Fall 2015

SHAREDWEALTH: A CRYPTOCURRENCY TO REWARD MINERS EVENLY

Siddiq Ahmed Syed
San Jose State University

SHAREDWEALTH: A CRYPTOCURRENCY TO REWARD MINERS EVENLY

A Thesis
Presented to
The Faculty of the Department of Computer Science
San Jose State University

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Siddiq Ahmed Syed
December 2015

The Designated Thesis Committee Approves the Thesis Titled

SHAREDWEALTH: A CRYPTOCURRENCY TO REWARD MINERS EVENLY

By

Siddiq Ahmed Syed

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2015

Dr. Thomas Austin	Department of Computer Science
Dr. Katerina Potika	Department of Computer Science
Mr. Ronald Mak	Department of Computer Science

Contents

Abstract

Bitcoin [19] is a decentralized cryptocurrency that has recently gained popularity and has emerged as a popular medium of exchange. The total market capitalization is around 1.5 billion US dollars as of October 2013 [28]. All the operations of Bitcoin are maintained in a distributed public global ledger known as a block chain which consists of all the successful transactions that have ever taken place. The security of a block chain is maintained by a chain of cryptographic puzzles solved by participants called miners, who in return are rewarded with bitcoins. To be successful, the miner has to put in his resources to solve the cryptographic puzzle (also known as a proof of work). The reward structure is an incentive for miners to contribute their computational resources and is also essential to the currency's decentralized nature.

One disadvantage of the reward structure is that the payment system is uneven. The reward is always given to one person. Hence people form mining pools where every member of the pool solves the same cryptographic puzzle and irrespective of the person who solved it, the reward is shared evenly among all the members of the pool. The Bitcoin protocol assumes that the miners are honest and they follow the Bitcoin protocol as prescribed. If group of selfish miners comes to lead by forming pools, the currency stops being decentralized and comes under the control of the selfish miners. Such miners can control the whole Bitcoin network [29]. Our goal is to address this problem by creating a distinct peer-to-peer protocol that reduces the incentives for the miners to join large mining pools. The central idea is to pay the “runners-up” who come close to finding a proof, thereby creating a less volatile payout situation. The work done by the “runners-up” can be used by other miners to find the solution of proof of work by building upon their work. Once they find the actual solution they have to include the solution of the other miner in order to get rewarded. The benefit of this protocol is that not only the miners save their computational resources but also the reward is distributed among the miners.

I. Introduction

The Bitcoin protocol assumes that miners who put in their computational resources to solve cryptographic puzzles are honest [29]. If a group of greedy miners comes into power, then the decentralized nature of the currency stops and comes under the control of these miners. The miners then can restrict transactions and only allow transactions that benefit them.

The idea of Bitcoin reflects the proposition that is usually associated with cryptography. T. Courtois et al. [30] call it a cryptographer's dream: *a dream about the world which functions with participants which do not see each other, do not trust each other a lot, and yet are able to somewhat function and achieve some sort of "secure function" or prevent fraud from being committed. An attempt to build systems which remove the necessity of having trusted parties such as financial institutions and other businesses, intermediaries, or providers of services. Or at the very least, to greatly decrease the trust assumptions which are necessary*.

The critical part of Bitcoin is *miners*. They are responsible for approving and verifying the bitcoin transaction. Apart from miners, no other user needs to verify the transactions except those who are associated with the transactions. All the transactions are available in the public ledger called a block chain, which miners use to verify the correctness of transactions and to prevent double spending attacks.

Miners are not only important for verification of transaction but also for generating new currency. They do so by putting in their computational resources to solve a cryptographic puzzle in return for newly created bitcoins. They might generate additional income known as a transaction fee if included in transactions (see Section II: Technical aspects of Bitcoin).

According to the Bitcoin protocol, one miner is rewarded approximately every ten minutes [19]. Bitcoin miners have come up with a strategy to form a mining pool to decrease the variance of their income rate [29]. These pools consist of miners who put in their computation resources to solve the cryptographic puzzle, and when they successfully find the solution, they share the reward based on the contribution of each miner.

The Bitcoin protocol is incentive compatible; the best strategy for the pool is to be honest, and if there is a pool of greedy miners, these miners cannot benefit from not following the protocol [4]. If a miner joins a large pool, he or she is supposed to earn the same reward as in the small pool because miners are rewarded based on the contribution. Therefore, there is no benefit for greedy miners to form large pools. Hence, pools consisting of honest miners pose no threat to Bitcoin protocol. Ittay Eyal et al. has found this assumption to be wrong [29].

If a minority pool follows a strategy called selfish mining [29], they can earn more than the total mining power. This strategy is about hiding information in selective ways and revealing the information if needed. It is opposite to the Bitcoin assumption where the information is broadcast to be used by others. For example, after creating the transaction, the transaction is broadcast so that the miner can record the transaction in a block. After successfully creating the block (i.e. after solving the cryptographic puzzle), the miner broadcasts it to other miners. The miners refer to this block when creating the subsequent blocks.

In selfish mining, these miners hide the block they have discovered and keep the block private by forking the *block chain* (refer to section II for more details on block chains). The honest miners continue to mine on the public block chain while the pool (consisting of greedy miners) hides the blocks they have discovered by keeping them private to themselves. When the time comes to check the length of the branch, the selfish miners reveal their discovered blocks to the other honest miners. This impacts the honest miners, for they have wasted their computational resources on discovering blocks that are not accepted by the network.

The number of computational resources wasted by honest miners is proportionally higher than that of selfish miners, and the pools consisting of selfish miners get rewards that are more than the share of mining power of the network. This lures the honest miners to join the selfish mining pool.

Ittay Eyal et al. have shown that the revenue of the selfish pool rises linearly as pool size increases [29]. It falls opposite to the Bitcoin protocol. Once the selfish mining pool reaches a certain stage, honest miners join the selfish mining pool to get more rewards, which indirectly increases their revenue. This causes the selfish mining pool to come into power (i.e. become the majority there by becoming the only source for creating the blocks). Hence, it causes the decentralized nature of Bitcoin to fall and become centralized by getting governed by this selfish mining pool.

This shows that the Bitcoin system is decentralized and safe only if all the miners are honest or if the miners do not join or form any mining pools.

To avoid this, we have come up with a protocol called SharedWealth protocol, which provides an incentive for miners to work individually. Not only is the reward shared among other miners, but the protocol also helps the miners save computational resources. For more details on SharedWealth protocol, refer to Section III. We have come up with protocol as a solution to the problem mentioned by Ittay Eyal et al. [29]

The results of our protocol show that on average, each miner earns at least 1% of the reward for every three blocks (refer to section IV for test results).

In summary, the contribution of this work is this:

1. Section I introduces the problem that we are solving and briefly overviews our protocol.
2. Section II gives detailed information about the cryptocurrencies and in particular the working of Bitcoin.
3. Section III discusses our protocol (SharedWealth Protocol) in more detail by providing examples.
4. Section IV consists of the results of the tests we conducted.
5. Section V is the conclusion and the future work.

II. Background of Bitcoin and Cryptocurrencies

This section provides a brief introduction to earlier forms of cryptocurrencies, for some of these protocols provided the base structure for Bitcoin. The first form of electronic money was introduced by Chaum and dubbed as untraceable electronic money [1] [2] to address the privacy issues of paper cash (i.e. the serial number present on paper cash makes it traceable). However, one security concern is that people can make copies of electronic money and use them at different places. To avoid this, one can have some online clearing system to keep track of the cash being generated, but this solution is expensive. Creating copies of paper money is highly infeasible. Credits cards are said to be unique, and banks can identify fraud if they are used illegally. One way of generating electronic cash is to make it difficult for people to regenerate unless they do so under the supervision of a trusted third party, such as a bank. The RSA digital signature can be used for generating electronic cash, as suggested in [1]. This money can be of the form (x, f(x) 1/3(mod n)), where factorization of n is only known to the bank and f is a cryptographic hash function. This can be summarized as follows:

1. Alice chooses a random x and r and supplies the bank with $B=r^3 f(x) \pmod n$)
2. The bank returns the third root of B modulo n: $r * f(x)^{1/3} \pmod n$ and withdraws one dollar from Alice's account.
3. Alice extracts $c=f(x)^{1/3} \bmod n$ from B.
4. If Alice wants to send Bob one dollar, Alice gives him $(x, f(x)^{1/3} \pmod n)$.
5. Bob can verify with the bank to check whether the electronic coin has been used.

