A

Bitcoins:A survey and comparison of parallel techniques

Paritosh P Ramanan, Georgia State University

Bitcoins have suddenly attained the limelight in recent times. Bitcoin a type of cryptocurrency is a digital token which guarantees the bearer an exchange for a certain amount of actual rela world currency. The exact value of a Bitcoin in terms of real money is determined by market forces. Bitcoin was first proposed in a white paper in 2008 which spoke about the necessity for such a concept and went on to provide a general outline for the network. In this paper, we take a look at the various aspects of the Bitcoin network like privacy, verifying legitimacy of transactions, the advantages and reward mechanism for participants followed by the disadvantages of the present system and alternate crypto currency techniques inspired by Bitcoin. We analyze each aspect by throwing light on how all these facets of Bitcoin network are accomplished in a distributed and parallel way.

General Terms: Parallel computing

Additional Key Words and Phrases: Bitcoins, public key encryption, distributed timestamping, pooling techniques

Contents

1. INTRODUCTION

The increased dependence of monetary transaction between two parties on computers has led to a wide variety of issues which need to be addressed. Current trends are more geared towards a financial institution acting as a third party who facilitates the transaction between the sender and receiver.This process is accompanied by mediation costs, the risk of a fraud by either parties due to the reversible nature of transactions. Inherent in this system is also the loss in the ability to make irreversible transactions which lead to an increase requirement of trust. As a result merchants must obtain permanent uniquely-identifiable information from prospective buyers in the hope of avoiding fraud. As mentioned earlier, this leads to increased costs, redundancy in information storage and trust issues.

1.1. Objective and problems in Cryptocurrency

In order to address all the above mentioned issues the system of a crypto currency has been suggested [Satoshi 2008] [Back 2002] which attempts to restrict the transaction only between two responsible parties and without depending on trust in facilitating the transaction. Cryptocurrency like Bitcoin works on the basis of cryptographic proof instead of trust. It removes the concept of the middle third party by directly engaging the buyer and seller. The premise of the crytocurrency is to develop a technique wherein a transaction once made is computationally too impractical to reverse. Cryptocurrency is analogous to legal tender or hard currency in the digital world. The analogy stems from the fact that like legal tender which once used in a transaction cannot be reversed theoretically. The purpose of Bitcoin is to replicate this mechanism prevalent in case of legal tender in the electronic domain which would allow for a seamless transaction of money between two individual removing the issues of trust and other logistical disadvantages. However to execute this principle a variety of challenges are faced. Primary among them is double spending. Since, a crypto currency is nothing but a sequence of bytes, it is easily possible of a malicious individual to make multiple copies of this sequence and try to spend them two or more times. This is impossible in case of hard cash but in the electronic domain this is very much possible. Thus there must exist in the network a mechanism to prevent double spending. This is accomplished by a variety of techniques which will be discussed in coming sections.

1.2. Bitcoin

It is nothing but a chain of transactions. It is a certificate which states what all transactions have taken place using this same Bitcoin previously. The bearer of one such bitcoin can exchange this in terms of actual real world currency. Why is a bitcoin so popular nowadays? It works completely on a peer2peer network which makes regulation nearly impossible. It is unlike any other real world currency because it is immune to regulation. No real world government or institution can claim ownership of the entire network or concept. It is entirely market regulated and depends on the basic economic principle of demand and supply. It is a cryptocurrency and using an unprecedented amount of parallel computing technique, the entire network is able to ensure fraud proof operation. In other words it is nearly impossible to fool the network and peddle ones own fake certificates or fake bitcoins posing as the real ones in the network. An interesting aspect is provided in [Ron and Shamir 2013] which gives a good overview of Bitcoin user behavior and general quantitative characteristics of the Bitcoin network.

1.3. How does Bitcoin work?: A Birds Eye view

The cornerstone of the entire bitcoin network is a transaction. A transaction is nothing but transfer of a bitcoin from one owner to another. There could be a few potential issues with this. One is that somewhere down the line when the bitcoin has changed many owners, a malicious owner might try to double spend the bitcoin. In other words, he might try to sell the same bitcoin to two different owners at the same time. Obviously such a transaction should be illegal in the system as it has the potential to throw the entire network out of gear. To solve it, the bitcoin introduces the concept of block chain which in simple terms is a record of all previous transactions. Therefore when a transaction is about to happen, one can easily check previous records to make sure there has been no double spending. This is done using a distributed timestamp server which is based on a peer to peer network, which would provide the necessary proof of previous transactions when required.

1.4. Bitcoin Transactions

Each owner has his own public key and a private key. In a nutshell what happens when a bitcoin is passed on to another owner is that a new record is created in the block chain which has the hash of the previous transaction and the new owners public key as inputs. The result is appended to the end of the bitcoin thereby completing the transactions. But problem of double spending still persists. What if the owner, were to make a digital copy and use it twice.

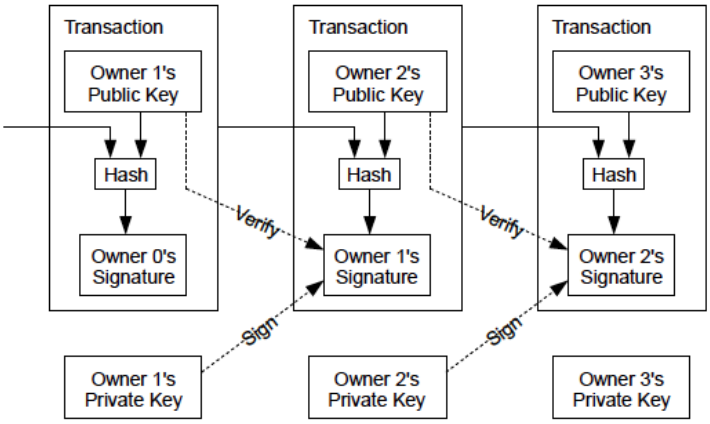


Fig. 1. Sequence of events representing a transaction[Satoshi 2008]

There must be a way to ensure that the previous owners did not sign any earlier transactions. Since the earliest transaction is the one that needs to be examined, later attempts to double spend can be neglected. If we are aware of all transactions we can be reasonably clear about the absence of any double spending operations. The mint is aware of all transactions and decides which arrived first. In order to completely do away with the concept of a thrid party validator all transactions must be publicly announced. This implies all participants must agree on a single history. On a related note, [Singh et al. 2013] propose an extension to the Bitcoin network that works on mutual trust but promises faster transaction time by moving it from a Pioneer model to a mutual trust based model.

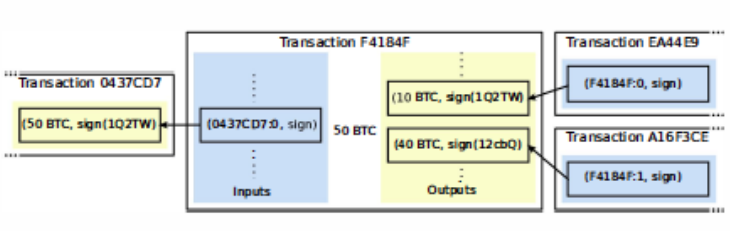


Fig. 2. An example transaction[Decker and Wattenhofer 2013]

To address this problem the requirement of a central mint is necessary. This mint will have the task of verifying each transaction and validating its correctness thereby preventing fraud. Mint is being implemented in a peer-2-peer model. But central mint is giving too much

Solution will be to establish a timestamp server. Each transaction will be timestamped. It is a unique way of making sure the amount of knowledge that was present at that exact point of time. This will help check whether any of the previous owners signed any of the previous documents. Each new transaction will take into account the previous timestamp and will include it in the hash to create the new timestamp for the new transaction.

Paper on [Back 2002] is taken as the backbone for implementing such a network. The technique of proof of work has been explained in detail in that work and is the basis for a similar technique being implemented in the bitcoin network too. Each block has a field called nonce. It is important to note that each transaction spawns a new block. The idea of the proof of work is that each block must hash and generate only a certain number of zeroes. By manipulating the nonce field and by trial and error in an increasing fashion, a nonce value will be found with will give upon hashing a certain number of bits to the block. This will form the block. When after each successive transaction, the chain keeps getting augmented with new blocks with different nonce values depending on the nonce value of not just the current block but all previous blocks because as can be noted, each transaction also depends on previous transactions. The advantage of this process is that if an attacker were to manipulate the network and try to insert a wrong value, he would have to change the nonce of the entire chain henceforth which would prove to be computationally impractical.

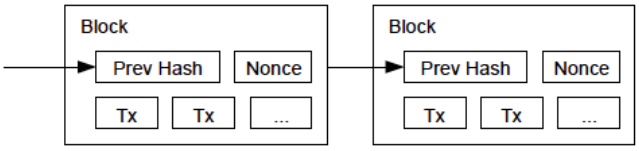


Fig. 3. Illustration of Proof-of-work[Satoshi 2008]

The paper then talks about how unlikely and computationally hard it is for an attacker to replicate the same chain and keep pace with the network with intent to destroy it successfully. Since it is not in the interest of a parallel technique we can skip this part.

Paper further talks about incentivizing those nodes that lend the CPU time and electricity for facilitating the transactions and being part of the peer to peer network by generating new bitcoins for such nodes. If the inout value is greater than the output

value, the difference is provided as incentive for solving the block. The logic behind this is to pre-empt any attacker from using his computing powers to attack the system by offering him an even more profitable opportunity by help being part of the network and minting new bit coins.

