Mathematics and the algorithms that guarantee the Bitcoin security - an internal seminar paper

Filip Makraduli, 2017

Abstract - Bitcoin is an electronic currency that works on a peer-to-peer basis and enables transactions to be made directly from one party to another without the use of a mediator in the form of a financial institution. This seminar paper provides an overview of how the technology of the public ledger (record book of transactions), popularly called blockchain, works. Digital signatures are part of the solution, but also it is necessary to protect the system from the possibility of creating fake transaction blocks. This revolves around the concept of proof-of-work, which provides evidence of whether a certain block is true based on how large the computing power, used to hash the specific block from the blockchain, is. The background mathematics and algorithms of this technology that guarantees the safety of the public book, transactions, and transaction blocks are explained. In the first part, the security of transactions achieved using digital signatures and cryptography of elliptic curves with the secp256k1 curve is analyzed. In the second part, the Secure Hash Algorithm (SHA256) is explained. The use of binary SHA 256 hashing in the form of Merkle trees are also discussed. This algorithm ensures the reliability of blocks in the ledger and protects the system from conducting fraudulent transactions.

Keywords - components; blockchain; Bitcoin; proof-of-work; SHA256; elliptical curves; Merkle trees; secp256k1; ledger.

I. INTRODUCTION TO THE FIRST FUNCTIONAL CRYPTOCURRENCY- BITCOIN

The idea of Bitcoin was born as an act of revolt. In 2009, an anonymous hacker or a group of hackers known under the pseudonym Satoshi Nakamoto created the first fully functional digital cryptocurrency. This idea is explained in the document published as a scientific paper with the original name of “The Bitcoin whitepaper”. This scientific paper, together with the original Bitcoin code, is the first and one of the main sources of information in this seminar paper. [1]

Bitcoin creation happens at the right time to declare the Bitcoin system as an opponent of the traditional financial system. After the collapse of the world stock exchange as a consequence of the economic crisis in 2008, the concept of cryptocurrencies was initially promoted as an opponent precisely to those responsible for the events of 2008 [2] bankers, intermediaries, and financial institutions. This revolt claimed that they were the ones at fault for the developments during the global economic crisis. Bitcoin seeks to replace all services provided by these mediation institutions with cryptography and mathematics translated into code. In the conventional banking system, there are a number of authentication processes i.e. confirmation of the truth between the banks even for one of the most common monetary transactions. [2]

Bitcoin and other cryptocurrencies are replacing this first step with the help of software, more specifically a technology of a distributed database called a blockchain. This database is actually a ledger and plays a central role in the way Bitcoin works. This technology creates an opportunity for a decentralized system, which does not depend on anyone, bank or government, but on a number of computers that have a role as system operators. [3]

What has been suggested by Satoshi Nakamoto is an electronic payment system based on cryptographic mathematical proof instead of "honesty" which is the main guarantor in traditional financial transactions. Such a proposal allows two parties to execute transactions eliminating the need for a third party. Satoshi's solution guarantees security against fake transactions as long as the "honest" blocks in the system (those who are not trying to create fake transactions) control more computing power than those who attempt to attack the blockchain. The genius behind this idea is that besides being limited by the availability of computing power, the attacker additionally has an economic incentive to work in favor of the system. In simple terms, even if the attacker has a lot of computing power available, it is more profitable to use that computing power in favor of the system rather than against it. The ethical behavior when participating in the system is supported by the potential for greater profits. [1]

II. WAY OF OPERATION

1. General overview

Bitcoin technology is based on the assumption that money, reduced to the lowest fundamental level, represents an accounting tool. It is a method of value abstraction, assignment of property, and execution of a transaction. Bitcoin managed to create a unique, universally available ledger called a blockchain. Like the English name itself says, it is a chain of blocks. Changes can only be made by adding new blocks at the end of the chain. Each block contains a set of transactions that contain references to older transactions and blocks. [1] [2], Unlike traditional transaction books, the Bitcoin blockchain has been replicated on a network of multiple computers as a kind of server, and the Bitcoin blockchain is available to anyone with an internet connection and a computer.

A certain group of people (due to economic incentive and financial reward) decide to be a system operator of Bitcoin, that is, to invest their available computing power to ensure proper operation of the system. These people are popularly called miners and are responsible for detecting transaction requests from users, joint collection of transactions in a pile, validation of transactions, and their addition to the end of the chain as new blocks. [3] The validation process consists of two steps. The first step is a confirmation that the user really owns a certain amount of Bitcoin, and the second step is a confirmation that the user has not already spent them, i.e. exclusion of the risk of double-spending. The ownership of Bitcoin is determined using cryptographic keys i.e. asymmetric encryption. There are two types of cryptographic keys, a public key on the blockchain that is made available to everyone via a publicly accessible repository or directory, and a private key that must remain confidential to its respective owner. [2] [4] In Figs. 1. the layout of the blockchain with the elements mentioned so far are shown.