The benefit of this approach is that anyone can verify whether the electronic cash has the right structure and was signed by the bank. Electronic cash provides anonymity, which means that the bank cannot tell the electronic cash is connected to Alice.

Throughout the 1990s, various versions and advancements of this protocol were proposed. One such contribution suggested removing the bank at purchase time [2]. This allowed the division of coins into smaller units [3]. This system allowed the user to divide the amount C into any number of values. The total value of the entire divided amount equals C. The security of the system depends upon the difficulty of factoring [3]. Various startup companies have sprouted based on this idea, such as DigiCash [4] and Peppercoin [5]. With Peppercoin, the micropayments received were identified using cryptographic selection, which was then used for upgrading these micropayments to macro-payments. The idea was to allow merchants to collect micropayments and create smaller collections of macro-payments. The value of these macro-payments should equal the sum of the total value of the micropayments. The benefit is that the cost of processing the macro-payments is less than that of the micropayments. Though the method was supposed to be secure, it did not gain much popularity. Both DigiCash and Peppercoin failed later.

Bitcoin uses *proof of work* as its core for validating transactions and generating new bitcoins. The proof-of-work system was introduced in the 1990s as a way to prevent email spam [5], although it was never used for that purpose [6]. In 1997, another protocol called Hashcash [11]

that was based on [5] was used to prevent email spam. Hashcash is based on CPU cycles. Generating hashes is expensive because it requires spending CPU cycles, but verifying is cheap and can be done instantly. One use of Hashcash is to detect email spam. Email spam refers to users abusing the email system by sending millions of emails. Mostly, the users belong to commercial industry and want to market products to people. From their perspective, sending email requires zero effort, and the success rate of making a profit is more than 0%. But on the other side, people who receive this email are annoyed due to advertisements that do not match their interest; moreover, their CPU resources are wasted in processing the emails. This is a major issue for Internet service providers because handling bulk emails can result in wasting time and money. Hence, Hashcash is used to prevent this type of attack. The following figure illustrates the workings of Hashcash.

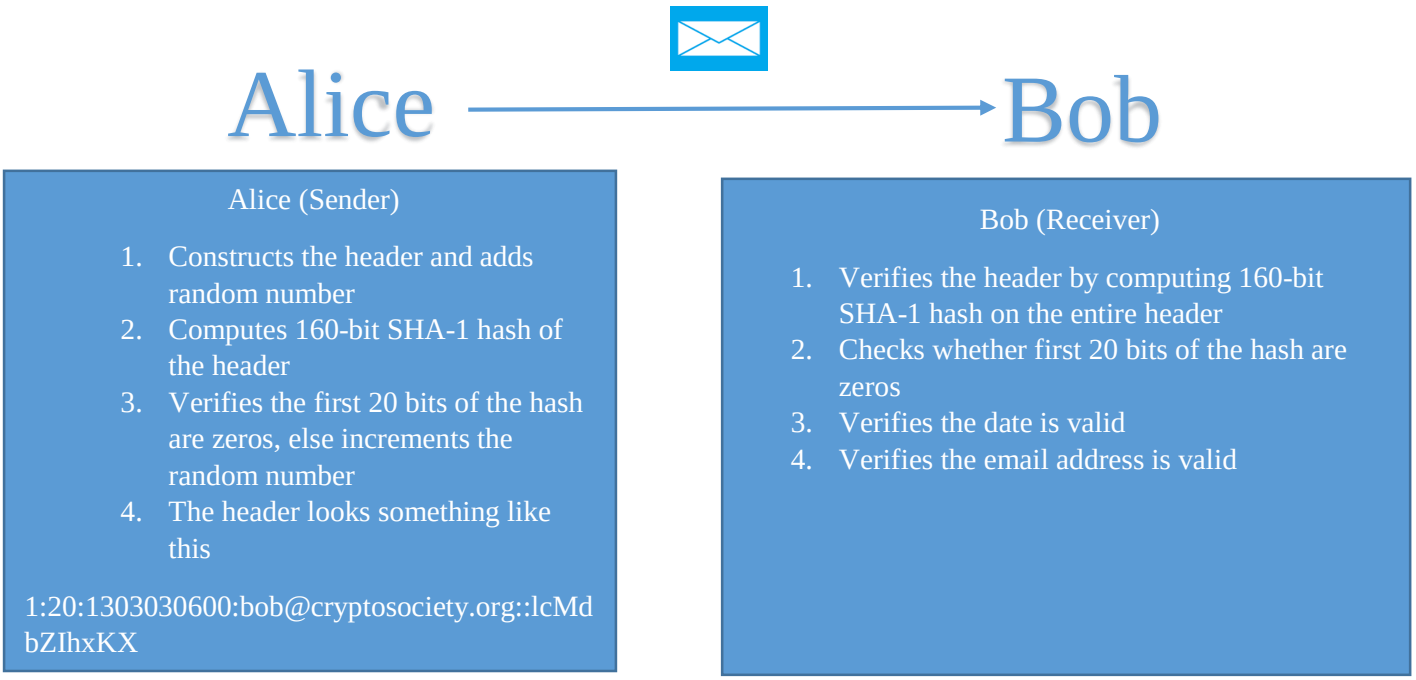


Fig. 1: Process of Hashcash

Header contains:

- Version: Hashcash format version, 1.
- Bits: Number of zero bits in the hashed code.
- Date: The time that the message was sent, in the format YYMMDD [hhmm [ss]].
- Resource: IP address or email address.
- Rand: String of random characters, encoded in base-64 format.
- Counter: Binary counter, encoded in base-64 format.

One important part of Bitcoin is the public ledger, which provides a way for detecting *double-spending*. Double-spending is a strategic way of spending money more than once. The attacker tries to misuse the trust of a merchant by convincing him that the transaction has happened and somehow cheats the network by making them accept some different transaction. This way, the merchant loses the product and gains no profit whereas the attacker gets the product and saves his or her money.

In the late 1990s, auditable ecash [12] [13] was proposed. One drawback of this approach is that the bank can detect unreported valid money [12]. Also, if the bank's secret key is stolen, then the whole system gets compromised. Auditable ecash provides an anonymous, auditable system that is signature free [13]. It means that the bank does not have any secrets, and also the transactions are untraceable.

B-money [14], proposed in 1998, was the first design that suggested transactions should be publicly available. It proposed two protocols:

- The first protocol suggested every individual should have a database that consists of information on the total money each user has. It suggested that anyone can create money by publishing the solution of the unsolved computational problem. The condition is that it should be easy enough to determine the computational effort spent on the problem. The units created will be equal to the cost of the effort spent. W.Dai. [14] provided an example to illustrate: if someone spends 100 hours to solve the problem and it takes three units to purchase 100 hours of computing time, then the user who spends 100 hours shall be credited with three units. The transfer of money in this model is such that if a user A broadcasts a message saying that he or she wants to transfer X amount to user B and that it is signed by user A, then everyone debits X amount from user A and credits X amount to user B, unless the balance of user B becomes negative, in which case this transaction is ignored.

- The second protocol suggested having a subset of participants or servers keep track of the money instead of every individual. The servers are connected using USNET-style broadcast channels [14]. The format of the transaction remains the same as the first protocol, but the transaction is verified by a subset of participants selected randomly. The question that arises is how we can ensure that the servers remain honest. The solution is that the server has to deposit some amount of money in certain accounts to be used as fines or rewards in case of proof of misconduct [14]. The server also regularly publishes and commits to a database consisting of money creation and ownership. The users have to check whether the account balance exceeds the total amount of money created. Hence, the protocols mentioned in [14] allow anonymity and some form of contracts.

