root node of a Merkle Patricia tree that represents the content of the account [31]. Merkle Patricia trees (tries) are used for storage of all (key, value) bindings in Ethereum ecosystem. The block header contains three roots from three tries representing state, transactions, and receipts [32].

C. Ethereum transactions and messages

A transaction is a single instruction that is cryptographically signed. There are two types of transactions based on their products (ones that result in message calls and ones that create new accounts). The transaction is defined as a signed data package dispatched from an externally owned account. Each transaction is consisted of the recipient of the message, a signature identifying the sender, amount of Ether to be sent, an optional data field, STARTGAS, and GASPRICE values [13], [31].

STARTGAS and GASPRICE fields are vital in the combat with attackers on the network. "Gas" is a fundamental unit of computation. Each transaction requires certain amount of computations, and the STARTGAS field denotes the maximum number of computational steps the transaction is allowed to consume. Usual price is 1 gas per 1 computational step plus fixed additional price of 5 gas for every byte in the data area, but this value can be greater and is defined in GASPRICE field [13].

Since the miners are rewarded more if they process the transaction with higher GASPRICE, the sender has to choose carefully the GASPRICE value if he wants his transaction to be processed. On the other hand, miners also have to acknowledge some minimal GASPRICE under which they refuse to process a transaction [31].

The Ethereum state transition function, which changes states of the sender and the recipient by executing a transaction, starts with verifying the correctness of the transaction (the signature is valid and the nonce matches the nonce in the sender's account). If this is correct, then it calculates the transaction fee as STARTGAS * GASPRICE, subtracts this value from sender's account balance, and increments his nonce. If this is correct, the fee is paid per byte in the transaction and the requested amount of Ether is transferred to the recipient. The receiving account is created if it doesn't already exist, and if it is a contract, then the contract's code is executed. If the sender doesn't have enough amount of Ether for transaction or the code execution spend all the gas, the state transition function reverts all state changes except the payment fees for the miners [13].

One contract can send a message to the other in the Ethereum network. The message resembles the transaction, but is produced by a contract. Same as with transactions, the message induces that the recipient account runs its code.

D. Ethereum blockchain

An Ethereum blockchain is similar to the Bitcoin blockchain. The main difference is that Ethereum blocks contain not only the block number, difficulty, nonce, etc. but also the transaction list and the most recent state. For every transaction in the transaction list, the new state is created by applying the previous state.

The block header in the Ethereum blockchain consists of the Keccak 256-bit hash of the parent block's header, the address of the mining fee recipient, hashes of the roots of state, transaction,

and receipts tries, the difficulty, the current gas limit of the block, a number representing total gas used in the block transactions, timestamp, nonce, and several extra hashes for verification purposes [31].

One of the biggest problems of the Bitcoin's network is the eligibility for ASIC mining. Ethereum uses Ethash as the proof-of-work algorithm which is memory heavy and thus less suitable for ASIC mining. Ethash represents the modification of the Dagger-Hashimoto algorithm [33], [34].

Every node in the Ethereum network runs under EVM and executes its instructions. The smart contracts are translated into EVM code and then executed by the nodes [31]. One of the most popular programming language for writing smart contracts is Solidity.

Ethereum block time is around 15 seconds, with several peaks up to 30 seconds for a period of time. Using Geth blockchain client with fast sync, the Ethereum blockchain size is 47.43GB as of the 29th of January 2018 [35].

Although there are concerns about Ethereum scalability, it was recorded that the Ethereum network successfully managed over one million unique transactions in 24 hours, averaging at around 11 transactions per second [36]. "Serenity" prototype of Ethereum platform based on the Casper consensus algorithm, intended for later implementation, is supposed to enable the transition to the proof-of-stake mining paradigm, where the reward is given to the miners not based on their computations, but on their coin holdings (the more coins the user possesses, the bigger reward gains) [37].

The potential usages of Ethereum are described as token systems, financial derivatives, identity and reputation systems, file storage, insurance, cloud computing, prediction markets, etc. [29]. The most important use case of Ethereum are decentralized apps (Dapps). Some of them are Golem (supercomputing), Augur (prediction markets), Civic (identity verification and protection), OmiseGO (exchanges on a public blockchain), Storj (renting the hard drive space), and many more who succeeded in raising enough money through ICOs (Initial Coin Offerings) to be represented on the cryptocurrency market [38].

E. Ethereum Classic

The DAO (Decentralized Autonomous Organization) was the venture capital fund with no management structure, who aimed to raise the money for Ethereum Dapps that were promising, by their own belief. The investments were distributed via DAO tokens.