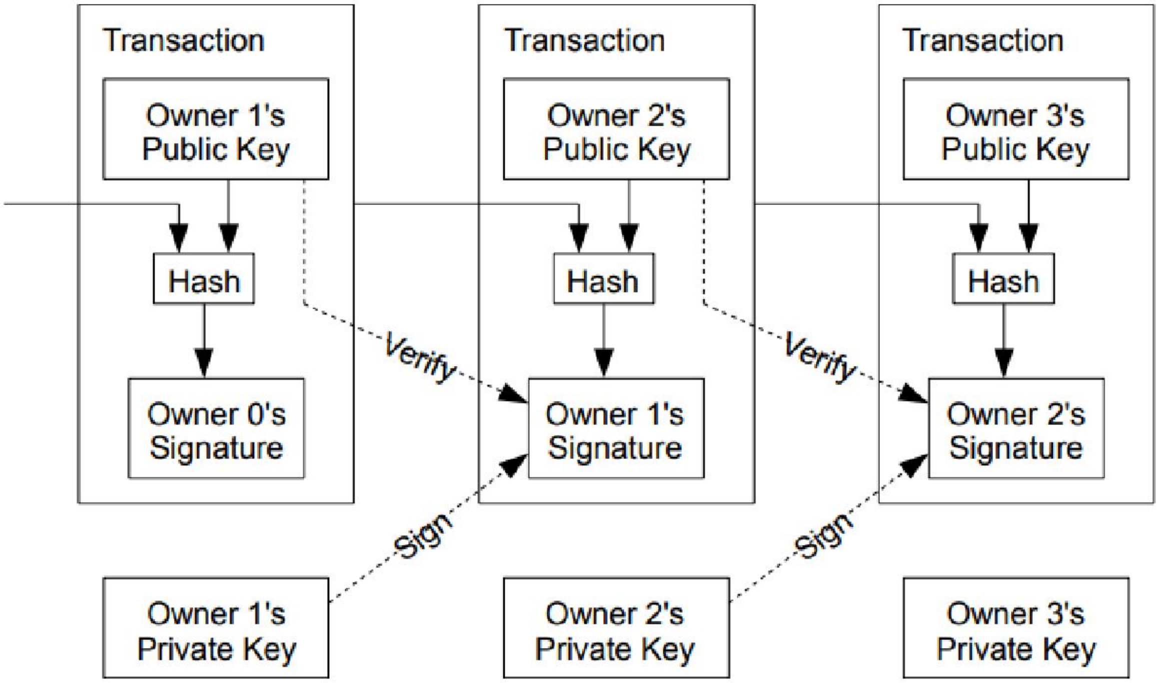

Fig. 1. The layout of the structure of the blocks [2]

The keys have specific mathematical relationships that make them useful for signing messages. The only way to avoid the absence of a transaction in this already mentioned system is to be aware i.e. to have some information about all transactions in the system. To achieve this without a third party that would act as a supervisor, it is required for the participants in the system to agree on a single ledger. Any user who wants to make some payment needs proof of authenticity that is proof that the block to which the transaction will be attached is a part of the true longest chain of blocks. Only then the user will be sure that all transactions from the beginning of the system history to the present have been taken into account, and only then the user will be sure that there is no fraudulent or hidden transaction in the system. Satoshi implements this concept by timestamping of blocks. The time marking confirms that the given block existed at the time defined with the marking. A timestamp server is hashing every transaction in one block and announces every transaction publicly, for example as a publication in a newspaper. [1] [2] Each hashed time marker in its hash contains information about the hash of the previous block. This is how a chain of blocks is formed where each block has information about all the blocks that existed before. This way, each existing block is connected through its hash to the first block, known as the block of genesis. Graphically this chain is shown in Fig. 2. It is important to mention the concept of Merkle trees (which will be further explained in this paper) which comes into play when it is necessary to save on memory space when hashing, i.e. to shorten the hash data size, but not lose hashable information

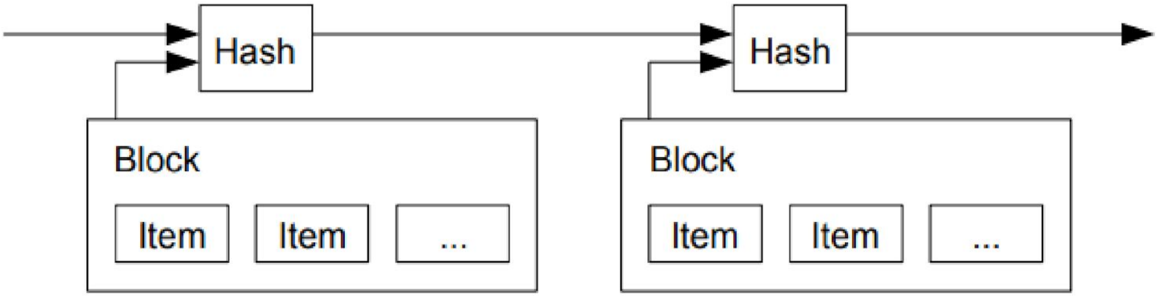

Fig. 2. Time mapping of blocks in the blockchain [2]

The main role of miners is to ensure the irreversibility of new transactions, making them final, permanent, and without the possibility of their falsification or cloning. The way this is achieved by Satoshi (together with the public and private key encryption), is one of his, her or their most significant contribution to computer science as a whole. [1] [2] [3] Achievement of the irreversibility of transactions is necessary for the system in which anyone and everyone can participate. There is no central authority that will record all transactions in the public ledger. There is no central bank with its policy to punish those who do not follow the rules. There is no institution in Bitcoin that will be the "executive power". There are Miners from all parts of the world, from China to Iceland, Venezuela, Ukraine, and Libya. There are different political and governmental systems with huge variants in judicial systems. The code of Bitcoin is the only one that can provide appropriate user behavior. To achieve this, Satoshi implements the scheme of Proof-of-work. [3]

2. Proof-of-work

To implement a distributed server of time marking on a point-to-point system, Satoshi uses a proof-of-work system. This proof is needed because the Bitcoin miners compete between themselves. The miners, as mentioned earlier, invest their available computing power to sort the transactions and to collect them into blocks. For this, they receive a reward (also miners receive a small percentage of the transactions they approve called taxes for mining or miner fees), because the first to succeed in making a valid block receives a reward. What prevents a certain user to make his or her own version of the blockchain and to add only the blocks that suit him or her, that is, the blocks in which that user does not spend his or her Bitcoins? The answer is obvious, that is the “proof-of-work” concept. This concept consists of scanning with an aim of finding a value that would be hashed (in this case with SHA 256). The hash would start with a certain earlier defined number of bits with a value zero. On average the required work is exponential to the number of zero bits and can be verified by simply executing the single hash. At the moment when the processing power is used to satisfy the condition defined by the proof-of-work, the block cannot be changed without doing the necessary work before (in the context of the computing processing power) which was made for generating the block. As more blocks are added in the chain, the work needed to change some of the blocks involves re-doing the work for all blocks after the specific block to which there is an attempt to make a change. Such a blockchain with interconnected hashes is shown graphically in fig. 3.