The next protocol is smart contracts [15]. A contract is nothing but a set of promises that are agreed upon. It gives a structure to a business relationship. Contracts are not only written, but oral agreements can also be considered contracts [15]. They are the essential elements of market economy. Computer scientists and cryptographers came up with a way of combining messages and algorithms to create new protocols. These protocols are deployed on public networks like the

Internet, which bring new facets similar to contract law. Smart contracts decrease computational transaction costs [15]. The building blocks of smart contracts are protocols. Protocol in computer science can be defined as a sequence of messages exchanged between at least two computers [15]. It is nothing but a set of algorithms communicating via messages.

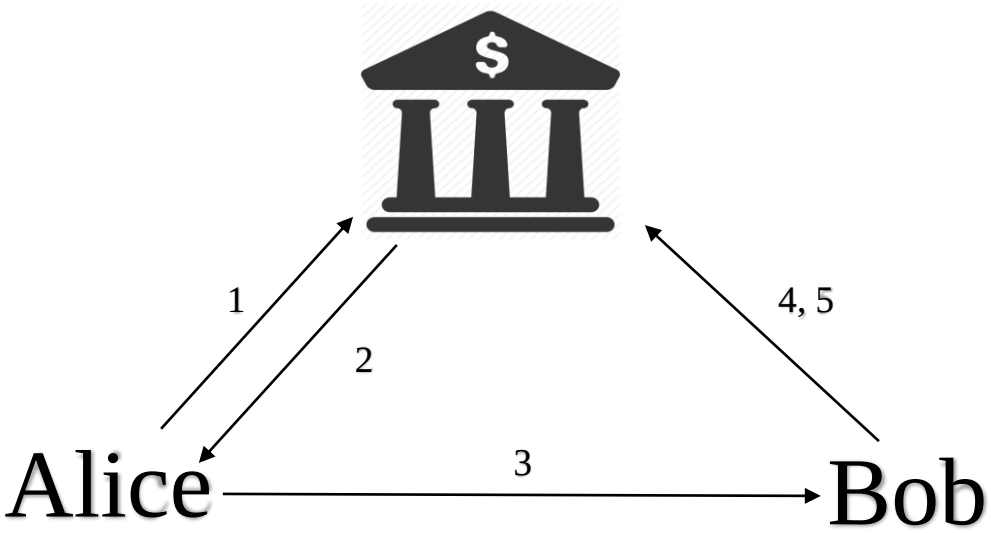
Protocols must be unique and complete in contrast to real-world contracts. Protocols for smart contracts come in three types: self-enforcing, where one user deals with another user; mediated, where a third party acts as intermediator between the two users; and adjudicated, which is based on evidence [15]. Cryptographic protocols are based on obscurity. The obscurity, or randomness, makes the system simple and public. Randomness is always associated with a key in cryptographic protocols, and guessing it correctly should be highly infeasible. Smart contracts are based on cryptographic protocols. Protocols of smart contracts should have structure to provide robustness, which protects from naive vandalism and sophisticated attacks. In conclusion, smart contracts consist of protocols, user interfaces, and promises that are shared through an interface to establish a relationship over a public network [15].

Digital cash [16] [17] brought a revolution in how we trade. It mimicked paper cash by ensuring anonymity and transferability of payment. It eased the way transactions happened by providing ways like sending a payment over email, USB, Bluetooth, etc. Digital cash represents values, and the architecture of digital cash consists of a trusted third party, such as a government or bank. One thing to note is that electronic cash is not digital cash because it does not provide properties such as anonymity and offline transferability.

Digital cash works as follows

1. Alice asks for digital cash from her bank.
2. The bank deducts the corresponding amount from her account and gives her the coins.
3. Alice uses digital cash to buy coffee from Bob's Coffee Shop, which accepts digital cash.
4. Bob verifies whether the cash has been authorized by the bank.
5. Bob then requests the bank to pay the cash equivalent of the digital cash.

Fig 2: Example of Digital Cash Transaction



The desired properties of digital cash follow: it is secured, so Alice or Bob cannot reproduce digital cash; it is anonymous, so neither Alice nor Bob have to reveal their identities to each other, nor can the banks find out who Alice paid or who Bob was paid by; it is portable, so it can be stored on a disk; it is two way (i.e. peer-to-peer payments should be possible) and offline capable, so Alice and Bob can have a transaction at any time without a third party authentication; it has wide acceptability, so it is accepted and well known in the market; it is user friendly, so it simplifies its complexity, and Alice and Bob can use it without knowing what is going on under the hood.

Two major concerns with digital cash are anonymity and double spending. Banks can deduce user identity by tracing the cash that they gave to Alice to match it with the cash that came from Bob. From tracing, they can potentially identify Alice gave the cash to Bob. Electronic money prevents this by having the bank acknowledge the payment for every transaction.

Bitcoin was proposed by Satoshi Nakamoto in 2008 [1]. The first Bitcoin block was mined on January 3, 2009 [18]. This block is known as the genesis block. The very first purchase done using Bitcoin was in May 2010[18], where a user bought a pizza for 10,000 bitcoins. Since then, the number of transactions using bitcoins has increased tremendously; bitcoin value has been increasing, and at one point in late 2013 it was $1200 per bitcoin [18].

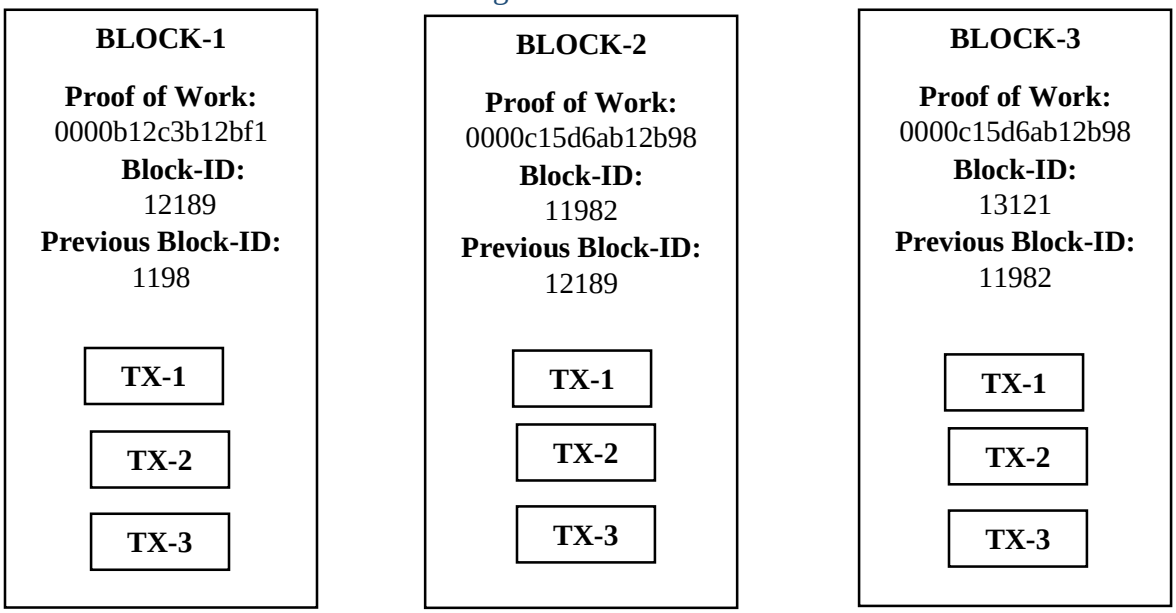
The next section provides a high-level working of Bitcoin, and later sections provide the in-depth technical details of Bitcoin.