1.5. Block and block chain

The block is nothing but a transaction. Block chain is nothing but a public ledger which keeps track of all transaction in the bit coin network. The blockchain is a public record of all transactions in the Bitcoin network. Blockchain.info allows you to navigate the bitcoin blockchain.[Decker and Wattenhofer 2013]. Thus when a transaction is performed the node handling the transaction must propagate this information and make sure that it is committed to other copies of the ledger. How this is done is explained in the next work.

2. INFORMATION PROPAGATION IN BITCOIN NETWORK

[Decker and Wattenhofer 2013] talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv message stating that it has these many blocks corresponding to some transactions. Any node not having the said information responds back and transfer takes place. This is done to save bandwidth as the information to be exchanged is of considerable size. The authors in [**?**] examined the

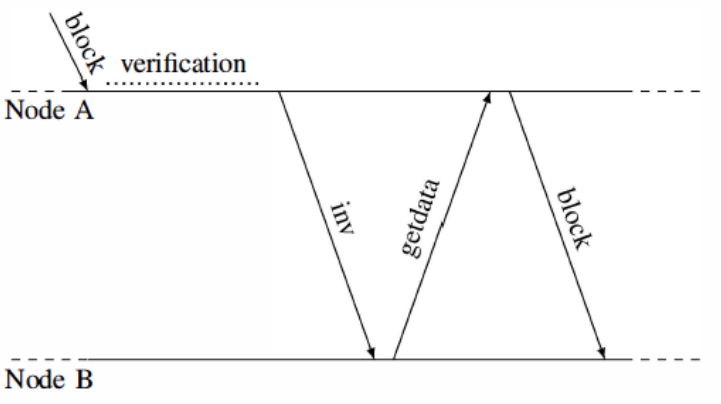


Fig. 4. Timeline diagram regarding block discovery and propagation[Decker and Wattenhofer 2013]

blocks propagated in the network between height 180000 and 190000. Figure 6 depicts the PDF for the time at which the last block was found at this level.

Figure 8 gives the probability distribution function on how much time on an average it takes for nodes to learn about the blocks. The paper talks about blockchain forks and how they are created. Due to the concept of proof-of-work the valid blocks are to be found independently at random.

2.1. Where is parallel computing involved?

Starting from the beginning this is the sequence of events which goes on. An entity A wants to send BTCs to another entity B, and this is referred to as a transaction.

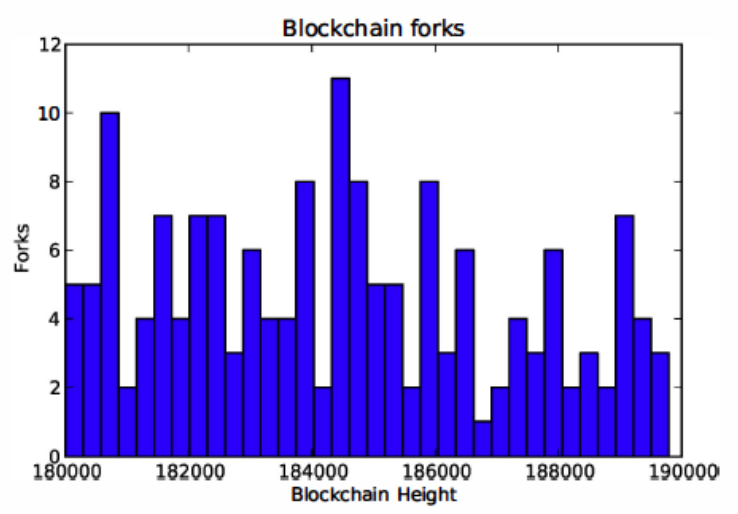


Fig. 5. Number of forks and their heights[Decker and Wattenhofer 2013]

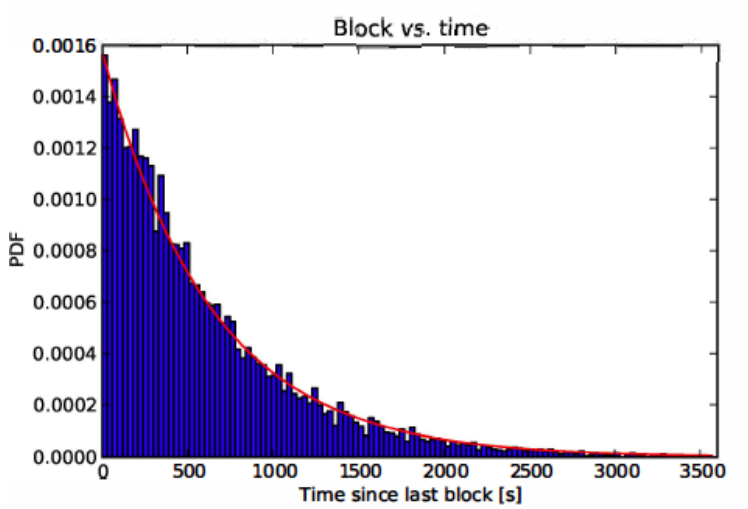


Fig. 6. Timeline diagram regarding block discovery and propagation[Decker and Wattenhofer 2013]

This transaction typically is validated by using the techniques described in the previous sections. This transaction is *heard* by one of the nodes which picks it up creates a *block* and tries to publish it publicly i.e. to all other nodes in the network. But there are other such transactions being handled by other nodes simultaneously.The problem now is that how will a node decide which transaction happened first- the one which is represented by the block it received or the one that happened under its own watch. It is important to note that to keep the copy of the ledger fresh, only the latest transactions ordered in the correct chronological order must be put in. To do this, each node tentatively commits the transactions which it has knowledge of. If an earlier transaction is received as a broadcast from some other node then it is supposed to roll back the commit, put the latest information in and then push the remaining information in. This

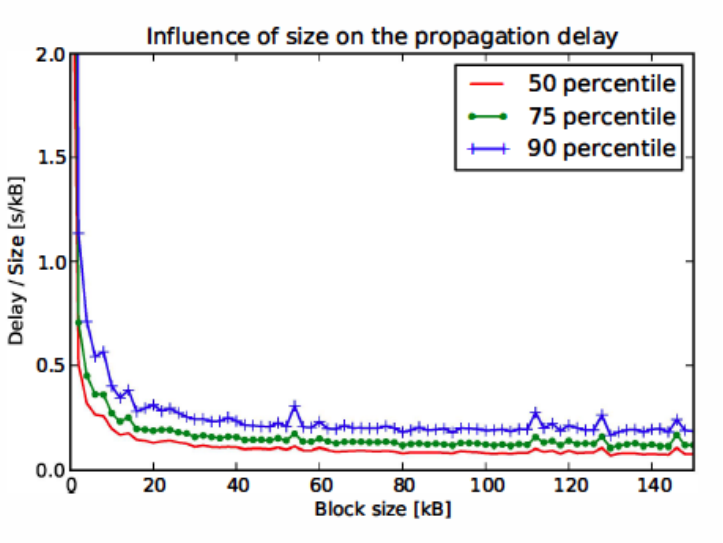


Fig. 7. Timeline diagram regarding block discovery and propagation[Decker and Wattenhofer 2013]

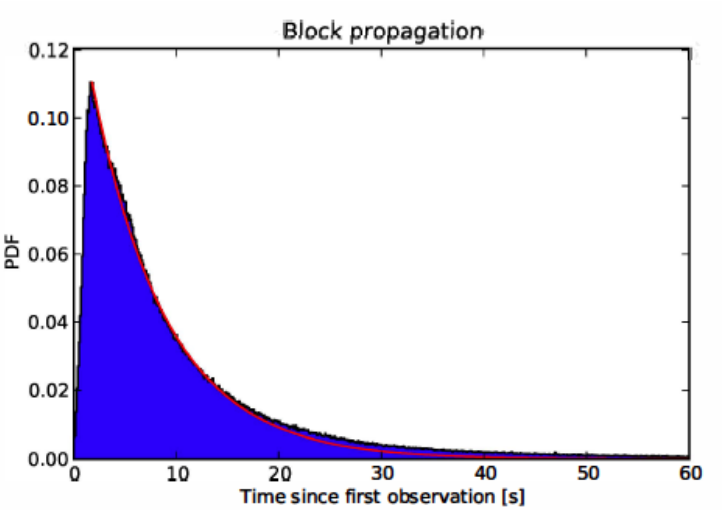


Fig. 8. Normalized histogram for all blocks in a given time interval[Decker and Wattenhofer 2013]

step though it looks simple actually is a bit more complicated than that. This problem refers to solve the proof of work problem referred to above. To determine the order the nodes attempt to find a solution to a proof-of-work. The proof-of-work consists in finding a byte string, called nonce, as illustrated in the previous section, that when combined with the block header has to yield a hash with a given number of zeroes. This problem is actually a computationally hard problem since cryptographic hash functions are one-way functions. To find the actual nonce string we have to analyze all possibilities which would give us the correct value. Once we obtain the correct value, it is easy to verify its correctness. The node finding the nonce value first will then send the block after embedding it with the nonce value to all other nodes. Since it is easy

to verify the authenticity, they will then accept or reject the solution and accordingly make changes to their own copies of the ledger. This aspect of the network is what is the most powerful feature of the network.