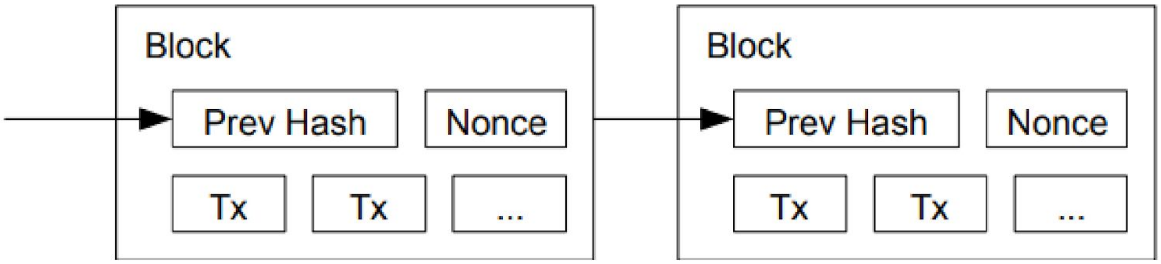

Fig. 3. Proof-of-work blockchain [2]

This concept can also be defined as a mathematically correct, practical form of representative democracy where the majority makes the next decision. One "voice" is one unit of computing power. The decision of the majority is embodied through the longest chain of blocks. To modify a particular block from the chain, the potential "attacker" of the system should re-perform the work required to obtain proof-of-work on the block, but also on all blocks after that concrete block, and should do that faster than all the "honest" users in the system. Later in this paper, it will be mathematically shown that the probability of this to happen is almost impossible. [3]

To compensate for the increased hardware speed and the growing number of users, the weight of the proof-of-work (the number of zero bits in the solution of a cryptographic hash function) varies over time and is determined according to the mathematical relation where a certain average number of blocks per hour is targeted. If blocks are generated very quickly, the weight increases. A maximum limit of Bitcoins in the system is also defined, and the reward for miners decreases with increasing the weight of proof-of-work. In the beginning, the reward for the miners was set at 50 Bitcoins, and now (2017) it has been reduced to 12.5 Bitcoins. The maximum number of Bitcoins that can exist in the system is 21,000,000, and it is assumed that this limit will be reached in 2140. [3] At this moment the system would work so that the miners would generate profit only from the transaction tax.

3. Merkle trees

In the real application of blockchain technology at Bitcoins, a possibility to save memory at the execution of transactions without loss of information exists. When the most recent transaction is below a certain number of blocks (a bunch of multiple transactions), all previous transactions are discarded in order not to take up memory space. This is achieved by hashing the transactions into Merkle trees, also called hash trees [6]. The Merkle root is a special hash of all transactions included in the block and it is derived from the "leaves" of the Merkle tree. In fig. 4 this form of hashing is explained. "H" denotes hash, and the letters in the Latin alphabet (A, B, C...) denote each transaction respectively. The transactions (A, B, C ...) make the "leaves" of the Merkle tree. The root or the root hash to be found, the transactions to be included in a new block are hashed, divided into pairs and each pair is hashed using a binary SHA 256 function (cryptographic hash functions will be further explained in the paper). [8] The hash result of each pair is then put in a pair and hashed, and so on until one hash of 256 bits remains. That one hash is called the Merkle root.

The purpose of this hashing is to create a unique identifier. When this hashing is done for all transactions, any attempt to change any information included in transactions hashed through Merkle trees will cause the root to change completely. This is a huge level of security inside the block. It can be seen if a certain transaction is a part of a specific block by showing whether it is included in the Merkle tree for that block. To do this, miners retrieve information about other transactions included in the block, hash the specified transaction together with the rest of the block and compare the resulting Merkle root with the root of the specific block.

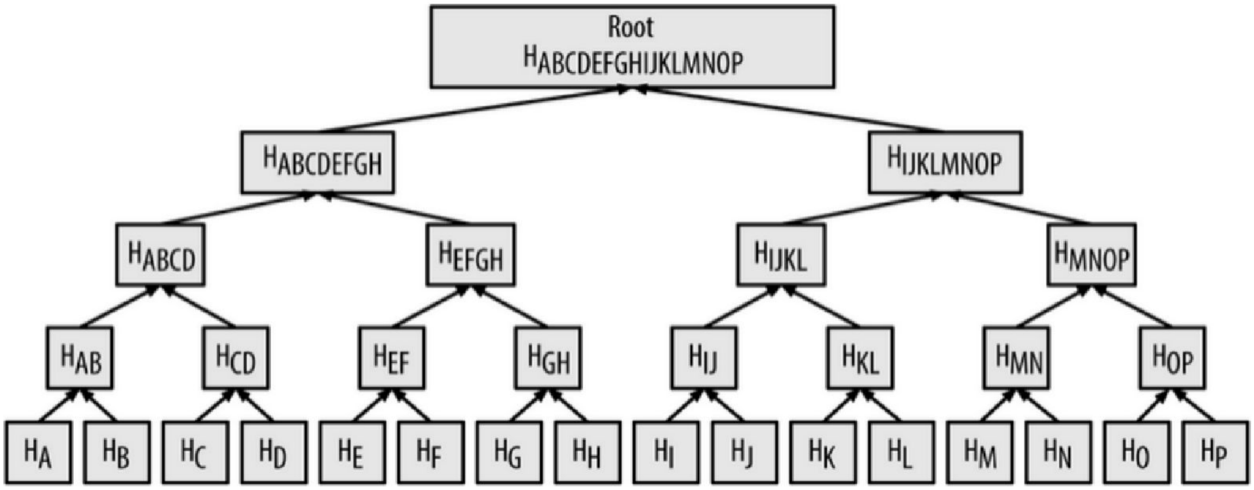

Fig. 4. The Merkle tree and its root [7]

In the example from Fig. 4, to verify that transaction J is included in this block, the hashes ABCDEFGH, MNOP, and KL are taken (hashes that do not contain the letter J) from other users. Then the rest of the branches of the Merkle tree are completed and that hash is compared to the real Merkle root from the block. [6] [7]

Only the root is included in the hash of the block. In this way, unnecessary branches from older blocks are discarded and the inner branches of the hash do not need to be saved. In fig. 5 the final form of the hash in Bitcoin blockchain blocks is graphically clearly shown.