The DAO succeeded to raise over $150 million in Ether from more than 11,000 investors [38]. However, it was hacked for $50 million and this caused a disagreement in the Ethereum community should the investors get their Ether balances back. This lead to a hard fork and on the block 1,920,000, on the 20th of July 2016, the Ethereum Classic (ETC) was born. The hard fork continued under the old name Ethereum (ETH), while the original Ethereum before the hard fork was renamed to ETC [38].

Some of the announced Ethereum forks to come are Expanse [39], Ethereum Fog [40], and Ethereum Zero [41].

F. ERC20 Token

Ethereum as a platform is suitable for the issuance of tokens. Ethereum based tokens are smart contracts that implement the ERC20 Token Standard. This standard defines a contract that has to be implemented and is consisted of 6 functions and 2 events that fully describe an account [42]:

```
contract ERC20Interface {
    function totalSupply() public constant returns (uint);
    function balanceOf(address tokenOwner) public constant returns (uint balance);
    function allowance(address tokenOwner, address spender) public constant returns (uint remaining);
    function transfer(address to, uint tokens) public returns (bool success);
    function approve(address spender, uint tokens) public returns (bool success);
    function transferFrom(address from, address to, uint tokens) public returns (bool success);
    event Transfer(address indexed from, address indexed to, uint tokens);
    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);
}
```

V. CONCLUSION

Bitcoin and Ethereum today are the most known and valuable cryptocurrencies. They are based on blockchain technology that is intended to promote a trust mechanism in a peer-to-peer network based on the consensus of the majority of the nodes. We have shown in this paper a short chronological survey of the early stages of the digital money implementation, as well as the foundations of blockchain technology, and it's most promising (or popular) implementations, Bitcoin and Ethereum.

In the past few years, there has been a rapid growth of numerous cryptocurrencies, hashing algorithms, and consensus agreements in the networks. Some of the cryptocurrencies worth mentioning are Ripple, Cardano, NEO, Stellar, Litecoin, EOS, IOTA, Dash, Monero, TRON, Qtum, Lisk, Tether, Stratis, Zcash, Steem, Siacoin, Verge, Electroneum, Nxt, Dogecoin, and many more. The full list can be found at [43], and there are total of 1,498 cryptocurrencies listed with 8,250 markets, with total market capitalization of $556.471.064.589, as of the 30th of January 2018, 03:00 GMT.

ACKNOWLEDGEMENT

The work presented in this paper was funded by grant TR32043 for the period 2011-2018, by the Ministry of Education, Science, and Technological Development of the Republic of Serbia.

REFERENCE

[1] D. Chaum, "Untraceable electronic main, return addresses, and digital pseudonyms," in Communications of the ACM, vol. 24, no. 2, pp. 84-88, February 1981

[2] L. Law, S. Sabett, and J. Solinas, "How to make a mint: the cryptography of anonymous electronic cash," American University Law Review, vol. 46, no. 4, pp. 1131-1162, 1996

[3] C. Dwork and M. Naor, "Pricing via processing or combatting junk mail," in 12th Annual International Cryptology Conference, pp. 139-147, 1992

[4] W. Dai, "B-money," 1998, available at: http://www:weidai:com/bmoney

[5] H. Finney, "RPOW," 2004, available at: http://nakamotoinstitute.org/finney/rpow/

[6] N. Szabo, "Bit Gold," 2005, available at: http://unenumerated.blogspot.rs/2005/12/bit-gold.html

[7] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2084-2123, March 2016

[8] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "KARMA: A secure economic framework for peer-to-peer resourse sharing," 1st Workshop on Economics of Peer-To-Peer Systems, 2003

[9] N. Szabo, "Secure property titles with owner authority," 1998, available at: http://nakamotoinstitute.org/secure-property-titles/

[10] D. Malkhi and M. Reiter, "Byzantine quorum systems," Distributed Computing, vol. 11, no. 4, pp. 203-213, 1998

[11] J. Douceur, "The Sybil attack," in Proceedings of IPTPS '01 Revised Papers from the First International Workshop on Peer-to-Peer Systems, pp. 251-260, March 2002

[12] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2008, available at: https://bitcoin.org/bitcoin.pdf

[13] Ethereum Community, "A next-generation smart contract and decentralized application platform," White Paper, available at: https://github.com/ethereum/wiki/wiki/White-Paper