2.2. Blockchain

Having covered the concept of blocks and transactions, we now come to the concept of block chains. Blockchains are nothing but a directed tree with individual blocks for nodes with the latest node being referred to as the block chain head. The height of the tree is referred to as h. There can be a situation in which different blockchain heads can exist at different heights. In such a scenario, if a node receives a block with a height greater than the block chain height of its own ledger copy, there can be two cases, one in which the existing block chain head is an ancestor of the received block or when its not. If it is the ancestor, it is obvious that the node has missed out on some transactions in between and will get the required information from other nodes and attempt to keep its copy of the ledger fresh. If however it is not an ancestor, it is clear that, both share a common ancestor, and the node will then find refresh the list starting from the nearest common ancestor to keep its copy fresh.

3. KEYS:MAINTAINING PRIVACY IN A DISTRIBUTED SETTING

A necessary aspect in the scenario for Bitcoins is the need to maintain privacy but at the same time ensuring that safety in the transaction. This is particularly important because we need to maintain the principles of authenticity,integrity and non repudiation while implementing the Bitcoin network. Authenticity is the ability to identify the sender of the message. Integrity implies that the message being sent must not be compromised by a malicious third party. Any person in the middle must not be able to significantly alter or replace a part of the message being sent. Non repudiation means that the sender must not be able to refute the sending of the message sometime in the future after it has been received by the receiver [Schneier 1993] . It is in this regard that cryptographic concepts of hash function is used in this network. hash functions provide the much needed support in the Bitcoin network with the use of two sets of keys i.e. the public and private keys. These keys accomplish the dual purpose of concealing the identity of the individual involved in the transaction while ensuring the safety of the transactions. Since the ledger is public and scattered across a p2p network, this is accomplished among the many nodes in a distributed way. It is vital to the network to avoid fraud or theft of coins. This section focusses on methods which ensure authenticity,integrity and non-repudiation in a Bitcoin transaction between two interested parties without the involvement of any third entity.

3.1. Hashing:A basic overview

If two parties want to send or receive messages, they can use encryption to hide the messages. The receiver can then perform what is referred to as decryption to recover the original message. It is often the case that the algorithms for encryption and decryption are well known and the receiver can recover the original message using what is referred to as a key. If the key is the same as the one used for encryption, it is referred to as a *symmetric encryption*. The advantage of symmetric algorithms lies in the aspect of confidentiality, but the process for maintaining a common key between sender and receiver is often cumbersome. This is the motivation for use of hash functions. A hash function akes a variable-length input string referred to as a pre-image and converts it to a fixed-length output string called a hash value.[Schneier 1993]. A subset of this set of hash functions is the one-way hash function in which it is easy to compute the hash from a given key, but computationally hard to compute the key if given the hash. When hash functions are employed, it gives us a way to implement

all the necessities mentioned before like authenticity, integrity and non-repudiation in a distributed way.Hash functions provide a way of *digitally signing* which can be thought of as analogous to a signature or a stamp of authenticity.

3.2. Public key cryptography

Public key cryptography is a novel way of going about solving the problem about encryption decryption and digital signing. As mentioned before, we implement this concept by creating two different keys or a key pair: the public key and the private key. It is worth noting that deriving the private key from the given public key is computationally hard. The advantage of this approach lies in the fact that anyone with a public key can encrypt the message but not decrypt it. Only the person with the private key can decrypt it. In the case of the Bitcoin network, public keys serve as unique identifiers which help one send messages but not let anyone read it. We can accomplish digital signing by finding the hash of the message and encrypting it with the private key to form a digital signature. Public keys also serves as a method for verifying the authenticity and the integrity because with the public key one can decrypt the signature and and compare the result with the hash of the message. This also implements non repudiation as the sender is not able to falsely deny the sending of the message. The paper [Diffie and Hellman 1979] provides more in-depth detail about public key encryption systems and the motivation behind it and is considered a seminal paper in the said domain.

3.3. Protocols for Public key encryption

Although the concepts for public keys is useful, there is still scope for developing techniques which handle the exchange of keys in a distributed environment. It is important to note that in case of the BTC network, the public and private key distribution becomes a challenge as it is completely a p2p network scattered all across the internet. The paper [Merkle 1980] provides some established protocols which deal with exchange of keys in a distributed environment. In [Merkle 1980] is highlighted several techniques relating to key distribution. The first one among them is the centralized key distribution technique which employs a central entity or a distribution center which serves as a repository for all the agents to deposit their respective keys. If any two agents wish to communicate with each other they contact the distribution center and obtain the keys. In case of the Bitcoin network however, such an approach is not feasible because, the entire network design being a peer-2-peer one. Maintaining a central repository will defeat the purpose of non-involvement of a third party in the transaction. Another disadvantage is the single point of failure in the distribution center which when compromised has the potential to bring down the entire network. The paper therefore proposes a techniques which can be used for key distribution. Let A and B be two agents wishing to communicate. Both A and B generate E_A,E_B which are the public keys and D_A,D_B which are the private keys. When A wants to communicate with B, it signs a message m by computing the hash of $D_A(m)$ and sends it to B. B then deciphers $m = E_A(D_A(m))$ and verifies the correctness of the sent message. This is possible in part due to the concept prevailing in the one way encryption. In this way, we achieve authenticity, integrity and no-repudiation in the network. The work presented in [Rivest et al. 1978] gives a detailed description of methods for obtaining digital signatures and public-key crypto systems. It proposes a scheme which involves an exponential rate hash function. The encryption is performed in the following way

— hash m to a generally known power e
— then divide the value obtained by the product n of two obscure prime numbers
— record the remainder

$$ed = l(mod(p-1) * (q-1)) \quad (1)$$

The authors claim that more the difficulty in factoring the published divisor more secure the system. [Denning 1984] presents another work which talks about vulnerabilities in public key crypto systems and goes on to describe a method to foil attacks arising out of such vulnerabilities. The author argues that an attacker can get the victim to sign new messages derived by intercepting messages to the victim and then forging the signature of the victim. [Okamoto et al. 1998] and [Fujisaki and Okamoto 1999] provide description of various types of probabilistic cryptographic techniques and an approach to enhance security in public key encryption systems by preventing cipher text attacks.

In this section we saw techniques which guarantee the safety of transactions in a Bitcoin network. Next section deals with how the Bitcoin network accomplishes the genuineness of a transaction and prevents double spending in a setting where all participating nodes might not have all the information all the time.

4. TIMESTAMPING:AN EXTENSION TO PROOF-OF-WORK

In Section 1.4 we provided an overview of the timestamping process which goes on in the network. In this section we deal with a deeper analysis of the concept of timestamping and its relevance to the Bitcoin network. This section describes approaches regarding timestamping in situations wherein the trust factor is distributed i.e. in cases where there is no centralized authority to guarantee trustworthiness of a transaction.

4.1. Why is it a necessity?

Timestamping basically is a measure to prevent double spending. By timestamping a certain transaction we guarantee the temporal aspect of the transaction. In the Bitcoin network, it is also essential to keep track of previous transactions too to instill legitimacy in the transaction. This section talks about some approaches which can be used for timestamping as referred to by [Satoshi 2008].

4.2. Timestamping: A general outline

Timestamping has been proposed as a solution to digitally certify the existence of a particular document or record at a certain point of time. There are certain conditions that need to be implemented to achieve a stable system of verifying against forgery of timestamping. Thus, there have been recent attempts to develop timestamping schemes which will be computationally very hard to fake. One such method is to link the present timestamp using an appropriate hash function to the previously found timestamp. As this chain grows, it becomes increasingly difficult to for any attacker to forge timestamps by manipulating the bit-strings[Haber and Stornetta 1991]. The timestamping scheme being used in the Bitcoin network is inspired from the works [Haber and Stornetta 1997] and [Massias et al. 1999]. The scheme consists of n requests for timestamping and a method which would convert it into one single value which could then be given out to all the requesters as the generated timestamp. This scheme uses a binary tree structure and works by ”rounds”. In order to timestamp, it now becomes essential to granulize time by dividing it into slots. This scheme groups all the timestamping requests arriving in a particular slot. It is important to note that timestamp requests usually consist of a hash of the document whose timestamp is sought. In case of bit coin, it is the transaction whose timestamp we are interested in finding. The method mentioned in [Haber and Stornetta 1997] consists of a round root value and the round value which are cumulatively used to find the timestamp. If y_i is one timestamping request, the binary tree is constructed in the following way.

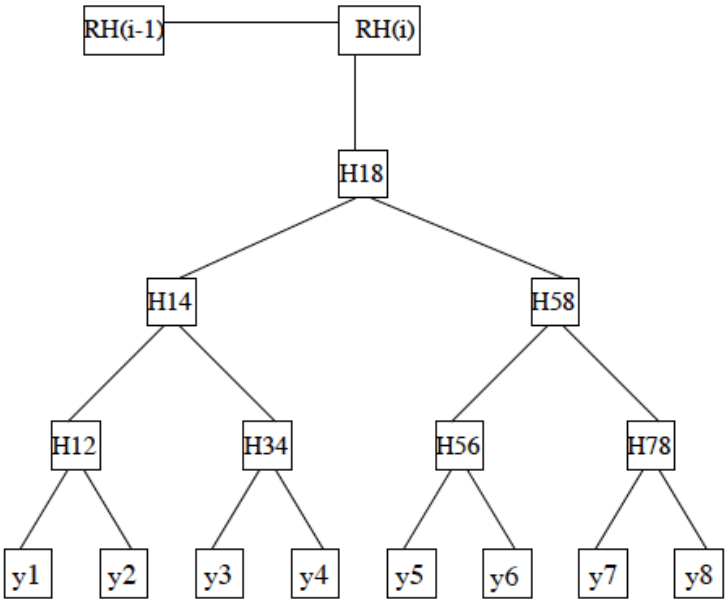


Fig. 9. Timestamping technique[Massias et al. 1999]

— Find the hash of y_i and y_{i-1} as the parent of y_i and y_{i-1}
— Find the transitive closure on these hashes by combining siblings until a single value remains.
— Set this value as the ”round root value” value for this particular slot and obtain the ”round value” by concatenating this with the previous value and hashing them again.