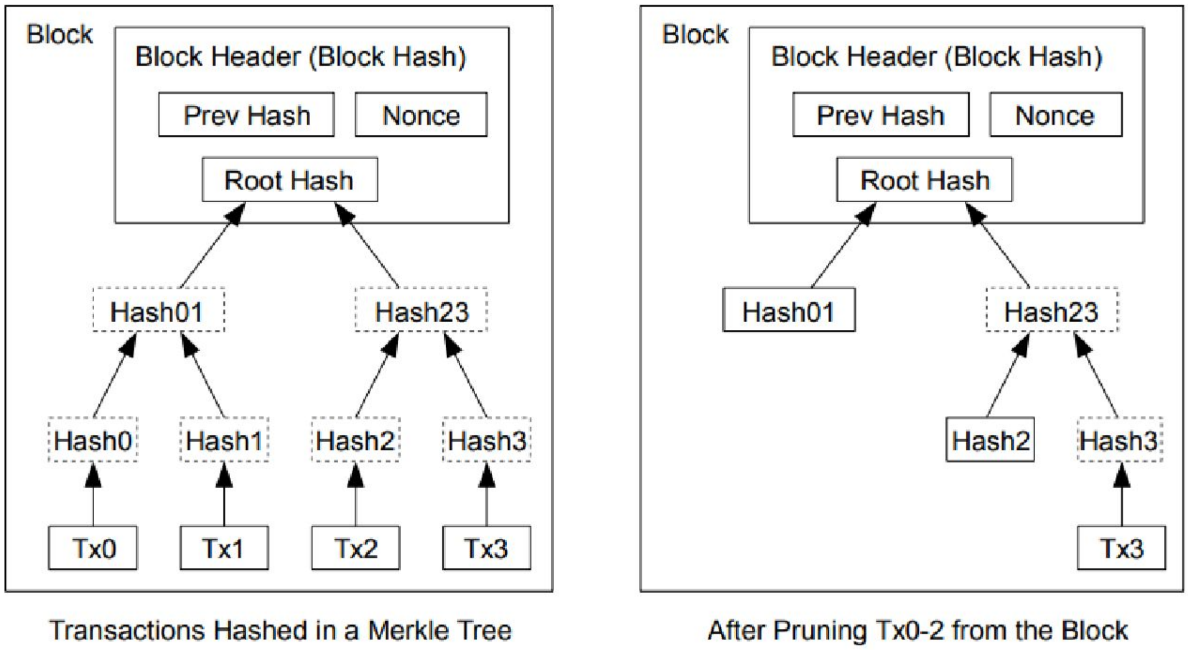

Fig. 5. The root hash without Merkle trees (left) and with Merkle trees (right)

Block header or block hash of fig. 5 is the part of the transaction block that contains all of the hashes. It can be said that this block represents some kind of "identifier" for the uniqueness of each block from the blocks in the chain. "Previous hash" (PrevHash) field contains the hashes from time markings (timestamp) explained afore in the paper. In [5] there is an example of a code that does the hashing of several given transactions according to the rules of the Merkle tree.

The "Nonce" field is a randomly selected number included in the block title. The miners are trying to "guess" this random number by trying random numbers until they find one that triggers the hash of the whole header to comply with the condition given by proof-of-work. The condition is actually a randomly selected number whose SHA 256 hash starts with a certain number of zeros. This concept with example code is given further in the section dedicated to SHA 256.

4. Summary

Now that the main features of blockchain and its way of functioning as well the algorithmic structures used were explained, the explanation of the structure and the way of blockchain functioning can be finalized. In Fig. 6. a simplified version of the Bitcoin blockchain is shown.

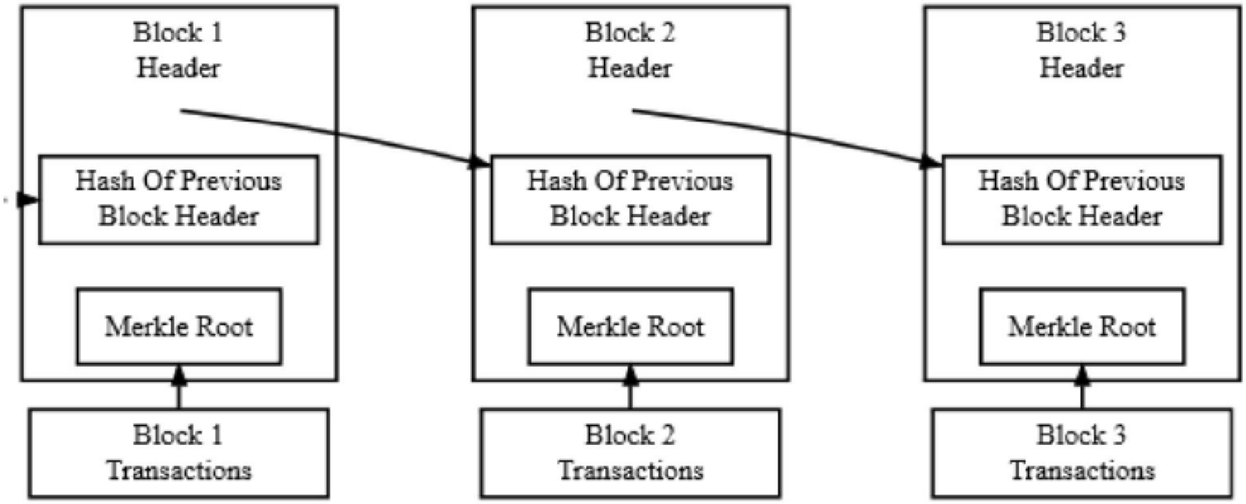

Fig. 6. A simplified version of the blockchain

The source [4] is actually a blockchain researcher. It was already said that all transactions are public and through the source, they can be viewed at the right time. It contains all the important information about the block and transactions, elements of which have already been explained in the paper, but it is interesting to see them at the right time and to feel their real application. The data are the following, given in fig. 7.

