Distributed Consensus Technologies in Cryptocurrency Applications

Francisco Rivera

July 15, 2014

Abstract

Today, most services are provided by centralized institutions: for example a bank, a credit card company, or a social media site. However, using a centralized authority requires *trust*. With the innovative idea of the blockchain as implemented in the Bitcoin network, distributed institutions that do not require trust become a feasible possibility. We explore the ideas behind Bitcoin's revolutionary security along with the possibility of the so called "51% attack" on the network. We also review the innovation present in a couple of the hundreds of alt-coins whose basic workings greatly resemble those of Bitcoin, with some crucial differences. Finally, we propose a new, useful alt-coin, NDCoin, along with a hypothesis as to the importance of distributed consensus technologies in the future.

Contents

List of Figures

1 Bitcoin

No discussion of cryptocurrencies or related technologies would be complete without first mentioning quite possibly the most revolutionary digital coin we have seen: Bitcoin.

1.1 What is Bitcoin?

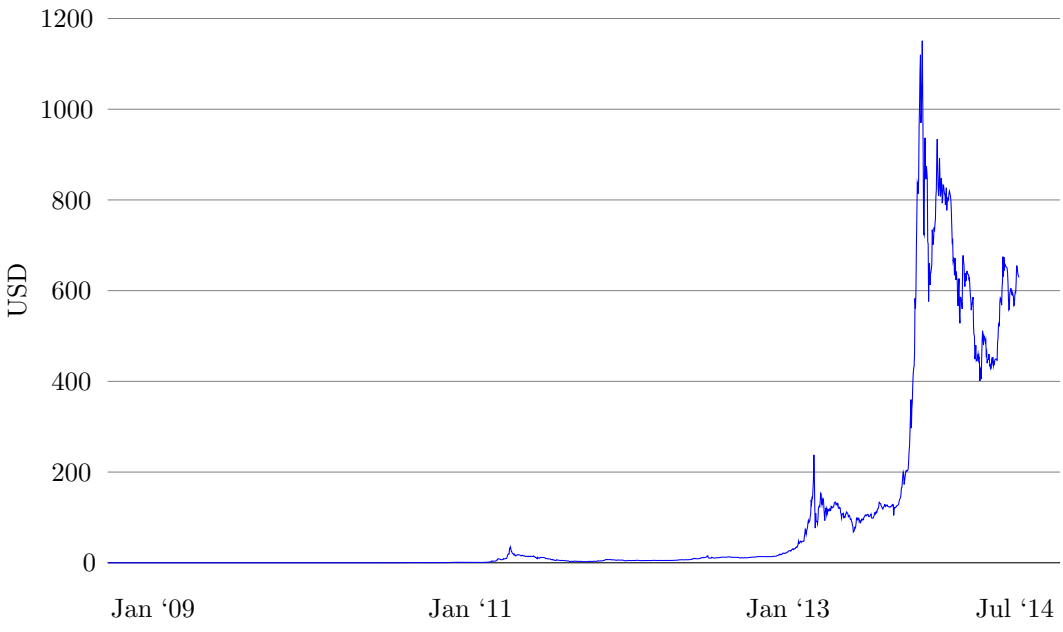
Bitcoin is a new currency. However, unlike traditional currencies, it's legitimacy does not come from printing presses but from innovative cryptographic ideas. Notably, it has all of the following properties:

- It operates without a trusted central authority (which is what separates a Bitcoin transaction from swiping a credit or debit card and makes it more like paying in cash).
- It is open-source (which means no one person controls the technology, and anyone can contribute to future development or inspect the present code for themselves).
- Transactions are anonymous[1].
- No trust is required in order to complete a Bitcoin transaction and transactions will be faithfully executed so long as the majority of computational power in the system is honest.

1.2 Bitcoin Economically

We can consider the exchange rate of Bitcoin (BTC) and U.S. dollars (USD) over the past couple of years to realize the tremendous boom Bitcoin has experienced since it's creation in 2009:

Figure 1: Market Price of BTC [14]



In explaining this boom, CNN considers that, "Small businesses may like [Bitcoins] because there are no credit card fees. Some people just buy Bitcoins as an investment, hoping that theyll go up in value" [18]. However, for our purposes, we are more interested with what we can do with Bitcoin-inspired technologies rather than the economic forces at play in giving Bitcoin its incredible appreciation to the US dollar.