As can be observed from Figure 9, the leaf nodes of the tree are the timestamping requests. Their transitive closure of hashes results in the formation of the tree. H_{18} is now obtained, and hashed with the previous hash value to obtain the timestamp RH_i. The timestamp of the document now contains all the values necessary to rebuild the entire tree. For instance, for y_3 the timestamp is $\{(y_3, L), (H_12, L), (H_58, R), (RH_i - 1, L)\}$. It is tone noted that this notation basically comprises of the left or the right sibling denoted by letter R or L and their hash values respectively. The general idea is that for verification purposes the ”Round value” is obtaining by systematically rebuilding the tree using the above mentioned information. The binary tree structure we are using in this method is referred to as the Merkle tree[Merkle 1980]. In a system where there is a centralized system of authority, [Massias et al. 1999] propose a technique wherein a timestamping scheme following the binary tree structure elaborated in [Haber and Stornetta 1997] is used and an improved scheme with minimum trust requirements. Figure 10 shows the above mentioned concepts of timestamping and formation of the Merkle tree. In the Bitcoin network, the Merkle tree is generated in a distributed way from all the transactions which have taken place in a given amount of time and bundling it into the Merkle tree. The author in [Satoshi 2008] argues that after some blocks have been buried deep enough, it is unnecessary to compute the Merkle tree from scratch as illustrated in the latter part of Figure 10.

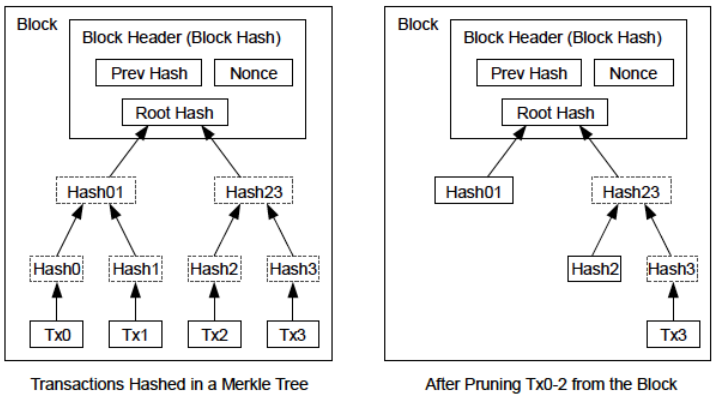


Fig. 10. Merkle tree[Satoshi 2008]

4.3. Timestamping:Relation to proof-of-work

The original paper makes a reference to [Back 2002] as the prototype for implementing a distributed crypto currency network like Bitcoin. Hashcash [Back 2002] was originally intended to be used as a throttling system for unwarranted use of internet resources like email spam. It basically consists of a client intending to take part in a protocol to fulfill certain computation tasks and generate a coin in order to be eligible for consideration by the server. It proposes the use of a cost function which is intended to be easily verifiable but expensive to compute. The function is dependent on the amount of work that needs to be done by the user to generate the token. The cost function is of the form:

$$\begin{cases} \Psi(): \text{The public hash function which is computationally hard} \\ \tau \leftarrow MINT(s,w): \text{The mint function generating the token} \\ \mu \leftarrow VALUE(\tau): \text{Token evaluation function} \end{cases}$$

The MINT function is used by the client to find a bit string commensurate with the challenge Ψ thrown by the server. It computes the token based on this and returns it back to the server. The server uses the VALUE function to check the veracity of this token.

Let us consider the following notation.

— $s \in \{0,1\}^*$ be a bitstring and $[s]_i$ be the left-most bit.
— $[s]_{i...j}$ be the substring from the i^{th} offset to the j^{th} offset.
— $=_b^{left}$ be a binary infix comparison operator where b is the length of the common left substring from both the bits strings.

Using these we can define the following two equations:

$$x =_0^{left} y, x \neq y \tag{2}$$

$$x =_b^{left} y, [x]_{1...b} = [y]_{1...b} \tag{3}$$

We now have all the information to calculate the required cost function

$$\begin{cases} \Psi(): \text{The public hash function} \\ \tau \leftarrow MINT(s,w): \textbf{find } x \in \{0,1\} \text{such that} \Psi(s,x) =_w^{left} 0^k, \textbf{return } x \\ \mu \leftarrow VALUE(\tau): \Psi(s,\tau) =_v^{left} 0^k, \textbf{return } v \end{cases}$$

In the Bitcoin network, the MINT and Value functions are implemented in a distributed way. With the collection of all transactions, each node attempts to solve the problem posed by Ψ. The first node which solves this problem gets to form the block of all transaction with the required $nonce$ value indicated in previous sections. The interesting thing to note is that a method originally intended to keep out intruders and prevent spam has been adopted seamlessly albeit with some modifications to facilitate monetary transactions in a safe way.

Moreover, each node attempts to form the block by taking in the transactions which have taken place since the last publicly known correct block. The nodes then make these transactions into a Merkle tree as illustrated before and then attempt to solve the $nonce$ problem. Another interesting thing to note is that the difficulty of the system can be changed by adjusting the number of bits which are being required for comparison. The Bitcoin network keeps increasing the number of bits and therefore keep increasing the level of difficulty and the computing resources required to solve the problem.

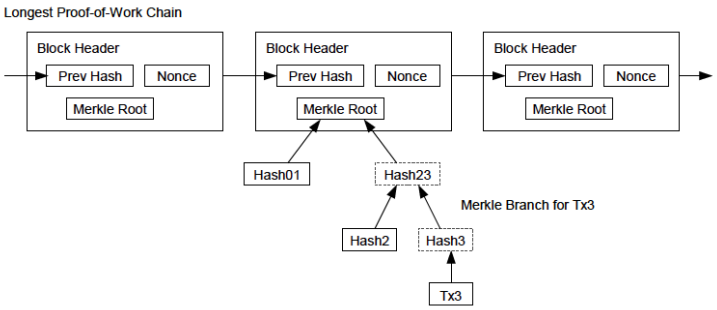


Fig. 11. Transaction verification[Satoshi 2008]

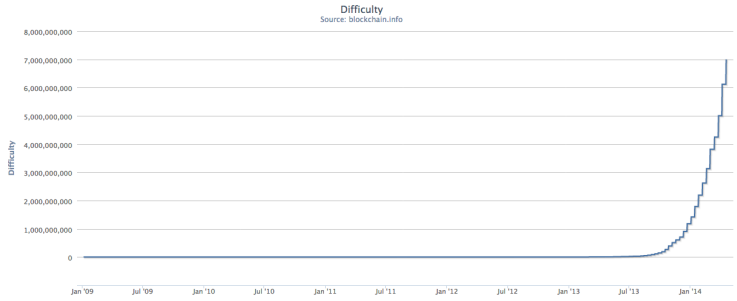
Figure 5.1 shows the longest proof of work chain. It is possible to verify payments in the Bitcoin network as illustrated in Figure 5.1. A user initially obtains the block header of the longest proof-of-work chain which can be easily obtained by querying the network. Then one can obtain the Merkle root by linking the transaction to the block its timestamped in. By linking the transaction to a particular place in the chain, it becomes obvious that a network node has accepted it and the blocks added after it can be used as proof that the network has accepted it. The authors in [Satoshi 2008] argue that as long as a majority of nodes in the network are honest nodes, the system will be fraud free as an attacker would have to be able to generate an even longer chain which is computationally impossible to achieve unless the majority of nodes are influenced to perpetrate the attack.

5. POOLING AND THE BITCOIN REWARD SYSTEM

We have seen in the previous sections how the Bitcoin network maintains privacy but yet at the same time has techniques which make manipulating the network fool-proof. However, the entire Bitcoin network is being run by nodes which contribute computing resources to this massive network. In order to incentivize the donation of computing resources, the network gives out a certain amount of Bitcoins for finding the nonce value correctly for a certain number of transactions. This is analogous to a credit card network charging the merchant a small fee for facilitating his transaction.

5.1. Reward System:Basics

Before proceeding further let us examine the concept of rewards and how they are calculated. As apparent in previous sections, the event relating to finding the block header i.e. the nonce value is a random event. A parameter known as difficulty D is set which is regulated by the network is set. The value D is chosen so that every computed hash will lead to a valid block with probability $\frac{1}{2^{32}D}$ [Rosenfeld 2011]. Thus a miner having a hash rate of h (the number of hash rates that he can compute per second) will be able to compute ht hashes in time t. This implies that on an average he will be able to find is $\frac{ht}{2^{32}D}$ of blocks. His expected payoff is thus $\frac{htB}{2^{32}D}$ where B is the reward for finding one block.



(a) Difficulty level since the beginning of the Bitcoin network

(b) Hashrate since the beginning of the Bitcoin network

5.2. Pooled mining versus solo mining

There are two techniques for going about mining for bit coins. One of them is solo mining and one of them is pooled mining. In the following subsections we will be discussing the importance of pooled and solo mining.