Overview of Bitcoin

Bitcoin is a distributed decentralized cryptocurrency [19]. The user of Bitcoin gets an address, which can be shared with other users to be used for forming a transaction. These transactions are added to a public ledger called a *block chain*. A group of miners ensures that the block chain is unsullied in return for bitcoins as a reward. All the transactions are recorded in *blocks*.

These blocks form a block chain. Each block includes a unique ID and the ID of the previous block. A valid block consists of a proof of work (i.e. a solution to the cryptographic puzzle) and the address of the miner. This process is called Bitcoin mining. Upon successfully creating the block, the miner publishes it to the Bitcoin network consisting of all the other miners. According to Bitcoin protocol, the miners add blocks to the longest chain. The average mining time interval is 10 minutes.

Fig. 3: Block Structure



A Bitcoin transaction can be thought of as an exchange of electronic coins. A user transfers these coins to the next user by digitally signing the hash of the previous transaction along with the public key of the next user [19]. A payee can verify the signatures to validate the ownership, and the transactions are publicly announced. The transaction structure is illustrated in Fig. 4. [19]

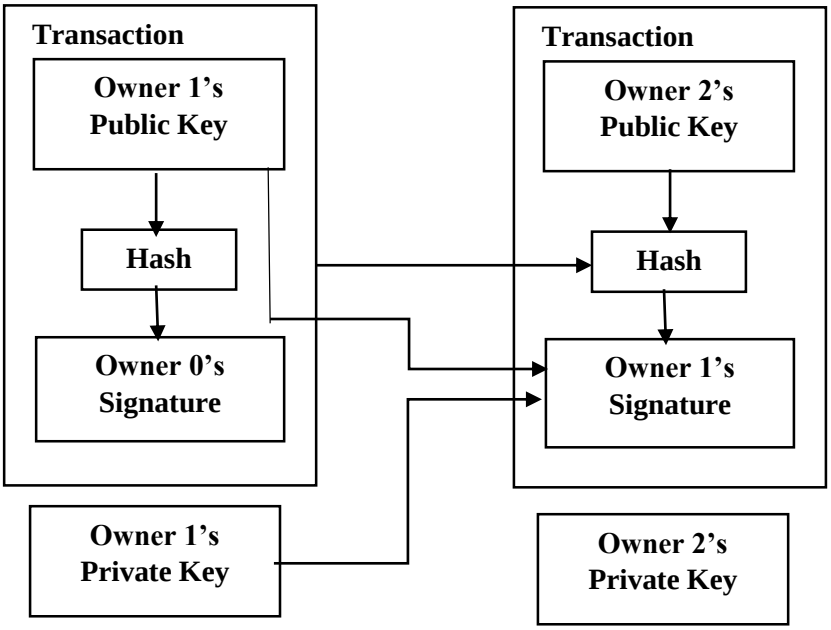


Fig. 4: Transaction Structure

The solution to cryptographic puzzles is similar to a timestamp server, which calculates the hash of a block of items and publishes it in a newspaper or Usenet post [20-23]. The timestamp validates the correctness of information for that particular date as a timestamp is used to generate the hash. Bitcoin uses a proof-of-work system similar to Adam Back's Hashcash [11]. It involves finding a value so that when hashed with SHA-256, the result begins with a certain number of zero bits. Hence, the difficulty of proof of work depends upon the zero bits required, but the solution can be verified with a single hash. This ensures that the block that is generated using the proof-of-work system cannot be modified without spending computation resources. Anyone can tell whether the block is modified by verifying the proof of work. The blocks are chained together to form the block chain. To modify one block, one has to recompute the proof of work of all the previous blocks, which is infeasible. According to the protocol, the miners work on the longest chain available, for it signifies the greatest proof of work invested [19]. The proof of work difficulty changes based on the number of blocks generated per hour. The difficulty is increased if they are generated too quickly [19].

The Bitcoin network is a peer-to-peer network [19]. The transactions are broadcast to all the nodes. It is not necessary for the transaction to reach all nodes, for as long as they reach many nodes, they are included in the block. These nodes collect the transactions into a block, and after collecting into the block, they work on solving the cryptographic puzzle (i.e. proof of work) of that block. Upon successfully computing the proof of work, they broadcast it to all the other nodes. The criteria of acceptance of the block depend upon the validity of the transactions in the block. Once the nodes accept the block, they use the accepted block for creating the next block in the chain, using the previous block's hash.

Technical Aspects of Bitcoin

There is no authoritative formal specification apart from the original Bitcoin white paper [19]. The three main components of Bitcoin are transactions, the consensus protocol, and the communication network [18] [19]. The implementation of bitcoind is considered to be the de facto specification; other details are included in "Bitcoin Improvement Proposals" (BIPs) [19].

Transactions with Bitcoin can be thought of as series of messages exchanged over the network. A transaction consists of inputs and an array of outputs. The entire transaction is hashed using SHA-256, and it is used as the transactions unique ID. The output, consisting of an integer value, represents the units of Bitcoin currency. The smallest unit of currency in Bitcoin is a satoshi, where 10^8 satoshis are worth one bitcoin (represented as BTC or XBT). The output of the transaction is the script called scriptPubKey, which provides the conditions for redeeming the transactions. The output of a transaction is included as an input of another transaction.

The scriptPubKey is also known as "pay-to-pub-key-hash." The transaction has to be signed with the key along with the hash. Most of the transactions in Bitcoin are pay-to-pub-key-hash. The scripting language use is an adhoc, non-Turing complete stack language with fewer than 200 commands called opcodes [18]. The opcodes include cryptographic operations like hashing and verification of signatures. The transaction format and the scripting languages are specified in bitcoind [18]

To redeem previous transactions, the scriptSig script and scriptPubKey script should execute successfully. According to the Bitcoin protocol, each transaction input should match with the previous transaction output. This means that the sum of all values of the transaction output should be less than or equal to the sum of all inputs. Users of Bitcoin own a private key that they can use to redeem certain outputs. Therefore, a user can have as many bitcoins as he or she can redeem. The address of the user is the public key hash, and the address does not contain any real-world name or identity [18].

It is very difficult to secure currency that is only based on transactions. Though signatures provide a certain form of authority for the user over previous transactions, they do not provide a solution to the double spending problem. The double spending problem in Bitcoin can be illustrated by Alice trying to redeem transaction inputs twice in separate transactions sent to Dave and to Charlie. Bitcoin provides an elegant solution to the double spending attack. The protocol specifies that all the transactions must be published publicly in a global ledger.

Therefore, anyone can verify a transaction by checking the public ledger. This public ledger is a data structure that consists of a series of blocks of transactions. Each block contains the hash of the previous block, and hence all the blocks are chained together. This chain is known as the block chain.

If Charlie and Dave find two different block chains, then they are susceptible to a double spending attack. One solution is to have a central authority that controls over the block chain, but this is against decentralized property of Bitcoin, and moreover the central authority might exert control on what transactions the block should have. Hence, having central authority for consensus is not feasible. Bitcoin uses Nakamoto consensus [18][19]. This protocol is one of groundbreaking innovations and one of the important aspects of Bitcoin's success. Anyone can collect the transactions into the block, but the challenge is to solve the computational puzzle known as the proof of work.

The steps of selecting the new block are as follows:

1. The miner who finds the solution to the computational puzzle will announce it publicly along with the solution to the puzzle.
2. All the other miners should validate the work of the miner.
3. If the work is invalid, the miners will reject the block and continue with their work.
4. If the work is valid, and the solution of the cryptographic puzzle is also correct, the miners accept the block.

The longest chain is said to have the highest mining power because it consists of more blocks and is the most difficult to produce. In some cases, two valid solutions can be found at the same time, which results in forking of the block chain. Miners can choose any one of the chains to work on, but later they have to adopt the one that has surpassed the other chain in length. The user must wait for blocks to be identified, which will provide confidence that transactions are included in the block chain [18]. Usually, if there are two branches, it is most likely that the branches contain the same transactions. Usually, Bitcoin clients require six confirmation blocks to accept a transaction.