[1] Although we will later see this is not *entirely* true, with good practice, Bitcoin can provide much anonymity—significantly more than any bank or credit card would ever offer

1.3 Bitcoin Technologically

Although most people focus their attention on the economics of Bitcoin because of it's sensational growth as an investment, the technology behind it is in many ways revolutionary. Bitcoin solves the Byzantine Generals problem [13] by distributing consensus on the order of transactions and employs modern public key cryptography to create digital signatures so that transactions cannot be faked.

1.3.1 An Overview

The Bitcoin network consists of *nodes*, none of which is more important than any of the others (the network is decentralized). Coins are owned/linked by/to *addresses*, which have a *public* part and a *private* part — as their name suggests, only the owner of an address knows the private key and the public adddress is available to everyone. Bitcoins can be deposited to an address only by knowing it's public address, whereas transferring Bitcoins from an address requires the private key. Addresses can be generated easily and are essentially disposable.

Public key cryptography makes it possible for someone to sign a transaction request with their private key, and have it be verified by everyone as originating only from someone who knows the private key using the public address. Transaction history is stored in a **public** file that every node has a copy of: the *blockchain*. *Miner* nodes work on adding new transactions to the blockchain, for which they are rewarded in new Bitcoins, hence their designation as miners.

1.3.2 Cryptographic Technologies

The Bitcoin network would not be possible without some useful cryptographic functions and algorithms that are called upon. Prominently among them:

- **SHA256** — A cryptographic hash, SHA256 takes in an input string as long or as short as desired and and *hashes* it to a 256 bit *digest* which reveals no information about the original message:

  $$\text{SHA256}(\text{message}) = \text{digest}.$$

  It is a function in the sense that the same message will **always** hash to the same digest. However, as is apparent from the fact that there are an infinite number of possibilities for messages and only a finite number of possible digest (albeit 2^{256} of them), it is not a one-to-one function. That is, it is **impossible** to determine the original message from a digest because multiple messages can make the same digest [5].

  In terms of computational complexity, calculating the hash of a message is a relatively easy task for a computer. However, finding *any* message so that the hash of the message is a given digest is presently an *NP-hard* problem. This essentially means there is no easier way to generate such a message than just checking many different random messages until one works.

- **Elliptic curve public key cryptography** — A form of asymmetric cryptography, public key cryptography is utilized in Bitcoin as part of the ECDS (Elliptic Curve Digital Signatures) required to authenticate a transaction[2]. It essentially consists of two keys (a public and a private one) and two publicly known functions.

  $$f(\text{public key}, \text{plaintext}) = \text{ciphertext}$$

  $$g(\text{private key}, \text{ciphertext}) = \text{plaintext}$$

  whose security is based on group theory calculations on the family of elliptical curves $y^2 = x^3 + ax + b$ not over the real numbers, but over a different field. This allows Alice to publish a public key and have anyone send her a message only she can decode because she has the private key. Although similar public key cryptography can be achieved through RSA, elliptic curve cryptography offers equal security with smaller key sizes as demonstrated by data provided by the National Institute of Standards and Technology shown in figure 2 (top of the following page) [3]:

[2] We consider the signing algorithm used next and focus only on the asymmetric cryptography aspect now

Figure 2: NIST Recommended Key Sizes (in bits)

Symmetric Key Size	RSA Key Size	Elliptic Curve Key Size
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

- **Digital Signatures (ECDS)** — For purposes of transactions it is crucial to be able to verify that a transaction request indeed come from the person who owns the money. Keeping in mind our ability to make use of public key cryptography (namely the two functions and the two keys) a way to digitally sign a message becomes apparent. Supposing Alice wishes to sign the message msg, she can use her private key to get:

$$g(\text{private key}, \text{msg}) = \text{msg}'.$$

If she publishes both msg and msg' then anyone can check that the message originated from Alice by confirming whether

$$f(\text{public key}, \text{msg}') \stackrel{?}{=} \text{msg}$$

1.3.3 Transactions

A transaction consists of a request sent to the system to transfer a number or fraction of Bitcoins (which must be an integer multiple of a *Satoshi*, the smallest unit of divisibility, 10^{-8} of a Bitcoin) from one address to another. The request is publicly sent out and signed by the private key of the address funds are being removed from, which every node individually verifies for itself. The coins being transferred have the new ownership added on to them by the nodes in the system creating a transaction chain.