5.2.1. Solo mining. If mining for t time results in a $\frac{ht}{2^{32}D}$ blocks on average then we can say that $\lambda = \frac{ht}{2^{32}D}$ which is also the variance of the number of blocks found. This means that for the payout the following is the variance[Rosenfeld 2011]:

$$payout = \lambda B^2 = \frac{htB^2}{2^{32}D} \tag{4}$$

Now, the standard deviation as a fraction of expected payoff is as follows

$$\frac{\sqrt{\lambda B^2}}{\lambda B} = \frac{1}{sqrt\lambda} = \sqrt{\frac{2^{32}D}{ht}} \tag{5}$$

It can be observed that even with a decent amount of hardware, a node which is participating solo might have to wait for a long amount of time to get to any rewards. In fact, even after computing the resources for a long time, the node is at the same probability as the start that the next hash will yield the required block header. This is referred to as Gamblers fallacy.

5.2.2. Pooled mining. Pooled mining is a technique when a group of miners join together and collectively try to find the next block header. If we consider H as the hash rate of all the miners together, then total average reward is $\frac{Ht}{2^{32}D}$. An individual miners share is q the he will have qH of the share of the total hashes. This means his reward is $q\frac{Ht}{2^{32}D}$ which is nothing but $\frac{ht}{2^{32}D}$. However his payout variance is now very small $q\frac{htB^2}{2^{32}D}$, which means that The potential benefit to the miner is greater if the miner is small and the pool is large.

Pools are typically maintained by a pool operator who will take a fB cut of total rewards garnered by the pool and give $(1-f)B$ to the participating miners. The concept of pooling further involves the concept of shares. Shares can be considered as a precursor to blocks in the sense that they are hashes of a prospective block header which would have been accepted as a block if the difficulty was 1. Every hash determined by a miner has a probability of $\frac{1}{2^32}$ to being declared as a share. It so happens that finding blocks entails finding shares and finding shares entails finding blocks in terms of the amount of work done by miners. The shares found by a miner is therefore in proportion to the number of hashes which would have been found had the miner calculated with the intention of finding a block for the pool.

The probability that each share is a block is $\frac{1}{D}$. Therefore in expectation, a miner will be paid pB for discovering a share, if he goes solo to find that one share. Since the miner contributes an average of pB to the reward of the pool, in a fair pool, he receives pB for every share he submits after accounting for (1-f)pB as the operators cut.

5.3. Different pooling strategies

The paper [Rosenfeld 2011] also lists out different pooling strategies or the ways in which the reward distribution takes place. The various reward systems are briefly explained in the following sections.

5.3.1. Simple reward System. This technique has two categories

— *Proportional reward scheme* In this system payments are calculated on the basis of rounds which is nothing but a time slot in which a block is found. At the end of a round when a pool receives a reward of B, the operator keeps a fee fB and distributes (1-f)B among the miners in direct proportions to the number of shares he contributed during the particular round. The payout strategy for this is that assuming that the miner contributed n shares in a particular round with total shares submitted by all other miners totaling to N, then he can expect to be paid $\frac{n}{N}(1-f)B$ for that particular round.

— *Pay-Per-Share* In this category, the operator outrightly proportionately rewards the miner for each share contributed without waiting for the block to be found. The operator in turn then collects all the rewards obtained by getting the correct block. In doing so he absorbs the variance associated with the rewards for miners. The advantage in this approach also lies in the fact that the miners do not have to wait for a block to be accepted.

5.3.2. Score Based methods. This technique has the following categories.

— *Slush system* This is a type in which miners are rewarded from the time they take to give out a share from the moment the round starts. Based on this aspect a score

is assigned to the share. It is obvious that this depends on the time tha has passed since the beginning of the round and greater the time taken the higher the score. Based on these characteristics a score is assigned. On the other hand in the proportional technique shares submitted early are naturally worth more than late shares, this scoring technique is intended to counter that effect. The scoring function is exponential, where $s = e^{\frac{T}{C}}$ where s is the score given for a share submitted at time T, and C is a constant.

— *Geometric System* This system has two types of fees, a fixed fee and a variable fee. The fixed fee is nothing but a constant amount taken from the reward of every block. The variable fee is a score generated at the start of the round by the pool operator and reduces exponentially with passage of time. This directly means that the longer the pool takes to solve the block, the lesser this reward becomes.

— *Pay per last n shares* This concept does away the concept of rounds. It instead rewards miners who have submitted the last N shares proportionately to the amount of rewards that were accumulated by the pool in that period.

In [Babaioff et al. 2012] some there techniques of rewards have been discussed which improve upon the reward mechanism and attempt to lessen the payment overhead plaguing current techniques. Thus in this section we have analyzed pooling techniques which give a huge advantage to the user in terms of maximizing his returns by joining in with other miners. The next section talks about some of the disadvantages of the current Bitcoin network and also suggests certain alternatives which have been proposed.

6. DISADVANTAGES AND ALTERNATIVES

As we have seen in the previous sections, the crux of the whole Bitcoin network lies in the finding of a block by nodes by solving a computationally hard problems. This requires intense use of computing resources which consume significant amount of resources in terms of electricity, CPU time and ultimately money. Another aspect of Bitcoin network is the apparent increase in difficulty level as the network becomes popular leading to more nodes joining the network introducing greater competition to existing nodes. All these aspects of the network have disadvantages of their own as illustrated below:

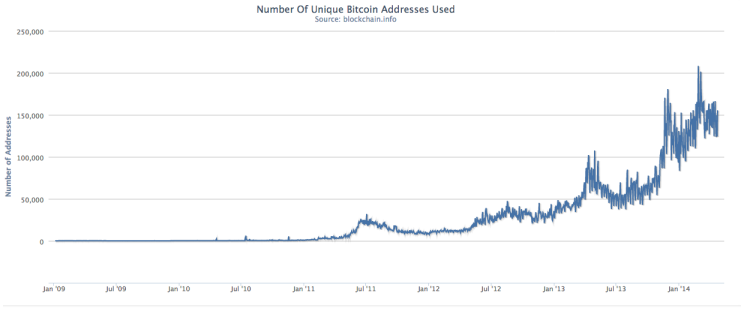


Fig. 12. Timeline diagram regarding block discovery and propagation

— The rapid increase in popularity of Bitcoins has led to a high number of nodes joining the network in a very short time competing to solve for blocks. Figure 12 shows the nearly exponential increase in unique addresses in the recent past. This adversely affects the prospects of simple miners.

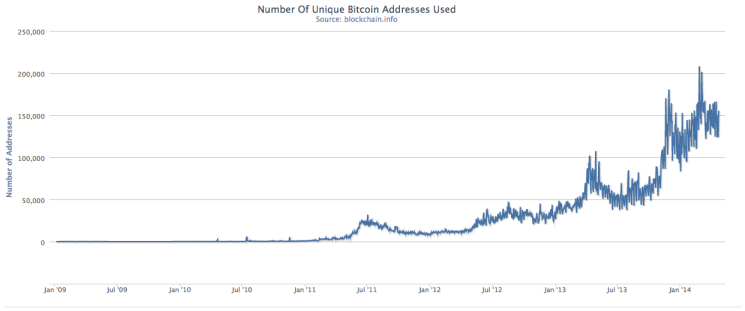


Fig. 13. Timeline diagram regarding block discovery and propagation

— The large number of nodes has meant that the revenue for miners has gradually started to decrease after seeing an initial spike as shown in Figure 13.
— As of the writing of this paper, the general trend in mining has become an exclusive hegemony of miners owning sophisticated hardware like ASIC(Application Specific Integrated Circuits) miners which typically mine at around 600GigaHashes/sec. Earlier GPU mining was considered faster and would guarantee a respectable profit to miners in terms of setup costs like electricity and CPU time. However with the advent of ASIC miners it is now virtually impossible to beat their speed using normal CPU or GPU mining.
— Privacy has also been an issue because as more miners accumulate more Bitcoins, there have been more number of widespread attempts to steal Bitcoins from miners' wallets.

As a result of all these factors many alternatives like Memcoin and Litecoin [Mackenzie 2010] have been proposed. These approaches use a sequential memory hard scheme which require more memory than normal schemes mentioned in previous sections. Another alternative known as Zerocoin [Miers et al. 2013]has been proposed which is an extension to the Bitcoin network and seeks to fully anonymize Bitcoin transaction without significantly altering the network internals.

7. CONCLUSION AND FUTURE SCOPE

In this paper we have attempted to anaylze the various parallel computing aspects of the Bitcoin network. The paper initially starts out with analyzing the key concepts of Bitcoin, its motivation and objective. It then proceeds to explain the various terms like Blocks, Block chains and transactions in the Bitcoin network. It then delves into the mechanism of how the Bitcoin network relying solely on volunteer nodes is able to guarantee authenticity, integrity and non repudiation in the network in a distributed way using the concept of public key cryptography. We have also described in sufficient detail how the Bitcoin network implements timestamping in order to avoid the concept of double spending and fraud. This paper also describes trends in information propagation in the Bitcoin network and provides insight into the same. Paper also examines the pooling strategy adopted in the network which attempts to maximize the gains by individual miners by reducing their payout variance. Finally the paper briefly talks about the various challenges and disadvantages existing in the Bitcoin network and also describes alternatives in which have appeared in recent literature which attempt to address the same. The Bitcoin network is fast proving to be a useful tool for monetary transactions but the entire concept hinges on certain key aspects like block finding. There have been speculations that with the advent of quantum computing, the problem of block finding will soon become immensely more easier and might lead

to the demise of the network. However, as per current technological trends, the Bitcoin network seems to be growing in popularity by the day.