[14] A. Back, "Hashcash – a denial of service counter-measure," 2002, available at: http://www.hashcash.org/papers/hashcash.pdf

[15] D. Eastlake, 3rd and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)," RFC 6234 (Informational), May 2011, available at: http://www.ietf.org/rfc/rfc6234.txt

[16] R. Merkle, "A digital signature based on a conventional encryption function," In: Pomerance C. (eds) Advances in Cryptology — CRYPTO '87. CRYPTO 1987. Lecture Notes in Computer Science, vol 293. Springer, Berlin, Heidelberg, pp. 369-378, 1987

[17] R. Merkle, "Protocols for public key cryptosystems," in Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pp. 122-133, April 1980

[18] Bitcoin Blockchain Size, https://charts.bitcoin.com/chart/blockchain-size

[19] J. Poon and T. Dryja, "The bitcoin lightning netwiork: scalable off-chain instant payments,", 2016, available at: https://lightning.network/lightning-network-paper.pdf

[20] M. Trillo, "Stress test prepares VisaNet for the most wonderful time of the year," 2013, available at: https://www.visa.com/blogarchives/us/2013/10/10/stress-test-prepares-visanet-for-the-most-wonderful-time-of-the-year/index.html

[21] A. Castor, "A short guide to Bitcoin forks," March 2017, available at: https://www.coindesk.com/short-guide-bitcoin-forks-explained/

[22] Transaction malleability, available at: https://en.bitcoin.it/wiki/Transaction_malleability

[23] Cointelegraph, SegWit explained, available at: https://cointelegraph.com/explained/segwit-explained

[24] A. Hertig, "SegWit goes live: why Bitcoin's big upgrade is a blockchain game-changer," August 2017, available at: https://www.coindesk.com/50-blocks-segwit-bitcoins-coming-upgrade-blockchain-game-changer/

[25] S. Sundararajan, "Blockstream launches micropayments processing system for Bitcoin apps," January 2018, available at: https://www.coindesk.com/blockstream-launches-micropayments-processing-system-for-bitcoin-apps/

[26] Bitcoin Cash, https://www.bitcoincash.org

[27] Bitcoin Gold Roadmap, available at: https://bitcoingold.org/wp-content/uploads/2017/10/BitcoinGold-Roadmap.pdf

[28] A. Biryukov, D. Khovratovich, "Equihash: asymmetric proof-of-work based on the generalized birthday problem," Ledger, vol. 2, pp. 1-30, April 2017

[29] V. Buterin, "Ethereum white paper: a next generation smart contract & decentralized application platform," 2013, available at: http://www.the-blockchain.com/docs/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

[30] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in Bitcoin," Financial Cryptography, pp. 507-527, 2015

[31] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger, Byzantium version," 2018, available at: https://ethereum.github.io/yellowpaper/paper.pdf

[32] Merkle Patricia Trie Specification (also Merkle Patricia Tree), available at: https://github.com/ethereum/wiki/wiki/Patricia-Tree

[33] V. Buterin, "Dagger: A Memory-Hard to Compute, Memory-Easy to Verify Scrypt Alternative," 2013, available at: http://www.hashcash.org/papers/dagger.html

[34] T. Dryja, "Hashimoto: I/O bound proof of work," 2014

[35] Etherscan, https://etherscan.io

[36] J. Filiba, "Ethereum Breaks One Million Transactions in a Single Day," December 2017, available at: https://discover.coinsquare.io/tech/ethereum-one-million-transaction-day/

[37] L. Silva, "Ethereum's road map for 2017," available at: https://www.ethnews.com/ethereums-road-map-for-2017

[38] A. Moskov, "What is Ethereum?", 2017, available at: https://coincentral.com/what-is-ethereum/

[39] M. Warner, "Expanse DAO Seeks to Develop Decentralised Ethereum," November 2017, available at: http://allcoinsnews.com/2017/11/09/expanse-dao-seeks-to-develop-decentralised-ethereum/

[40] JP Buntinx, "What is Ethereum Fog?," December 2017, available at: https://themerkle.com/what-is-ethereum-fog/

[41] EtherZero — a Revolutionary Ethereum Hard Fork, Forking on 19th Jan 2018, available at: https://news.bitcoin.com/pr-etherzero-a-revolutionary-ethereum-hard-fork-forking-on-19th-jan-2018/

[42] ERC20 Token Standard, available at: https://theethereum.wiki/w/index.php/ERC20_Token_Standard

[43] Cryptocurrency Market Capitalizations, https://coinmarketcap.com/