The only way the currency is generated is by mining. All new currency is received by the miners and gets circulated over the network in the form of transactions. The computational puzzle (or cryptographic puzzle) is to find a block containing n number of transactions, the hash of the previous block, the timestamp, the version number, and a random nonce value that when supplied to a one-way hash function results in a value, which is less than the target value. In other words, solving the puzzle is finding the hash of the block whose result starts with enough zero bits. The general technique used is to add a random nonce to the hash function until finding the desired solution. Due to the randomized nature of the puzzle, each miner finds a block based on the computational resources he puts in. The difficulty of puzzle is adjusted in such a way that on average new blocks are introduced every 10 minutes. The adjustment is done after every 2016 blocks, which is approximately two weeks.

The mining reward is structured. It halves every four years, and it is said by the year 2140, no new bitcoins will be created [18]. Initially, the mining reward was 50 BTC, and it was reduced to

25 BTC. The miners earn the block reward, but they also earn the difference in value between the input and output of transactions in the block. This is considered their transaction fee. Users can include the transaction fee in their transaction for their transaction to be included in the block more quickly.

Mining depends on the difficulty of the cryptographic puzzle. The difficulty (D) is adjusted by network. The probability of finding the valid block is $1/ (2^{32} D)$. The average number of blocks found by the miner is $ht / (2^{32} D)$ where *h* is the hash rate and *t* is time [27].

For example, if Dave has an ASIC that can perform billion hashes per seconds, $h=10^9$ hash/seconds. If he mines for a day based with difficulty D = 1990906 and the block reward is 25 BTC, he will find an average of 0.010 block/day, so he will make 0.2525 BTC per day.

Usually miners form a pool to solve the computation puzzle and to lower the variance of their revenue by dividing the reward with the others members of the pool [18]. The mining pools are usually governed by the manager, who distributes the rewards to the members of the pool based on the amount of work they have done for the pool. Mining pools were not part of original protocol, but since 2013 mining is done mostly in pools [18]. The pools are structured in a way that the revenue is divided among the members with stress on loyalty and discouraging pool hopping [25]

If the hash rate of the mining pool is H, then the pool will find an average of $Ht / (2^{32} D)$ blocks [27]. The average reward earned by the pool is $HtB / (2^{32} D)$ where B=bitcoins. If a miner with hash rate h=pH (where p is the fraction of pool's total power contributed by the miner), he will receive a $qHtB / (2^{32}D)$ reward.

The pool is maintained by an operator who charges f percentage of the block reward [27]. Therefore, the operator for each block received fR where R is the reward and the rest (1-f) R is distributed between the pool members.

One of the core components of Bitcoin is the communication network. It is a decentralized ad hoc peer-to-peer network. The consensus protocol depends on the performance and stability of the network because latency in finding a block might increase the possibility of a temporary fork, and an attacker will benefit from this fact and might be able to control a substantial portion of the network by winning the fork and there by getting more reward. Thus Bitcoin has to have a decentralized network with low latency.

The network topology of Bitcoin is such that any node can join the network, and each node has eight outgoing connections and can handle up to 125 incoming connections [18]. Note that mobile clients do not handle incoming connections. Peers who join the network initially have to introduce themselves to other peers. This is similar to other peer-to-peer networks. Bitcoin establishes this using dedicated directory servers, which consist of lists of peer addresses. Peers send information about one another using two ways. First, when the peer joins the network, it broadcasts messages containing its information. Second, when it receives incoming connections, the node asks for a sample from the list of addresses [18] [26]. The transactions and blocks are

broadcast to the entire network by flooding. Node forwards new data only once to prevent infinite propagation [18].

III. Shared Wealth Cryptocurrency

The security of Bitcoin lies in its decentralized nature and its being public since all transactions are open and stored in series of blocks (refers to Technical Aspects of Bitcoin, Section II). Miners play a crucial role in ensuring that the Bitcoin works as explained in [19], and they are also responsible for preventing double-spending attacks. Miners are compensated with bitcoins for their effort, which includes creating blocks and solving cryptographic puzzles.

As the level of difficulty is adjusted based on the total mining power of the network, an individual miner has to mine a block for years if he or she is using an ASIC [31].

To illustrate individual mining, consider four miners: Ted, Charlie, Ross, and Keith. They record *N* transactions in blocks and toil to find the proof of work by solving the cryptographic puzzle. After working on an individual block for *T* amount of time, Ted finds the proof of work. For ease of understanding, we assume that the solution of proof of work should initially have four bytes as zeroes (as shown in Fig. 5), and as an outcome Ted earns the reward. As portrayed in the diagram, Keith and Ross nearly got the solution, but due to the Bitcoin reward structure, their work is wasted (futile).

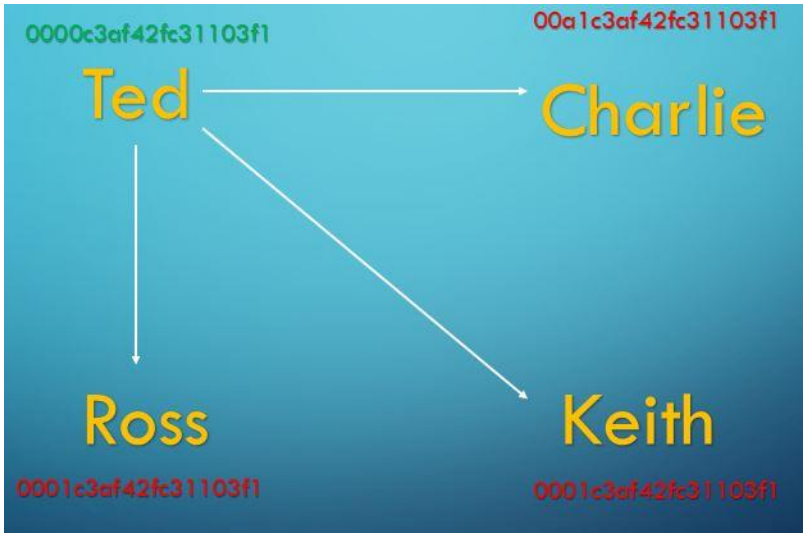


Fig. 5: Individual mining

People join mining pools to earn bitcoins on regular intervals instead of waiting for a year and risking the possibility of their work going to waste [31]. The mining pool wait time is generally predictable, and the income they earn is steady; in contrast, if they mine alone, there are higher chances of their work ending up wasted. Hence, the miners decide to form their own mining pools or collaborate with others [18]. The members of the mining pool work together in solving the cryptographic puzzle. Upon completion, the reward is distributed based on the amount of mining power (computational resources) each individual exerted.

SharedWealth Protocol

We have come up with a protocol that provides an incentive for miners to share their partially solved cryptographic puzzle (we dubbed it a near-miss) with other miners. The outcome reward earned is dispersed among the miners based on their effort. Here is how our protocol would operate:

1. Instead of giving the reward to one individual miner, the reward is divided into 80% and 20%.
2. 80% of the reward will go to the miner who finds the actual proof of work (solution to the cryptographic puzzle) as well as an additional coin for including the near-miss solutions of the other miners.
3. The remaining 20% of the reward is dispersed among the miners who shared the "near-miss."
    a. The near-miss is the partial solution found by the miners. If the actual solution starts with n bits of zeroes, solution starting with 1, 2 or 3 … n-1 bits of zeroes is considered as near-miss. We did the tests for near-miss = 1,2,3 and 4 (refer section IV for more details)
    b. The wining miner cannot include his near-miss. If he includes his near-miss, it will be discarded and he will be only paid for the actual solution.
    c. Miners cannot send multiple near-misses.
    d. If they are multiple winners of near-miss, the 20 % reward is divided among the winners based on their near-miss. For example, if they are three near-miss winners who send near-miss of 2, 2 and 3 then 10% of reward is given to the one who have send the near-miss 3 and the rest 10% is divided among the two miners who send the near-misses of 2.
4. If the winning miner does not include the near-miss of other miners. He only gets the 80% reward and the 20% reward is discarded.