REFERENCES

Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. 2012. On Bitcoin and Red Balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12)*. ACM, New York, NY, USA, 56–73. DOI:http://dx.doi.org/10.1145/2229012.2229022

Adam Back. 2002. Hashcash - A Denial of Service Counter-Measure. (2002).

C. Decker and R. Wattenhofer. 2013. Information propagation in the Bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*. 1–10. DOI:http://dx.doi.org/10.1109/P2P.2013.6688704

Dorothy E. Denning. 1984. Digital Signatures with RSA and Other Public-key Cryptosystems. *Commun. ACM* 27, 4 (April 1984), 388–392. DOI:http://dx.doi.org/10.1145/358027.358052

W. Diffie and M.E. Hellman. 1979. Privacy and authentication: An introduction to cryptography. *Proc. IEEE* 67, 3 (March 1979), 397–427. DOI:http://dx.doi.org/10.1109/PROC.1979.11256

Eiichiro Fujisaki and Tatsuaki Okamoto. 1999. How to Enhance the Security of Public-Key Encryption at Minimum Cost. In *Public Key Cryptography*. Lecture Notes in Computer Science, Vol. 1560. Springer Berlin Heidelberg, 53–68. DOI:http://dx.doi.org/10.1007/3-540-49162-7_5

Stuart Haber and W. Scott Stornetta. 1991. How to Time-stamp a Digital Document. *Journal of Cryptology* 3 (1991), 99–111.

Stuart Haber and W. Scott Stornetta. 1997. Secure Names for Bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security (CCS '97)*. ACM, New York, NY, USA, 28–35. DOI:http://dx.doi.org/10.1145/266420.266430

Adam Mackenzie. 2010. MEMCOIN2: A HYBRID PROOF OF WORK/PROOF OF STAKE CRYPTOCURRENCY. (2010).

H. Massias, X. Serret Avila, and J.-J. Quisquater. 1999. Design Of A Secure Timestamping Service With Minimal Trust Requirement. In *the 20th Symposium on Information Theory in the Benelux*.

Ralph C. Merkle. 1980. Protocols for Public Key Cryptosystems. *2012 IEEE Symposium on Security and Privacy* 0 (1980), 122. DOI:http://dx.doi.org/10.1109/SP.1980.10006

Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. 2013. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy (SP '13)*. IEEE Computer Society, Washington, DC, USA, 397–411. DOI:http://dx.doi.org/10.1109/SP.2013.34

Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. 1998. EPOC: Efficient Probabilistic Public-Key Encryption. In *IEEE P1363a*.

R. L. Rivest, A. Shamir, and L. Adleman. 1978. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM* 21, 2 (Feb. 1978), 120–126. DOI:http://dx.doi.org/10.1145/359340.359342

Dorit Ron and Adi Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*, Ahmad-Reza Sadeghi (Ed.). Lecture Notes in Computer Science, Vol. 7859. Springer Berlin Heidelberg, 6–24. DOI:http://dx.doi.org/10.1007/978-3-642-39884-1_2

Meni Rosenfeld. 2011. Analysis of Bitcoin Pooled Mining Reward Systems. *CoRR* abs/1112.4980 (2011).

Satoshi. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. (2008).

Bruce Schneier. 1993. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA.

P. Singh, B.R. Chandavarkar, S. Arora, and N. Agrawal. 2013. Performance Comparison of Executing Fast Transactions in Bitcoin Network Using Verifiable Code Execution. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*. 193–198. DOI:http://dx.doi.org/10.1109/ADCONS.2013.42

Appendix A: Detailed Comments and Classification scheme

1 Introduction

1.1 What is a bitcoin?

It is nothing but a chain of transactions. It is a certificate which states what all transactions have taken place using this same bitcoin previously. The bearer of one such bitcoin can exchange this in terms of actual real world currency. Why is a bitcoin so popular nowadays? It works completely on a peer2peer network which makes regulation nearly impossible. It is unlike any other real world currency because it is immune to regulation. No real world government or institution can claim ownership of the entire network or concept. It is entirely market regulated and depends on the basic economic principle of demand and supply. It is a cryptocurrency and using an unprecedented amount of parallel computing technique, the entire network is able to ensure fraud proof operation. In other words it is nearly impossible to fool the network and peddle ones own fake certificates or fake bitcoins posing as the real ones in the network.

1.2 How does this work?

The cornerstone of the entire bitcoin network is a transaction. A transaction is nothing but transfer of a bitcoin from one owner to another. There could be a few potential issues with this. One is that somewhere down the line when the bitcoin has changed many owners, a malicious owner might try to double spend the bitcoin. In other words, he might try to sell the same bitcoin to two different owners at the same time. Obviously such a transaction should be illegal in the system as it has the potential to throw the entire network out of gear. To solve it, the bitcoin introduces the concept of block chain which in simple terms is a record of all previous transactions. Therefore when a transaction is about to happen, one can easily check

previous records to make sure there has been no double spending. This is done using a distributed timestamp server which is based on a peer to peer network, which would provide the necessary proof of previous transactions when required. An interesting aspect is provided in [15] which gives a good overview of Bitcoin user behavior and general quantitative characteristics of the Bitcoin network. On a related note, [19] propose an extension to the Bitcoin network that works on mutual trust but promises faster transaction time by moving it from a Pioneer model to a mutual trust based model.

1.3 Transactions

Each owner has his own public key and a private key. In a nutshell what happens when a bitcoin is passed on to another owner is that a new record is created in the block chain which has the hash of the previous transaction and the new owners public key as inputs. The result is appended to the end of the bitcoin thereby completing the transactions. But problem of double spending still persists. What if the owner, were to make a digital copy and use it twice. There must be a way to ensure that the previous owners did not sign any earlier transactions. Since the earliest transaction is the one that counts,later attempts to double spend can be neglected. If we are aware of all transactions we can be reasonably clear about the absence of any double spending operations. The mint is aware of all transactions and decides which arrived first. In order to completely do away with the concept of a thrid party validator all transactions must be publicly announced. This implies all participants must agree on a single history. To address this problem the requirement of a central mint is necessary. This mint will have the task of verifying each transaction and validating its correctness thereby preventing fraud. Mint is being implemented in a peer-2-peer model. But central mint is giving too much

Solution will be to establish a timestamp server. Each transaction will be timestamped. It is a unique way of making sure the amount of knowledge that was present at that exact point of time. This will help check whether any of the previous owners signed any of the previous documents. Each new transaction will take into account the previous timestamp and will include it in the hash to create the new timestamp for the new transaction.

Paper on [2] is taken as the backbone for implementing such a network. The technique of proof of work has been explained in detail in that work and is the basis for a similar technique being implemented in the bitcoin network too. Each block has a field called nonce. It is important to note that each transaction spawns a new block. The idea of the proof of work is that each block must hash and generate only

a certain number of zeroes. By manipulating the nonce field and by trial and error in an increasing fashion, a nonce value will be found with will give upon hashing a certain number of bits to the block. This will form the block. When after each successive transaction, the chain keeps getting augmented with new blocks with different nonce values depending on the nonce value of not just the current block but all previous blocks because as can be noted, each transaction also depends on previous transactions. The advantage of this process is that if an attacker were to manipulate the network and try to insert a wrong value, he would have to change the nonce of the entire chain henceforth which would prove to be computationally impractical.

The paper then talks about how unlikely and computationally hard it is for an attacker to replicate the same chain and keep pace with the network with intent to destroy it successfully. Since it is not in the interest of a parallel technique we can skip this part.

Paper further talks about incentivizing those nodes that lend the CPU time and electricity for facilatating the transactions and being part of the peer to peer network by generating new bitcoins for such nodes. The logic behind this is to prempt any attacker from using his computing powers to attack the system by offering him an even more profitable opportunity by help being part of the network and miniting new bit coins.

1.4 Block and block chain

The block is nothing but a transaction. Block chain is nothing but a public ledger which keeps track of all transaction in the bit coin network. *The blockchain is a public ledger of all transactions in the Bitcoin network. Blockchain.info allows you to navigate the bitcoin blockchain.*[3]. Thus when a transaction is performed the node handling the transaction must propagate this information and make sure that it is committed to other copies of the ledger. How this is done is explained in the next work.

2 Information Propagation in bitcoin network

[3] talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv message stating that it has these many blocks corresponding

to some transactions. Any node not having the said information responds back and transfer takes place. Paper talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv essage stating that it has these many blocks corresponding to some transactions. Any node not having the said information responds back and transfer takes place. This is done to save bandwidth as the information to be exchanged is of considerable size. The paper talks about blockchain forks and how they are created. Due to the concept of proof-of-work the valid blocks are to be found independently at random. The proof-of-work causes valid blocks to be found independently at random.