Summary
Number Of Transactions 303
Output Total 5,336.46944189 BTC
Estimated Transaction Volume 101.84009035 BTC
Transaction Fees 0.12508524 BTC
Height 520466 (Main Chain)
Timestamp 2018-04-29 18:56:23
Received Time 2018-04-29 18:56:23 Relayed By AntPool
Difficulty 4,022,059,196,164.95
Bits 390462291
Size 305.005 kB
Weight 1128.928 kWU
Version 0x20000000
Nonce 2249578904
Block Reward 12.5 BTC

Hashes
Hash 00000000000000000022e10f79aa
7792e6ebf2b82aaac9bd06ee91644e4171ea
Previous Block 0000000000000000001ca536123
9ede719aa2107c1b41194b4c7456462c24dd
Next Block(s) 0000000000000000001cc67b6
51d0861ae03fc334c6e1eabe6f947f20d678dd9
Merkle Root
A70cbf6551a835dd633efd2d
02192261d4b4f2995d5f03457c0f9912baab1e39

Fig. 7. The data from a specific block in real-time

The whole process of execution of transactions can be reduced to the following 6 steps:
1) New transactions are announced to all users.
2) Each user collects a new batch of transactions in one block.
3) Each user works on finding the unique number that is, proof-of-work.
4) When a certain user finds the proof-of-work, he or she announces the block to other users.
5) Users check if transactions are valid (Merkle trees concept is mentioned above).
6) Users accept the block by starting to work on creating the next block in the chain, using the hash of the accepted block as a previous block of the block they create.

Users always consider the longest chain as a correct one and continue to work on extending his length. If two users post different versions of the next block at the same time, then the other users work on the first block they received, but they also save the second block in a new branch. This process ends when the next proof-of-work is found and one branch will become longer. According to the hash of the proof-of-work, it will be known whether the first or second published block is the one that is part of the longest chain.

III. SECURE HASH ALGORITHM 256

1. Description of the algorithm

Security hash algorithms are iterative, one-way hash functions that can process messages and produce an abbreviated representation of the message. This representation is called a hash or in some books, it is also referred to as "message digest". [8] [7]

Once the message is processed this way, any change of message reflects on the result i.e. it changes the hash of the message. Digital signatures, messages for authenticity, and random numbers can be generated with this algorithm. This algorithm is irreversible only because of the large number of possibilities that exist, that is, there is no better way to guess the data whose hash is given than ordinary circulation through all possible variants.

The SHA 256 hash algorithm takes the input of 512 bits, combines data cryptographically, and generates an output of 256 bits. This algorithm actually repeats one relatively simple round of data combining 64 times. In Fig. 8 the hashing process is shown graphically, hashes are generated here from the letter A to the letter H. [8]

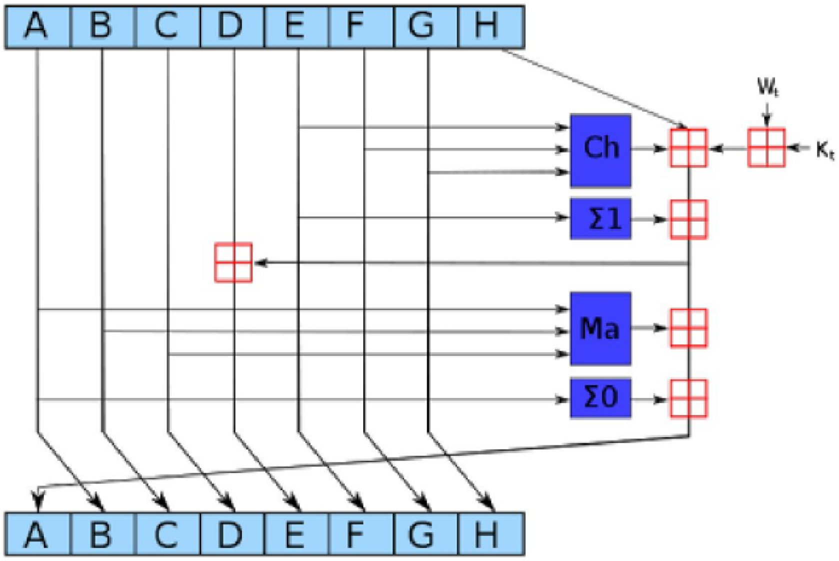

Fig. 8. A round of the SHA256 algorithm shows the steps of processing and new blocks

The blue blocks combine the values in nonlinear ways that are difficult to be analyzed cryptographically. The "Ma" block is called the block of majors, it sees the bits of A, B, and C. For each position, if there is a greater number of 0 bits, 0 is generated. If there is a greater number of 1 bits, 1 is generated.

The block "Σ0" rotates or moves the bits of each of the letters A to obtain three rotated versions of the letter and then assembles them together by module of 2 (this can also be interpreted as an operation of exclusivity i.e. XOR between the three rotated versions). In practice this means that if the number 1 is odd, the sum is 1, otherwise, it is 0. The three rotated values give the value of the specific letter rotated by 2 bits, 13 bits, and 22 bits.

The "Ch" block chooses output bits based on the input value of E. If the bit of E is 1, the output bit is the corresponding bit of F. If the bit of E is 0, the output bit is the corresponding bit of G. In this way the values of F and G are mixed with respect to the value of E.