To illustrate, let us again consider four miners: Ted, Ross, Charlie, and Keith. They are working by spending their individual resources to find the solution to the cryptographic puzzle. As Fig. 6 shows, only Ted finds the proof of work which starts with four bits of zeroes, but Keith and Charlie send their near-miss (partial solution) to him. Therefore, per our protocol design, the reward is dispersed as follows:

Reward structure:

Ted = 80% of Reward

Keith = (20/2) % of Reward

Ross = (20/2) % of Reward

Ted + = 2

Keith and Ross also receive some of the reward, and Ted retains 80% of the reward and an additional two coins for including Keith and Ross's partial solution.

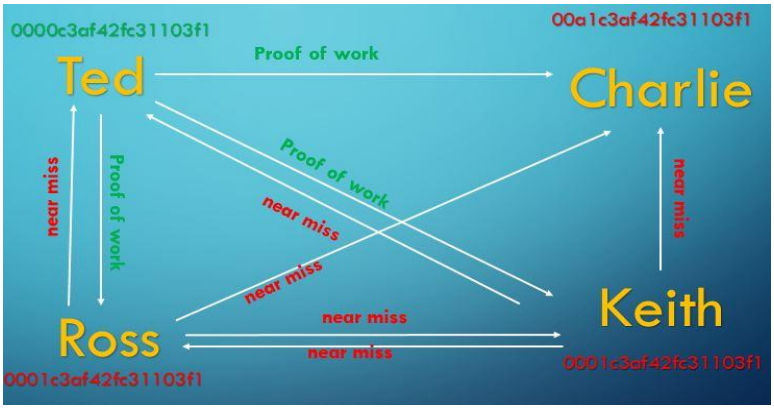


Fig. 6: Illustration of SharedWealth Protocol

Let's Examine Possible Attacks:

1. Trudy ignores the near-miss and does not include it in her solution.

Let us propose that Trudy is working alone, and due to her selfish nature, she wishes not to share the reward. She will not include the near-miss in her solution. The solution for this problem can be this:

- Provide an incentive for Trudy to include the near-miss of other miners. As you can see from our reward structure, Trudy will receive extra coins if she includes the near-miss of other miners.
- If Trudy still wishes not to include the near-miss, then 20% of the reward is discarded. Consequently, Trudy receives only 80% of the reward. As a result, Trudy is forced to include the near-miss of other miners because it will provide her additional income on top of the regular 80%.

2. Trudy finds the near-miss on her own, including it in her solution but discarding the near-miss of others.

Suppose Trudy comes up with a plan to find the near-miss on her own, and she includes it in her solution along with her proof of work to obtain the full reward. This is the proposed solution:

- Assign a unique ID to each miner. Each time the solution is broadcast, verify to see whether the ID of the miner who sent the near-miss matches the ID of the miner who found the actual proof of work. If it matches, then discard the 20% reward and give only 80% of the reward to the miner.

3. Trudy only includes the near-miss of her friends while discarding the near-miss of other miners.

Suppose Trudy has or is forced to have friends. Due to our above design, she wishes only to include the near-miss of her friends and discard the near-miss of others. This is the proposed solution:

- As the solution is broadcast over the network, the miners who find the near-miss will check the solution and notice if their near-miss is not included. In return, those miners who were not compensated because of Trudy's selfish act will not include her near-miss in the future. This is the conventional tit-for-tat approach [32].

Advantages of Our Proposed Solutions:

1. Selfish mining [29] and other attacks [31] are prevented when miners do not form mining pools.
2. Overall mining cost is reduced when miners share their work.
3. New currency generated is distributed among miners depending on the amount of mining power.
4. An individual miner does not have to wait for a year to earn a reward.

IV. Testing

For testing our protocol, we have created two models as follows:

Winner-Take-All

This model is based on Bitcoin protocol. It follows the reward strucure as described in Section II and Section III.

SharedWealth

This model is based on SharedWealth protocol. It follows the reward structure as described in Section III.

Setup

The tests were were conducted on a Dell Laptop with the following configurations:

Processor: Intel(R) Core(TM) i7-4500U CPU @ 1.80GHz 2.40Hz

RAM: 8.00 GB

Operating System: Windows 10 (64-bit)

Java version: 1.7.0_85

Java HotSpot(TM) 64-Bit Server VM (build 25.65-b01, mixed mode)

Simutation

The simulation was done using Sockets and Multi-Threading. Each of the thread represents the miner and transaction. All the transaction details are send to the miner Threads. These threads are chosen randomly to illustrate peer-to-peer connectivity. All the threads shares the same computational resources.

Demonstration of Reward Distribution in Winner-Take-All

We simulated Bitcoin Network with six miners (Alice, Bob, Trudy, Charlie, Dave, and Nate). They compete against each other to create blocks and solve the proof of work. The difficulty is increased after every six blocks.

The following tests were conducted with difficulty levels starting from 2 to 5. The idea behind this test is to show how the wealth is unevenly distrubuted and also how the work of other miners gets wasted. For this demonstration, we are rewarding the winner with 20 coins.

The bars refers to the total reward earned after six blocks.

Demonstration of reward distribuition in SharedWealth

We simulated the SharedWealth Network with six miners (Alice, Bob, Trudy, Charlie, Dave, and Nate). They share their near-miss with each other and get rewarded for sharing the near-miss.

We did a test with difficulty levels starting from 2 to 5. The idea behind this test is to show how the wealth is evenly distrubuted and how the work of other miners does not get wasted. For this demonstration, the reward structure is based on SharedWealth protocol mentioned in Section III.

In these tests, when the difficulty is *n*, some of the miners find proof of work for *n-1*, which is the near-miss, and share it with other miners.

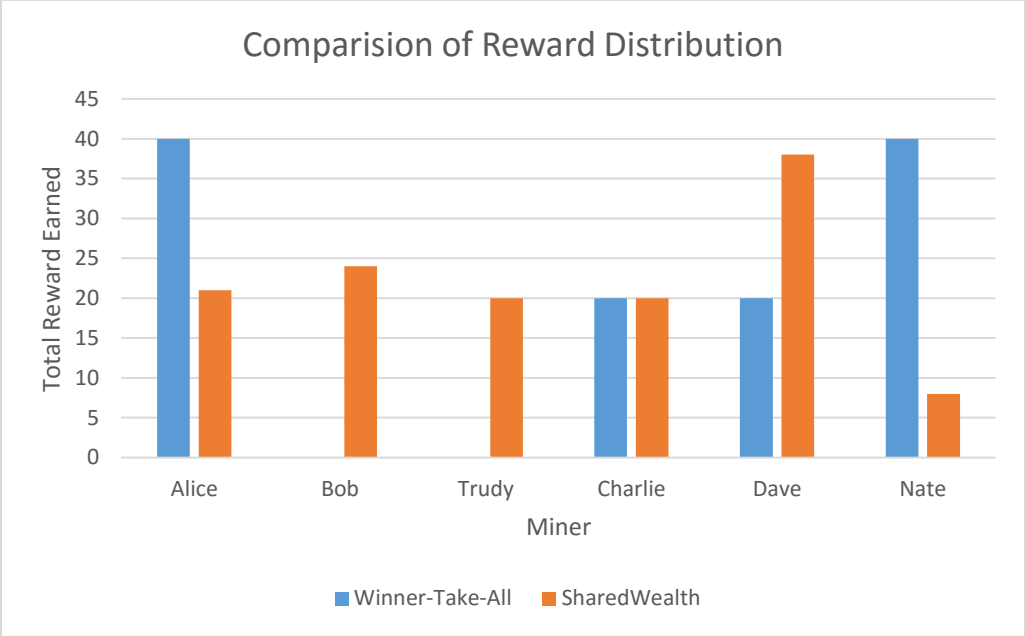


Figure 7: Reward distribution when difficulty is 2

Miner	Rewards Earned in Winner-Take-All
Alice	40 (Block 1 and Block 2 winner)
Bob	0
Trudy	0
Charlie	20
Dave	40 (Block 3 and Block 5 winner)
Nate	20

Miner	Rewards Earned in SharedWealth	Total
Alice	2 + 2 + 16 + 1 (Block 1, Block 3 near-miss winner, and Block 5 winner)	21
Bob	2 + 2 + 2 + 16 + 2 (Block 1, Block 2, Block 4 near-miss winner, and Block 6 winner)	24
Trudy	2 + 16 + 2 (Block 1 near-miss winner and Block 4 winner)	20
Charlie	16 + 2 + 2 (Block 3 winner and Block 4 near-miss winner)	20
Dave	16 + 2 +16 + 2 + 2 (Block 1, Block 2 winner, and Block 6 near-miss winner)	38
Nate	2 + 4 + 2 (Block 1, Block 5, and Block 6 near-miss winner)	8

From the above graph/data, we can see that in Winner-Take-All, though other miners put their resources towards solving the cryptographic puzzle, their work was futile because they did not get any reward.