2.1 Where is parallel computing involved?

Starting from the beginning this is the sequence of events which goes on. An entity A wants to send BTCs to another entity B, and this is referred to as a transaction. This transaction typically is validated by using the techniques described in the previous sections. This transaction is *heard* by one of the nodes which picks it up creates a *block* and tries to publish it publicly i.e. to all other nodes in the network. But there are other such transactions being handled by other nodes simultaneously.The problem now is that how will a node decide which transaction happened first- the one which is represented by the block it received or the one that happened under its own watch. It is important to note that to keep the copy of the ledger fresh, only the latest transactions ordered in the correct chronological order must be put in. To do this, each node tentatively commits the transactions which it has knowledge of. If an earlier transaction is received as a broadcast from some other node then it is supposed to roll back the commit, put the latest information in and then push the remaining information in. This step though it looks simple actually is a bit more complicated than that. This problem refers to solve the proof of work problem referred to above. To determine the order the nodes attempt to find a solution to a proof-of-work. The proof-of-work consists in finding a byte string, called nonce, as illustrated in the previous section, that when combined with the block header has to yield a hash with a given number of zeroes. This problem is actually a computationally hard problem since cryptographic hash functions are one-way functions. To find the actual nonce string we have to analyze all possibilities which would give us the correct value. Once we obtain the correct value, it is easy to verify its correctness. The node finding the nonce value first will then send the block after embedding it with the nonce value to

all other nodes. Since it is easy to verify the authenticity, they will then accept or reject the solution and accordingly make changes to their own copies of the ledger. This aspect of the network is what is the most powerful feature of the network.

2.2 Blockchain

Having covered the concept of blocks and transactions, we now come to the concept of block chains. Blockchains are nothing but a directed tree with individual blocks for nodes with the latest node being referred to as the block chain head. The height of the tree is referred to as h. There can be a situation in which different blockchain heads can exist at different heights. In such a scenario, if a node receives a block with a height greater than the block chain height of its own ledger copy, there can be two cases, one in which the existing block chain head is an ancestor of the received block or when its not. If it is the ancestor, it is obvious that the node has missed out on some transactions in between and will get the required information from other nodes and attempt to keep its copy of the ledger fresh. If however it is not an ancestor, it is clear that, both share a common ancestor, and the node will then find refreh the list starting from the nearest common ancestor to keep its copy fresh.

3 Keys:Maintaining Privacy in a distributed setting

A necessary aspect in the scenario for Bitcoins is the need to maintain privacy but at the same time ensuring that safety in the transaction. This is particularly important because we need to maintain the principles of authenticity,integrity and non repudiation while implementing the Bitcoin network. Authenticity is the ability to identify the sender of the message. Integrity implies that the message being sent must not be compromised by a malicious third party. Any person in the middle must not be able to significantly alter or replace a part of the message being sent. Non repudiation means that the sender must not be able to refute the sending of the message sometime in the future after it has been received by the receiver [18] . It is in this regard that cryptographic concepts of hash function is used in this network. hash functions provide the much needed support in the Bitcoin network with the use of two sets of keys i.e. the public and private keys. These keys accomplish the dual purpose of concealing the identity of the individual involved in the transaction while ensuring the safety of the transactions. Since the ledger is public and scattered across a p2p network, this is accomplished among the many nodes in

a distributed way. It is vital to the network to avoid fraud or theft of coins. This section focusses on methods which ensure authenticity,integrity and non-repudiation in a Bitcoin transaction between two interested parties without the involvement of any third entity.

3.1 Public key cryptography

Public keys also serves as a method for verifying the authenticity and the integrity because with the public key one can decrypt the signature and and compare the result with the hash of the message. This also implements non repudiation as the sender is not able to falsely deny the sending of the message. The paper [5] provides more in-depth detail about public key encryption systems and the motivation behind it and is considered a seminal paper in the said domain.

3.2 Hashing:A basic overview

If two parties want to send or receive messages, they can use encryption to hide the messages. The receiver can then perform what is referred to as decryption to recover the original message. It is often the case that the algorithms for encryption and decryption are well known and the receiver can recover the original message using what is referred to as a key. If the key is the same as the one used for encryption, it is referred to as a *symmetric encryption*. The advantage of symmetric algorithms lies in the aspect of confidentiality, but the process for maintaining a common key between sender and receiver is often cumbersome. This is the motivation for use of hash functions. A hash function akes a variable-length input string referred to as a pre-image and converts it to a fixed-length output string called a hash value.[18].

3.3 Protocols for Public key encryption

The paper [11] provides some established protocols which deal with exchange of keys in a distributed environment. In [11] is highlighted several techniques relating to key distribution. The first one among them is the centralized key distribution technique which employs a central entity or a distribution center which serves as a repository for all the agents to deposit their respective keys. If any two agents wish to communicate with each other they contact the distribution center and obtain the keys. In case of the Bitcoin network however, such an approach is not feasible because, the entire network design being a peer-2-peer one. he work presented in [14] gives a detailed description of methods for obtaining digital signatures and public-key crypto systems.

It proposes a scheme which involves an exponential rate hash function. [4] presents another work which talks about vulnerabilities in public key crypto systems and goes on to describe a method to foil attacks arising out of such vulnerabilities. The author argues that an attacker can get the victim to sign new messages derived by intercepting messages to the victim and then forging the signature of the victim. [13] and [6] provide description of various types of probabilistic cryptographic techniques and an approach to enhance security in public key encryption systems by preventing cipher text attacks.

4 Timestamping:An extension to proof-of-work

In Section 1.3 we provided an overview of the timestamping process which goes on in the network. In this section we deal with a deeper analysis of the concept of timestamping and its relevance to the Bitcoin network. This section describes approaches regarding timestamping in situations wherein the trust factor is distributed i.e. in cases where there is no centralized authority to guarantee trustworthiness of a transaction.

4.1 Why is it a necessity?

Timestamping basically is a measure to prevent double spending. By timestamping a certain transaction we guarantee the temporal aspect of the transaction. In the Bitcoin network, it is also essential to keep track of previous transactions too to instill legitimacy in the transaction. This section talks about some approaches which can be used for timestamping as referred to by [17].

4.2 Timestamping: A general outline

[10] [8] In a system where there is a centralized system of authority, [10] propose a technique wherein a timestamping scheme following the binary tree structure elaborated in [8] is used and an improved scheme with minimum trust requirements. One method to link the present timestamp using an appropriate hash function to the previously found timestamp. As this chain grows, it becomes increasingly difficult to for any attacker to forge timestamps by manipulating the bit-strings[7].

4.3 Timestamping:Relation to proof-of-work

The original paper makes a reference to [2] as the prototype for implementing a distributed crypto currency network like Bitcoin. Hashcash [2] was originally intended to be used as a throttling system for unwarranted use of internet resources like email spam. It basically consists of a client intending to take part in a protocol to fulfill certain computation tasks and generate a coin in order to be eligible for consideration by the server. It proposes the use of a cost function which is intended to be easily verifiable but expensive to compute.

5 Pooling and the Bitcoin Reward system

The paper [16] also lists out different pooling strategies or the ways in which the reward distribution takes place. There are two techniques for going about mining for bit coins. One of them is solo mining and one of them is pooled mining. In the following subsections we will be discussing the importance of pooled and solo mining.

5.0.1 Solo mining

If mining for t time results in a $\frac{ht}{2^{32}D}$ blocks on average then we can say that $\lambda = \frac{ht}{2^{32}D}$ which is also the variance of the number of blocks found. This means that for the payout the following is the variance[16]:

5.0.2 Pooled mining

Pooled mining is a technique when a group of miners join together and collectively try to find the next block header. If we consider H as the hash rate of all the miners together, then total average reward is $\frac{Ht}{2^{32}D}$. An individual miners share is q the he will have qH of the share of the total hashes. This means his reward is $q\frac{Ht}{2^{32}D}$ which is nothing but $\frac{ht}{2^{32}D}$. However his payout variance is now very small $q\frac{htB^2}{2^{32}D}$, which means that The potential benefit to the miner is greater if the miner is small and the pool is large.

In [1] some there techniques of rewards have been discussed which improve upon the reward mechanism and attempt to lessen the payment overhead plaguing current techniques.

6 Disadvantages and alternatives

As we have seen in the previous sections, the crux of the whole Bitcoin network lies in the finding of a block by nodes by solving a computationally hard problems. This requires intense use of computing resources which consume significant amount of resources in terms of electricity, CPU time and ultimately money. Another aspect of Bitcoin network is the apparent increase in difficulty level as the network becomes popular leading to more nodes joining the network introducing greater competition to existing nodes.

As a result many alternatives like Memcoin and Litecoin [9] have been proposed. These approaches use a sequential memory hard scheme which require more memory than normal schemes mentioned in previous sections. Another alternative known as Zerocoin [12]has been proposed which is an extension to the Bitcoin network and seeks to fully anonymize Bitcoin transaction without significantly altering the network internals.

7 Conclusion and Future Scope

References

[1] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On bitcoin and red balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 56–73, New York, NY, USA, 2012. ACM.

[2] Adam Back. Hashcash - a denial of service counter-measure. 2002.

[3] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10, Sept 2013.

[4] Dorothy E. Denning. Digital signatures with rsa and other public-key cryptosystems. *Commun. ACM*, 27(4):388–392, April 1984.

[5] W. Diffie and M.E. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, March 1979.

[6] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer Berlin Heidelberg, 1999.

[7] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

[8] Stuart Haber and W. Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 28–35, New York, NY, USA, 1997. ACM.

[9] Adam Mackenzie. Memcoin2: A hybrid proof of work/proof of stake cryptocurrency. 2010.

[10] H. Massias, X. Serret Avila, and J.-J. Quisquater. Design of a secure timestamping service with minimal trust requirement. In *the 20th Symposium on Information Theory in the Benelux*, 1999.