The block "Σ1" rotates the sums of the bits of E, but this time for 6, 11, and 25 bits. Red blocks make the addition of 32 bits, generating new values for A and E. The input Wt is based on input data. Kt is a constant defined in each new round. As seen from the diagram in fig. 8, only A and E change in one round. The rest of the values stay unchanged; the old value of A becomes the new value of B, the old value of B becomes the new C, and so on. Although it seems that a lot does not happen in one round, this step is repeated 64 times, the hash becomes unrecognizable (specifically with Bitcoin a binary SHA 256 algorithm is used, because the hash of blocks is usually larger than 512 and instead of 64 rounds it has 128 rounds, i.e. the algorithm is only adapted for a larger number of bits). A great example of this is the hash made in [7] on page 34. There the difference of the hash is noticeable between the word "cryptography3" which looks like this c266a5386fd9eb9330cfe0bcea32c31611578eb1db9fa57246006c1fb6626396, and the word "Cyprtography2" at the hash output which looks like this 24dabb19cd16be8dce985ec

10847cc2e7e 38634fb50c263593d865d134e9077. This concept of hashing is crucial in the process of mining.

2. Bitcoin mining

Mining requires a very difficult task to perform, but easy to verify. Bitcoin mining uses cryptography with a hash function called binary SHA256. There is no better way to find the hash input of cryptographic hash functions than the principle of trial and error, trying multiple input data [8]. Once the input is found it is easy to verify data. This does this algorithm suitable for implementation in the proof-of-work of Bitcoin. Block transactions are collected first. The block is then hashed to form a 256-bit hash value. If the hash starts with a sufficient number of zeros (an indicator of the input number exponent), then the block is successfully "mined" and sent to the Bitcoin network and the hash becomes an identifier for that block. Most of the time the hash is not successful, so the block is modified (usually time markers) and it is tried again. The goal is to find a "unique number" (nonce) and time marking that corresponds to the specific block. In practice, it is necessary to actually guess the number of zeros before the proof-of-work (the process of defining the weight of the proof-of-work is simplified by the statement that it is enough just to guess the number of zeros of the hash). The definition of the weight of proof-of-work is much more complex and variable i.e. it depends on the specific weight at a certain point in time and the value of a reference basic hash. In terms of computing power invested in the Bitcoin network, this weight changes. The average that is targeted to maintain is 1 block every 10 minutes. The source [9] provides a Github link to the code that performs single block mining. It is important to note that the code is just an example and the values are taken from a block that already exists on the blockchain. In reality, most of the "mining" attempts of a particular block are unsuccessful. Therefore, none of the "unique numbers" correspond to the set goal. In this case, it is necessary to modify the timestamp. At the same time it is also possible to add new transactions to the block and this as mentioned above changes Merkle's root. A new Merkle root means a complete change of hash of the block or searches for the unique value again. The specific Python code [9] generates 42,000 hashes per second, which is approximately 1 million times slower than the hardware used by real miners. At this speed, that code would take 11 million years on average, for digging up a block. From this point, it can be concluded that mining is very difficult and that is exactly one of the principles that guarantee the security of Bitcoin.

3. The security of Bitcoin blocks and the chance of creating a fake chain

The weight of the mining blocks is huge even difficult to conceive. At the specific weight of the proof-of-work at this point, the chance of finding a successful hash is 10^19. Finding a successful hash is harder than looking for one specific grain of sand from all grains of sand on planet Earth. [11] At the moment the whole network of miners of Bitcoin generates about 25 million giga-hashes in second (this estimate is made from the total Bitcoin representation in the cryptocurrency market and the average hash rate of commercial hardware specially made for Bitcoin mining). The reward for a mined block is 12.5 Bitcoins per block at the moment and also the miners receive tax from transactions (about 0.1 Bitcoin per 1 block). This is approximately 20,000.00 USD per block although this estimate is difficult due to price volatility.

When these two things are taken into account, the economic incentive and weight of mining, it becomes clear how small it is the likelihood of a system attacker creating a fake block or chain of blocks (reminder that hash of every block contains information about all blocks before). In source [2], a mathematical prediction based on Poisson's equation of what would happen if anyone wants to do the above-said attack is stated. The bottom line is that the probability is so small that the full computing power that exists at the moment would not be sufficient to succeed to consistently create a fake blockchain. The Moore's Law of hardware development shows that the country would rather reach the thermal death of the universe than to develop hardware capable of hacking the SHA 256 algorithm. [13] [2]

IV. DIGITAL KEYS

The concept of digital keys belongs to the class of asymmetric cryptography. This is a concept that is not unique to Bitcoin and has many applications outside the blockchain in many open communication networks like for example the Internet. The unique part about Bitcoin is the way of coding the information and protocols that are used. The process will be explained, but there will not be detailed talks about these protocols.

This encryption uses a pair of keys, one public and the other private, to define the reliability of the information. Public keys can be freely released to the public. Without the private key, it is almost impossible to detect the original message. Fig. 9 is a general outline of how the process of generating Bitcoin addresses using public and private key takes place. [12] The “Base 58 check encode” type of encodings are required only to make the private key and the public key more presentable and understandable.