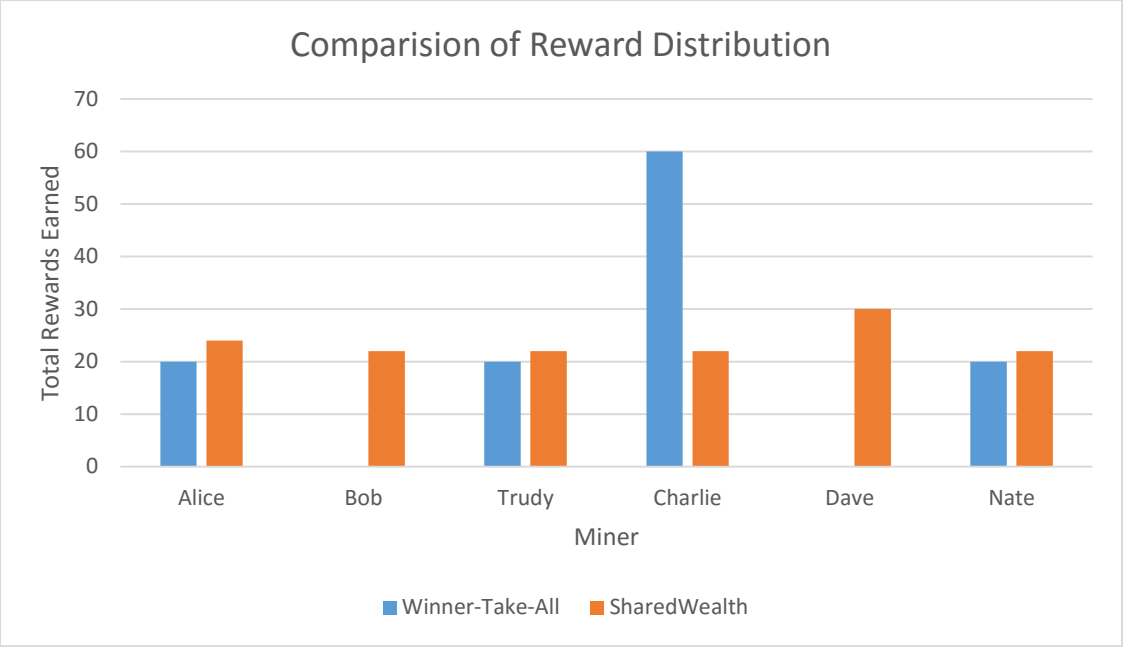


Figure 8: Reward distribution when difficulty is 3

Miner	Rewards Earned in Winner-Take-All
Alice	20
Bob	0
Trudy	20
Charlie	60 (Block 1, Block 3, and Block 6 winner)
Dave	0
Nate	20

Miner	Rewards Earned in SharedWealth	Total
Alice	2 + 2 + 16 + 2 + 2(Block 1, Block 2, Block 5 near-miss winner, and Block 4 winner)	24
Bob	16 + 2 + 2 + 2 (Block 3 winner, Block 4, and Block 6 near-miss winner)	22
Trudy	2 + 16 + 2 +2 (Block 1, Block 6 near-miss winner, and Block 2 winner)	22
Charlie	2 + 2 +16 + 2 (Block 2, Block 4 near-miss winner, and Block 6 winner)	22
Dave	2 + 16 + 2 (Block 3 near-miss winner and Block 5 winner)	30
Nate	16 + 2 +2 +2 (Block 1 winner, Block 3, and Block 5 near-miss winner)	22

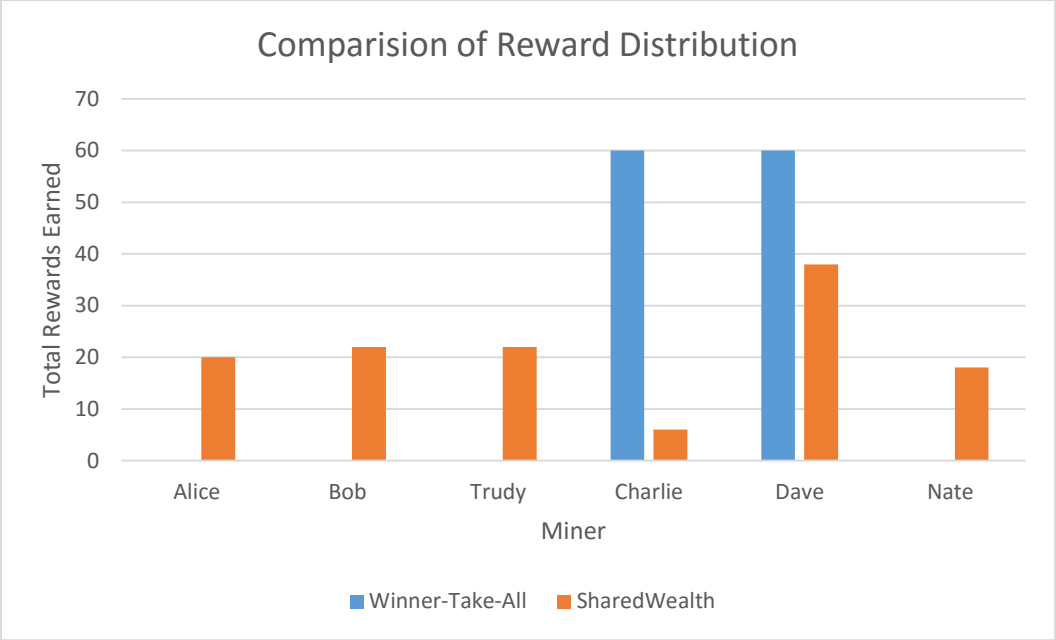


Figure 9: Reward distribution when difficulty is 4

Miner	Rewards Earned in Winner-Take-All
Alice	0
Bob	0
Trudy	0
Charlie	60 (Block 2, Block 4 and Block 5 winner)
Dave	60 (Block 1,Block 3 and Block 6 winner)
Nate	0

Miner	Rewards Earned in SharedWealth	Total
Alice	16 +2 + 2 (Block 2 winner and Block 3 near-miss winner)	20
Bob	2 + 16 + 2 + 2 (Block 3 winner, Block 1, and Block 5 near-miss winner)	22
Trudy	2 + 2 + 16 + 2 (Block 2, Block 3, Block 6 near-miss winner, and Block 4 winner)	22
Charlie	2 + 2 + 2 (Block 2, Block 5, and Block 6 near-miss winner)	6
Dave	2 + 16 + 2 +16 + 2 (Block 1 near-miss winner, Block 5, and Block 6 winner)	38
Nate	16 + 2 (Block 1 winner)	18