[11] Ralph C. Merkle. Protocols for public key cryptosystems. *2012 IEEE Symposium on Security and Privacy*, 0:122, 1980.

[12] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 397–411, Washington, DC, USA, 2013. IEEE Computer Society.

[13] Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. Epoc: Efficient probabilistic public-key encryption. In *IEEE P1363a*, 1998.

[14] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[15] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 6–24. Springer Berlin Heidelberg, 2013.

[16] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[17] Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[18] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[19] P. Singh, B.R. Chandavarkar, S. Arora, and N. Agrawal. Performance comparison of executing fast transactions in bitcoin network using verifiable code execution. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 193–198, Dec 2013.

Appendix B: Annotated Bibliography

Paritosh P. Ramanan

The system of a crypto currency has been suggested [16] which attempts to restrict the transaction only between two responsible parties and without depending on trust in facilitating the transaction. Cryptocurrency like Bitcoin works on the basis of cryptographic proof instead of trust. It removes the concept of the middle third party by directly engaging the buyer and seller. The premise of the crytocurrency is to develop a technique wherein a transaction once made is computationally too impractical to reverse. Cryptocurrency is analogous to legal tender or hard currency in the digital world. The analogy stems from the fact that like legal tender which once used in a transaction cannot be reversed theoretically. The purpose of Bitcoin is to replicate this mechanism prevalent in case of legal tender in the electronic domain which would allow for a seamless transaction of money between two individual removing the issues of trust and other logistical disadvantages.

It works completely on a peer2peer network which makes regulation nearly impossible. It is unlike any other real world currency because it is immune to regulation. No real world government or institution can claim ownership of the entire network or concept. It is entirely market regulated and depends on the basic economic principle of demand and supply. It is a cryptocurrency and using an unprecedented amount of parallel computing technique, the entire network is able to ensure fraud proof operation. In other words it is nearly impossible to fool the network and peddle ones own fake certificates or fake bitcoins posing as the real ones in the network. An interesting aspect is provided in [14] which gives a good overview of Bitcoin user behavior and general quantitative characteristics of the Bitcoin network.

Since the earliest transaction is the one that needs to be examined, later attempts to double spend can be neglected. If we are aware of all transactions we can be reasonably clear about the absence of any double spending operations. The mint is aware of all transactions and decides which arrived first. In order to completely do away with the concept of a thrid party validator all transactions must be publicly announced. This implies all participants must agree on a single history. On a related note, [18] propose an extension to the Bitcoin network that works on mutual trust but promises faster transaction time by moving it from a Pioneer model to a mutual trust based model.

Paper on [1] is taken as the backbone for implementing such a network. The technique of proof of work has been explained in detail in that work and is the basis for a similar technique being implemented in the bitcoin network too. Each block has a field called

nonce. It is important to note that each transaction spawns a new block. The idea of the proof of work is that each block must hash and generate only a certain number of zeroes. By manipulating the nonce field and by trial and error in an increasing fashion, a nonce value will be found with will give upon hashing a certain number of bits to the block. This will form the block. When after each successive transaction, the chain keeps getting augmented with new blocks with different nonce values depending on the nonce value of not just the current block but all previous blocks because as can be noted, each transaction also depends on previous transactions. The advantage of this process is that if an attacker were to manipulate the network and try to insert a wrong value, he would have to change the nonce of the entire chain henceforth which would prove to be computationally impractical.

[2] talks about various concepts like transactions, blocks and block chain that are being used in the bitcoin network. The work describes the implementation details of the p2p network. Since the purpose is to keep updating and synchronizing the ledger replicas, the relevant entities are the transaction and the block chain. Thus each node advertises with the inv message stating that it has these many blocks corresponding to some transactions. Any node not having the said information responds back and transfer takes place. This is done to save bandwidth as the information to be exchanged is of considerable size.

Authenticity is the ability to identify the sender of the message. Integrity implies that the message being sent must not be compromised by a malicious third party. Any person in the middle must not be able to significantly alter or replace a part of the message being sent. Non repudiation means that the sender must not be able to refute the sending of the message sometime in the future after it has been received by the receiver [17] . It is in this regard that cryptographic concepts of hash function is used in this network. hash functions provide the much needed support in the Bitcoin network with the use of two sets of keys i.e. the public and private keys. These keys accomplish the dual purpose of concealing the identity of the individual involved in the transaction while ensuring the safety of the transactions. The paper [4] provides more in-depth detail about public key encryption systems and the motivation behind it and is considered a seminal paper in the said domain. The paper [10] provides some established protocols which deal with exchange of keys in a distributed environment. A hash function akes a variable-length input string referred to as a pre-image and converts it to a fixed-length output string called a hash value.[17]. The work presented in [13] gives a detailed description of methods for obtaining digital signatures and public-key crypto systems. It proposes a scheme which involves an exponential rate hash function. [3] presents another work which talks about vulnerabilities in public key crypto systems and goes on to describe a method to foil attacks arising out of such vulnerabilities. The author argues that an attacker can get the victim to sign new messages derived by intercepting messages to the victim and then forging the signature of the victim. [12] and [5] provide description of various types of probabilistic cryptographic techniques and an approach to enhance security in public key encryption systems by preventing cipher text attacks.

There are certain conditions that need to be implemented to achieve a stable system

of verifying against forgery of timestamping. Thus, there have been recent attempts to develop timestamping schemes which will be computationally very hard to fake. One such method is to link the present timestamp using an appropriate hash function to the previously found timestamp. As this chain grows, it becomes increasingly difficult to for any attacker to forge timestamps by manipulating the bit-strings[6].

The timestamping scheme being used in the Bitcoin network is inspired from the works [7] and [9].

The timestamp of the document now contains all the values necessary to rebuild the entire tree. For instance, for y_3 the timestamp is $\{(y_3, L), (H_12, L), (H_58, R), (RH_i - 1, L)\}$. It is tone noted that this notation basically comprises of the left or the right sibling denoted by letter R or L and their hash values respectively. The general idea is that for verification purposes the "Round value" is obtaining by systematically rebuilding the tree using the above mentioned information. The binary tree structure we are using in this method is referred to as the Merkle tree[10].

A user initially obtains the block header of the longest proof-of-work chain which can be easily obtained by querying the network. Then one can obtain the Merkle root by linking the transaction to the block its timestamped in. By linking the transaction to a particular place in the chain, it becomes obvious that a network node has accepted it and the blocks added after it can be used as proof that the network has accepted it. The authors in [16] argue that as long as a majority of nodes in the network are honest nodes, the system will be fraud free as an attacker would have to be able to generate an even longer chain which is computationally impossible to achieve unless the majority of nodes are influenced to perpetrate the attack.

The paper [15] also lists out different pooling strategies or the ways in which the reward distribution takes place. There are two techniques for going about mining for bit coins. One of them is solo mining and one of them is pooled mining. In the following subsections we will be discussing the importance of pooled and solo mining. In [**?**] some there techniques of rewards have been discussed which improve upon the reward mechanism and attempt to lessen the payment overhead plaguing current techniques.

As a result of all these factors many alternatives like Memcoin and Litecoin [8] have been proposed. These approaches use a sequential memory hard scheme which require more memory than normal schemes mentioned in previous sections. Another alternative known as Zerocoin [11]has been proposed which is an extension to the Bitcoin network and seeks to fully anonymize Bitcoin transaction without significantly altering the network internals.

References

[1] Adam Back. Hashcash - a denial of service counter-measure. 2002.

[2] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In

Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on, pages 1–10, Sept 2013.

[3] Dorothy E. Denning. Digital signatures with rsa and other public-key cryptosystems. *Commun. ACM*, 27(4):388–392, April 1984.

[4] W. Diffie and M.E. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of the IEEE*, 67(3):397–427, March 1979.

[5] Eiichiro Fujisaki and Tatsuaki Okamoto. How to enhance the security of public-key encryption at minimum cost. In *Public Key Cryptography*, volume 1560 of *Lecture Notes in Computer Science*, pages 53–68. Springer Berlin Heidelberg, 1999.

[6] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3:99–111, 1991.

[7] Stuart Haber and W. Scott Stornetta. Secure names for bit-strings. In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, CCS '97, pages 28–35, New York, NY, USA, 1997. ACM.

[8] Adam Mackenzie. Memcoin2: A hybrid proof of work/proof of stake cryptocurrency. 2010.

[9] H. Massias, X. Serret Avila, and J.-J. Quisquater. Design of a secure timestamping service with minimal trust requirement. In *the 20th Symposium on Information Theory in the Benelux*, 1999.

[10] Ralph C. Merkle. Protocols for public key cryptosystems. *2012 IEEE Symposium on Security and Privacy*, 0:122, 1980.

[11] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 397–411, Washington, DC, USA, 2013. IEEE Computer Society.

[12] Tatsuaki Okamoto, Shigenori Uchiyama, and Eiichiro Fujisaki. Epoc: Efficient probabilistic public-key encryption. In *IEEE P1363a*, 1998.

[13] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[14] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, volume 7859 of *Lecture Notes in Computer Science*, pages 6–24. Springer Berlin Heidelberg, 2013.

[15] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[16] Satoshi. Bitcoin: A peer-to-peer electronic cash system. 2008.

[17] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[18] P. Singh, B.R. Chandavarkar, S. Arora, and N. Agrawal. Performance comparison of executing fast transactions in bitcoin network using verifiable code execution. In *Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on*, pages 193–198, Dec 2013.