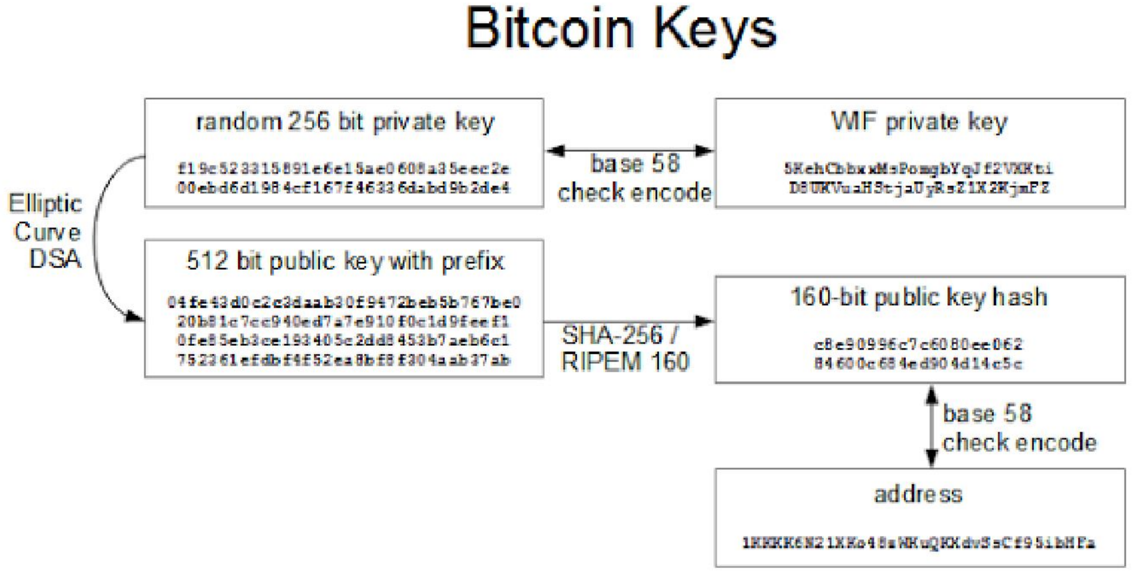
Bitcoin Keys
random 256 bit private key
f19c523315891e6e15ae0608a35eec2e
00ebd6d1984cf167f46336dabd9b2de4
base 58
check encode
WIF private key
5KehCbbxxMsPomgbYqJf2VXKti
D8UKVuaHStjaUyRsZ1X2KjmFZ
Elliptic
Curve
DSA
512 bit public key with prefix
04fe43d0c2c3daab30f9472beb5b767be0
20b81c7cc940ed7a7e910f0c1d9feef1
0fe85eb3ce193405c2dd8453b7aeb6c1
752361efdbf4f52ea8bf8f304aab37ab
SHA-256 /
RIPEM 160
160-bit public key hash
c8e90996c7c6080ee062
84600c684ed904d14c5c
base 58
check encode
address
1KKKK6N21XKo48zWKuQKXdvSsCf95ibHFa

Fig. 9. The process of generating addresses with the help of public and private key

In Fig. 10 a graphic representation of the transactions between addresses is shown. It will not be discussed in detail, because it is not the subject of the seminar paper, but it is good to know that transactions are not just Bitcoin shipments from one address to another. It is combining inputs and outputs of transactions (input-how many Bitcoins have reached the address, output-how many Bitcoins are sent from the address). The picture illustrates the rule that says that every entry must be fully consumed in one transaction. If for example, one address received 100 Bitcoins and wants to spend 1 Bitcoin, the transaction takes place in a way that the first transaction sends all 100 Bitcoins to a specific address, and then 99 Bitcoins are sent from that address to the source. Transactions also have fees as was already said.

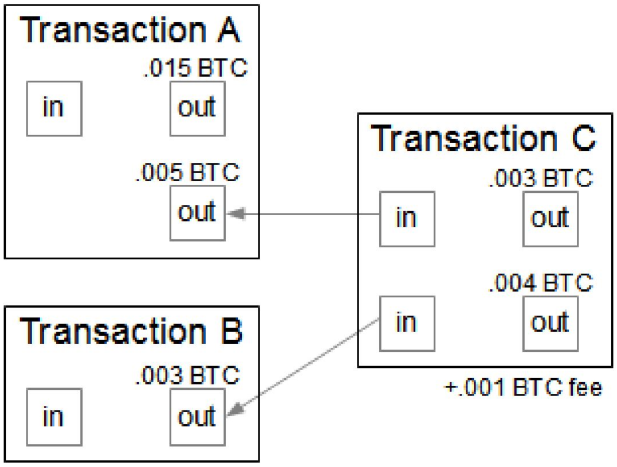

Fig. 10. How a Bitcoin transaction takes place

The next section that will be explained is related to signing transactions, (part of this was explained above) i.e. how from the private key a public key is generated.

The Elliptic Curve algorithm is used. Elliptic curves are an interesting class of curves that are also used to solve Fermat's Last Theorem [14]. In fig. 11 the elliptic curve that is used in Bitcoin is shown. [10] The characteristic of these curves is that if you pull a non-vertical line that cuts two non-tangent points (P and Q) of the curve, the curve then it will certainly intersect a third point R '. The formula P + Q = R applies here.

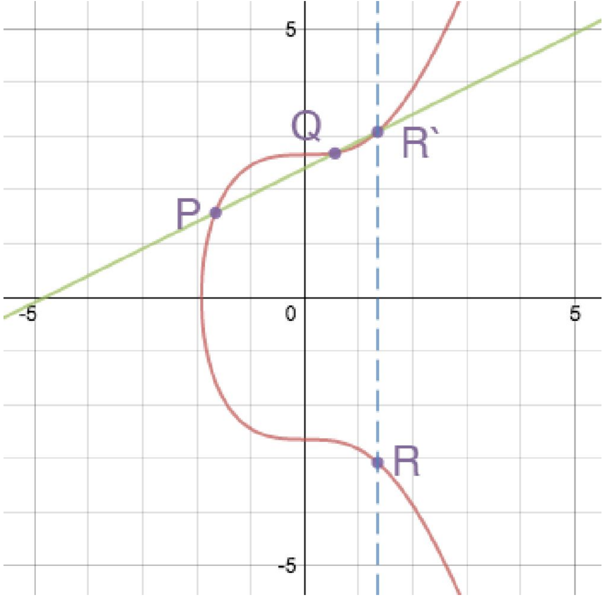

Fig.11. The elliptic curve with equation: y ^ 2 = x ^ 3 + ax + b, a = 0, b = 7

Bitcoin retains this mathematical concept, but instead, elliptic curves use ECDSA (Elliptic Curve Digital Signature Algorithm) in the context of finite fields. The same equation in Figure 11, but drawn in a finite field with a modulus of 67, looks like it is shown in Figure 12. This looks different visually, but the symmetry of the curves is retained (although now the graph looks like randomly arranged points on a plane). The reliability of this system is based on the use of specific values known as “secp256k1” curves.