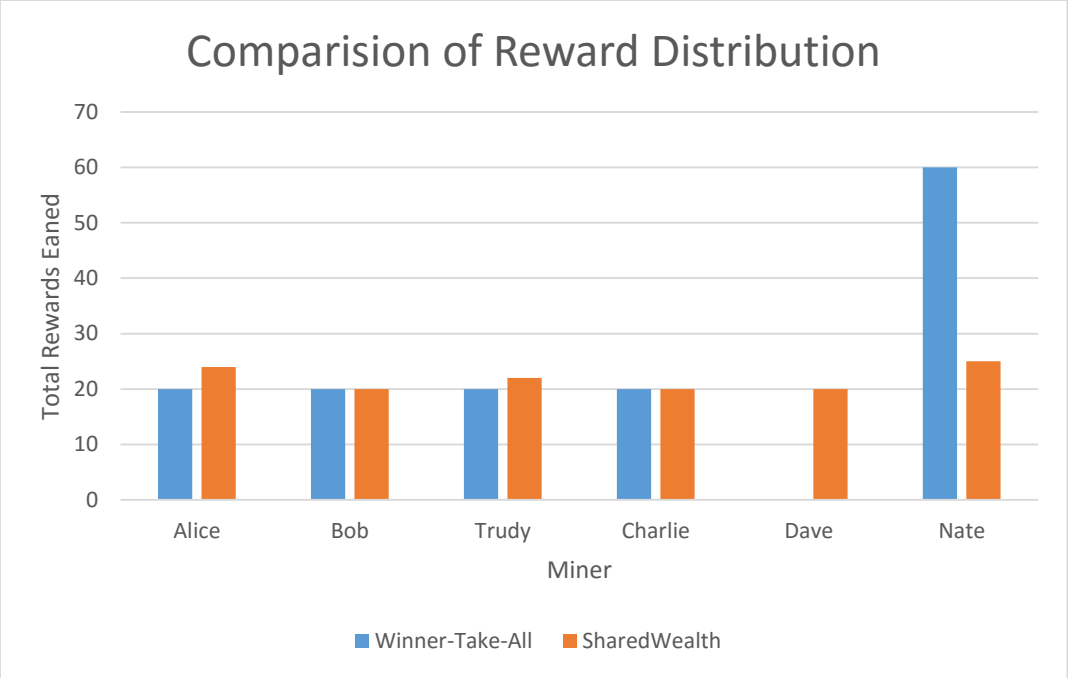


Figure 10: Reward distribution when difficulty is 5

Miner	Rewards Earned
Alice	20
Bob	20
Trudy	20
Charlie	20
Dave	0
Nate	40 (Block 3 and Block 6 Winner)

Miner	Rewards Earned	Total
Alice	2 + 16 + 2 + 4 (Block 1, Block 5 near-miss winner, and Block 4 winner)	24
Bob	16 + 2 +2 (Block 1 winner and Block 4 near-miss winner)	20
Trudy	2 + 2 + 16 + 2 (Block 1, Block 2 near-miss winner, and Block 6 near-miss winner)	22
Charlie	16 + 2 + 2 (Block 3 winner and Block 6 near-miss winner)	20
Dave	16 + 2 +2 (Block 2 winner and Block 3 near-miss winner)	20
Nate	2 + 2 + 2 + 16 + 1 + 2 (Block 2, Block 3, Block 4, Block 6 near-miss winner, and Block 5 winner)	25

Demonstration of Reward Distribution when number of miners are 500 and 1000

We conducted another test to show reward distribution in winner-take-all and SharedWealth when blocks discovered are 40, 50 and 100. The number of miners are 500 and 1000. The following are the results

Number of Miners	Blocks Discovered	Winner-Take-All*	SharedWealth* (near-miss:2)
500	40	36 (7.19%)	81 (16.2%)
500	50	44 (8.79%)	147 (29.4%)
1000	100	59 (11.79%)	237 (47.4%)

*** Count of Miners who receive some reward**

From the above results we see that the number of miners who receive some reward in SharedWealth model is more than the Winner-Take-All model.

V. Conclusion

We studied the Bitcoin protocol and how miners join or form mining pools to earn steady rewards. We explored the possible attacks associated with mining pools [29].

To prevent such attacks, we created a distinct peer-to-peer protocol called the SharedWealth protocol, which ensures that each reward is evenly distributed among miners. With our demonstration, we showed how the reward is evenly distributed.

In the future, we want to add some restriction on miners so that they cannot earn more coins than the total reward. For example, right now we are giving miners an additional coin for each near-miss they include in their solutions. The possible vulnerability here is that miners can add 100 near-misses, and this will result in them earning 100 coins, which is more than the total reward. We also want the difficulty level adjusted based on the average number of near-misses the miners find for a given time *t*.

References:

[1] Chaum, D., Security without identification: Transaction systems to make big brother obsolete, Comm. ACM 28, 10 (October 1985).

[2] Chaum, David. "Demonstrating that a public predicate can be satisfied without revealing any information about how." *Advances in Cryptology—CRYPTO'86*. Springer Berlin Heidelberg, 1987.

[3] T. Okamoto and K. Ohta. Universal electronic cash. In CRYPTO, 1992.

[4] B. Schoenmakers. Security aspects of the EcashTM payment system. State of the Art in Applied Cryptography, 1998

[5] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In CRYPTO, 1992

[6] B. Laurie and R. Clayton. Proof-of-work proves not to work. In WEIS, 2004.

[7] D. M. Goldschlag and S. G. Stubblebine. Publicly Verifiable Lotteries: Applications of Delaying Functions. In Financial Cryptography, 1998

[8] R. L. Rivest and A. Shamir. PayWord and MicroMint: Two simple micropayment schemes. In Security Protocols Workshop, 1997

[9] M. Sirbu and J. D. Tygar. NetBill: An internet commerce system optimized for network-delivered services. IEEE Personal Communications, 2(4):34–39, 1995

[10] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In Workshop on Economics of Peer-to-Peer Systems, 2003.

[11]Back, Adam. "Hashcash-a denial of service counter-measure." (2002).

[12] T. Sander and A. Ta-Shma. Auditable, anonymous electronic cash.In CRYPTO, 1999.

[13] T. Sander, A. Ta-Shma, and M. Yung. Blind, auditable membership proofs. In Financial Cryptography, 2001.

[14] W. Dai. b-money. www.weidai.com/bmoney.txt, 1998

[15] N. Szabo. Formalizing and securing relationships on public networks.First Monday, 2(9), 1997

[16]Pater Wayner, "Digital Cash Commerce on the Net," Academic Press Inc 1996

[17]Digital Cash and Net Commerce. http://www2.pro-ns.net/~crypto/toc12.html

[18] Bonneau, Joseph, et al. "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies." (2015).

[19] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." *Consulted* 1.2012 (2008): 28.

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

[21] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology , vol 3, no2, pages 99-111, 1991.

[22] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[23] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[24] N. T. Courtois, M. Grajek, and R. Naik. Optimizing sha256 in bitcoin mining. In Cryptography and Security Systems, 2014.

[25] M. Rosenfeld. Analysis of Bitcoin Pooled Mining Reward Systems. Technical report, CoRR, 2011.

[26] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In IEEE P2P, 2013

[27] Rosenfeld, Meni. "Analysis of Bitcoin pooled mining reward systems." arXiv preprint arXiv:1112.4980 (2011).

[28] blockchain.info: Bitcoin market capitalization. http://blockchain.info/charts/market-cap, retrieved Oct. 2013

[29] Eyal, Ittay, and Emin Gün Sirer. "Majority is not enough: Bitcoin mining is vulnerable." *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 2014. 436-454.

[30]Courtois, Nicolas T., and Lear Bahack. "On subversive miner strategies and block withholding attack in bitcoin digital currency." arXiv preprint arXiv: 1402.1718 (2014).

[31] Swanson, E.: Bitcoin mining calculator. http://www.alloscomp.com/bitcoin/calculator, retrieved Sep. 2013

[32] Nowak, Martin, and Karl Sigmund. "A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game." Nature 364.6432 (1993): 56-58