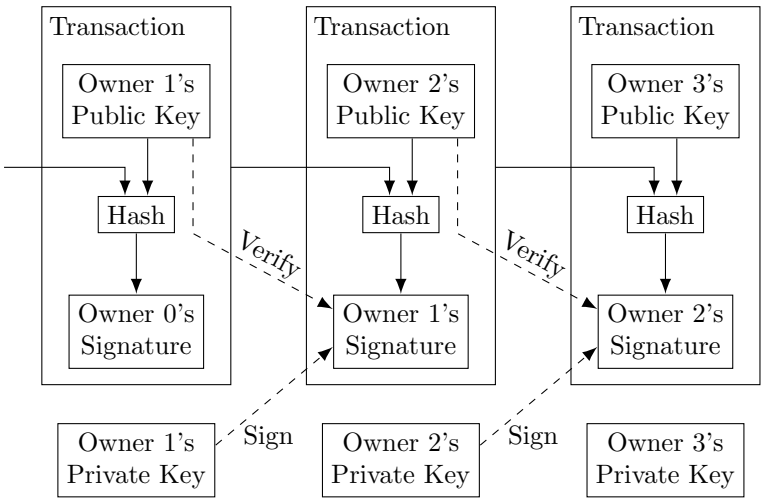


Figure 3: Transaction Chain [15]

In this sense, a Bitcoin, if one could be loosely said to exist, is simply a record of a series of transactions hashed on top of each other and signed using private keys at each step. The transaction is considered after it has been incorporated into a certain number of blocks in the block chain; the more blocks the lesser the probability of the transaction potentially being taken back by a malicious node with significant computational power.

1.3.4 The Block Chain

Although digital signatures and the transaction chain ensure that **only** the owner of Bitcoins can transfer them to another address, this technology alone would be vulnerable to the double spending problem. Although only the owner of funds can transfer them, he could, in theory using only this technology, create a fork in the transaction chain giving the same coin to two different people essentially creating money at will.

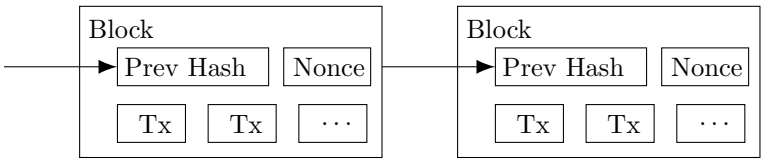
To fix this problem, the Bitcoin network only accepts the first transaction as legitimate. This means that if the coin has already been spent, then it cannot be spent again, which is in accordance with the way we think of actual money. However, this creates the problem of coming to a consensus of the order in which transactions occurred. We could simply ask each node to time stamp every transaction they make, but a malicious node could easily forge a timestamp and progressively insert transactions further back in time with disastrous consequences.

The solution is the **block chain**, which is, as it's name suggests, a chain of blocks. Each block consists of three main parts:

- A hash (SHA256) of the previous block — This ensures that blocks have a well-defined order of existence. Because the content of one block depends recursively on all previous blocks, all of them must have existed in an unmodified form upon the creation of the new one.
- A group of transactions — These are the information that the block chain wishes to sort by chronological order. Supposing block n comes later in the chain than block m, it must follow that transactions stored in block m must have been made before transactions stored in block n.
- A nonce — The nonce, which is just a random integer, serves as a *proof of work* for the block. A block is only considered valid if its hash starts with a certain number of 0's. This mean that miner nodes are constantly checking millions of different nonces to find one that satisfies the leading 0's criteria. This has two important implications:
  - A node cannot create a block at will, and as long as the majority of computational power of the system is in the hands of honest nodes, it is more likely that an honest node will submit an addition to the block chain.
  - A block cannot be tampered with, because doing so would require finding a different nonce to satisfy the leading 0's criteria, a very computationally intensive task.

Graphically, the block chain is depicted in the Bitcoin whitepaper:

Figure 4: Block Chain [15]



Honest nodes in the network are programmed to work on adding **only** to the longest block chain that exists and are rewarded with 25 Bitcoins if they generate a block accepted by the network. Also, the difficulty (the number of leading 0's required in the hash of the block) adjusts itself so that the network produces about one block every ten minutes on average. This means that if a malicious node wanted to "redefine transaction history", it would have to redo the proof of work for all the blocks that have been made from the moment it wishes to alter (insert/delete a transaction/modify a transaction with the necessary private key) faster than the present block chain grows. If the malicious node has less than 50% of the network's computational power, the probability of ever succeeding decreases exponentially with the time into the past the node wants to alter. In the next section, we consider the possibility of what is called a 51% attack.