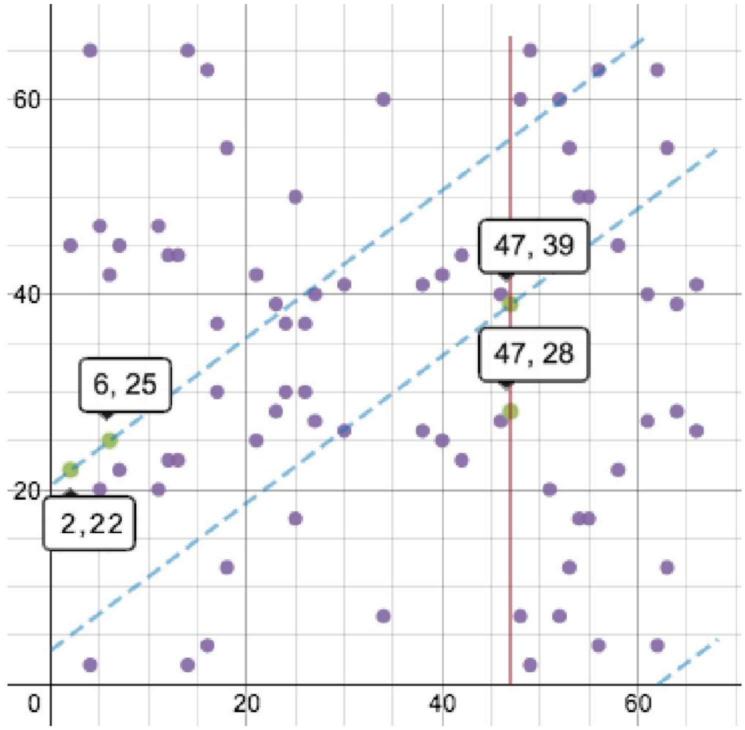

Fig. 12. For the pair of points (2.22) and (6.25) the intersection point is (47.39), and (47.28) is the point of reflection (analogous to R' in ordinary curves)

Besides these characteristics, one of the main features is that with the help of these curves the multiplication of integers is done quickly, but for division, a trial and error algorithm is needed and there is no better way to do it. For example, 12345 * A = Q can be calculated very fast (using degrees of 2) but if only A and Q are known, solving n * A = Q is difficult. In the given example the secret number “n” or private key is in V_{il} 12345 and the public key would be Q.

The algorithm of digital signatures with elliptic curves (Elliptic Curve Digital Signature Algorithm -ECDSA) takes the hash of the message, and then does the arithmetic of the elliptical curves, using the message, a private key, and a random number. In this way, a new point on the curve that gives the signature is generated. Anyone who has the public key, the message, and the signature can perform a simple arithmetic operation with elliptic curves to verify that the signature is valid. It can be concluded that only the owner of the private key can sign the message, but anyone who has the public key can verify the message. Combining this with SHA256 and the mathematics behind number theory, Satoshi creates one algorithmic structure whose reliability is excellent secured via computer code. These details are explained much more in [12], but I consider that the detailed analysis of these algorithms is a separate topic in itself.

V. CONCLUSION

Bitcoin creates a trust system through SHA256 hashing and cryptography of elliptic curves. This trust is key to the nature of decentralized currencies like Bitcoin. Traditional currencies are based on banking systems and governments as systems of trust. Many economists criticize these currencies because they actually have no reference to the movement of price, but as can be easily deduced from this paper, the price of Bitcoin is actually dependent on the computing power invested in the system, and the price of Bitcoin. With that, much of the models of consumption and demand are satisfied. The purpose of this paper was to expose Bitcoin’s background and show the ingenious idea of Satoshi from a computer mathematical point of view. With the help of algorithms, the first functional system of consensus is created. It is a perfect mediator that can be trusted and proof of this is the cryptography exposed in this paper. Satoshi's idea at its core is really idealistic from a political and momentary point of view and is a revolt and a message to the financial giants of the major stock exchanges in the world. A message that maybe, soon, in the near future they need to prepare for a different monetary system where trust is determined by mathematical algorithms, not by centralized authorities at high political positions.

References:

[1] IEEE Spectrum issue No. 10.2017 p.20

[2] Bitcoin white paper by S Nakamoto published 2009

[3] https://bitcoin.org/en/developer-guide

[4] https://blockchain.info

[5] https://gist.githubusercontent.com/shirriff/c9fb5d98e6da79d9a772/raw/18520930523f8e2f729b30033c2f90ee6a2bf4f0/merkle.py

[6] Merkle, R. C. (1988). "A Digital Signature Based on a Conventional Encryption Function". *Advances in Cryptology — CRYPTO '87*. doi:10.1007/3-540-48184-2_32. ISBN 978-3-540-18796-7.

[7] https://csrc.nist.gov/publications/detail/fips/180/2/archive/2002-08-01

[8] FIPS 180-2 (August 2002 (Change Notice 1, 2/25/2004)) by National institute of standards and technology

[9] https://gist.githubusercontent.com/shirriff/cd5c66da6ba21a96bb26/raw/85a75b2fc2457f9e8bf5c148fde98b5ddd2094c3/mine.py

[10] An Introduction to the Theory of Elliptic Curves by Joseph H. Silverman Brown University, June 19 – July 7, 2006

[11] MS-ESS2-2 Construct an explanation based on evidence for how geoscience processes have changed Earth's surface at varying time and spatial scales.

[12] https://www.coindesk.com/math-behind-bitcoin/ by Eric Rykwalder, Oct 19, 2014 at 14:08 UTC | Updated Oct 19, 2014, at 20:23 UTC

[13] Thomson, William. (1851). "On the Dynamical Theory of Heat, with numerical results deduced from Mr. Joule’s Equivalent of a Thermal Unit, and M. Regnault’s Observations on Steam." Excerpts. [§§1–14 & §§99–100], Transactions of the Royal Society of Edinburgh, March 1851; and Philosophical Magazine IV. 1852. [from Mathematical and Physical Papers, vol. i, art. XLVIII, pp. 174]

[14] http://www.math.vt.edu/people/brown/doc/ellip.pdf by Ezra Brown, Mathematical Association of America