1.4 Pools and the 51% Attack

By providing a distributed consensus system, the Bitcoin network can be shown to be secure without the necessity of trusting any one entity so long as they do not control the majority (51% or more) of the hashing power of the system. However, this possibility is not entirely impossible and we explore the implications of pools to this so called "51% attack".

1.4.1 Pools

Although it may seem inconceivable to us that any one individual can control a node with enough computational power to significantly rival that of the entire system which consists of an incredible number of computers, the growth of the network is a double edged sword. Because as the network has grown, the difficulty has gone up, any individual with an average computer cannot realistically hope to ever successfully mine a block.

This has led to the creation of *pools*, which are groups of miners that share computational power and split the reward if any of them every successfully mine a block. In order for these pools to work, there must be an an entity that organizes all the worker nodes that wish to be a part of the pool. However, some of these pools have gained great popularity achieving a significant proportion of the total hash power of the system. The exact proportions of hashing power per pool vary wildly, but below is a a representative instance of the distribution:

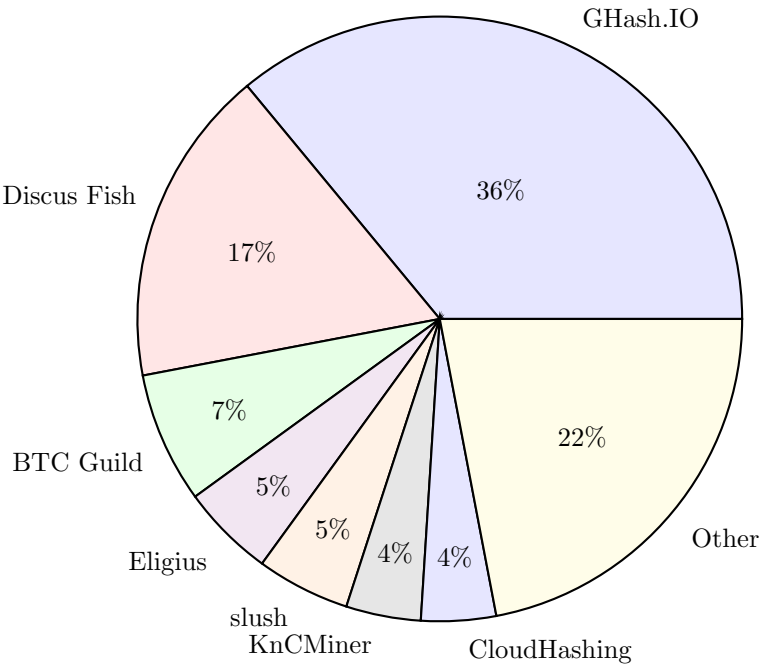


Figure 5: Hashrate Distribution as of July 8, 2015 11:00AM [12]

1.4.2 The Upside

The above snapshot of hashrate distribution does not show any one party controlling more than 51% of the network's power even though this has been known to happen for short periods of time in the past. Although two entities together (namely Discus Fish and GHash.IO in this case) control 53% of the network's power, we claim a pool (or a small number of pools) controlling a majority of hashing power is not a severe threat to the integrity of the network.

If colluding parties that control the majority of the system's hashing power decided to use it for malicious purposes, they would be able to "go back in time" and take back any Bitcoins that they may have spent. This has three severe limitations:

1. By virtue of the ECDS required to initiate a transaction, they would not be able to do anything with Bitcoins that have never been in their possession.
2. Because they would have to construct a parallel blockchain and all blockchains are public in the network, any third party inspecting the blockchain could find proof of the illegitimate transactions inserted.
3. Say an ill-intentioned entity controls $M\%$ of the network hashing power where $M > 50$. Supposing that they begin generating illegitimate blocks, on average the fake blockchain will only be longer by:

   $$M\% - (100\% - M\%) = (2M - 100)\%$$

   meaning if the malicious entity has been working for time Δt after time t_0, we can expect them to have a successful blockchain that can modify transactions as far as:

   $$(0.02M - 1)\Delta t$$

   before time t_0.

   To put this into perspective, assuming Discus Fish and GHash.IO maintain the above level of hashing power for a month and use it entirely for a malicious blockchain (both of which are exorbitant assumptions, because workers would realize the pool wasn't actually hashing for the legitimate blockchain) they would be able to modify transactions as far back as:

   $$(0.02 \cdot 53 - 1)(1 \text{ month}) \cdot \frac{31 \text{ days}}{1 \text{ month}} \approx 2 \text{ days}$$

   which is a very short time in the grand scheme of the network.

Of these difficulties to the attack, 1 and 3 imply the attackers can only take back coins they have spent a short period of time in the past, limiting the profitability of such an attack. Furthermore, the fact that the tampering would be noticed (point number 2) would lead to a huge distrust in not only the pools but Bitcoin as a whole. Seeing as mining is a lucrative business, and the potential profit to an attack is not enormous, it seems against the pools' best interests to consider such an attack.

1.4.3 The Downside

Even though malicious pools would probably harm themselves more than they would profit in the long run, the fact that they *could* undermine the security of the entire system defeats the general purpose of Bitcoin to create a currency that does not require trust in a central authority. After all, what is the difference in trusting a bank or a couple of large mining pools[3]?

A large part of the argument against the danger of the 51% attack lies in the fact it would probably not be financially incentivized. However, this ignores the possibility of an entity that would like to harm the Bitcoin network rather than profit off it. This is not hard to imagine should Bitcoin become a threat to other financial institutions. Either by hacking or more political means, this entity could take control of one or two large pools and conduct a simple 51% attack. This could easily destroy public confidence in the system, plunging the price of Bitcoin and filling the market with instability.

Although this is a worst case scenario which may never happen, we must always keep this possibility in mind when developing cryptocurrencies with alternate purposes that may store more sensitive information in their blockchains.

[3]Although, granted, there are severe limitations to the possible abuse of power of mining pools, primarily the fact that others could detect the tampering

2 Alt-Coins

Bitcoin effectively pioneered an incredible technology with the implementation of the blockchain, and naturally, this idea has been innovated into the creation of a vast array of cryptocurrencies. We consider the top 5 cryptocurrencies by market capitalization[4]:

Figure 6: Cryptocurrencies Market Cap as of July 9, 2014 at 11:37PM [4]

Name	Market Cap (USD)	Price (USD)	Available Supply
Bitcoin	$8,120,541,317	$624.63	13,000,500 BTC
Litecoin	$231,459,399	$7.70	30,068,800 LTC
Nxt	$47,944,256	$0.047944	999,997,000 NXT
Darkcoin	$31,524,434	$7.08	4,455,190 DRK
Ripple	$30,582,022	$0.003912	10^{11} XRP

Although as of the time the data on the table was retrieved, Bitcoin accounts for more than 93% of the market capitalization of all 400 cryptocurrencies indexed by Coin Market Cap [4], a wealth of innovation can be found by looking through the inner workings of these cryptocurrencies. For the purposes of this paper, we consider Gridcoin and Ethereum, each of which can provide us with inspiration for the future development of useful cryptocurrencies.

2.1 Gridcoin

Gridcoin describes itself on its website as, "a new peer-to-peer cryptocurrency that uses distributed computing (BOINC) to benefit humanity by advancing the progress of medicine, biology, climatology, mathematics, astrophysics, and more" [17].

2.1.1 Purpose

An incredible amount of computing power is spent on proof of work (hashing) in the Bitcoin network. Although the system could not work without requiring this amount of computing power, it could still be said that this is wasteful. On the other hand, BOINC (Berkeley Open Infrastructure for Network Computing) is UC Berkeley's project to distribute computing power across volunteers' computers for computation intensive projects [2].

Gridcoin attempts to match a need for computing power with a "waste" of it by incentivizing participation in whitelisted BOINC projects as part of the mining process for Gridcoin.

2.1.2 New Technologies

A nontrivial dfference between the proof of work of Bitcoin and Gridcoin (other than the integration of BOINC which we explore in subsequent sections) is the cryptographic hash used. Bitcoin uses SHA256 (explained in section 1.3.2) whereas Gridcoin (along with other cryptocurrencies such as Litecoin and Dogecoin) use a different one called Scrypt.

The purpose of a hash is that going from the message to the digest is "easy" whereas going from the digest to the message is impossible and generating *any* message which could have created a digest is hard. However, in Scrypt, getting the digest from the message is a significantly harder process than it is for SHA-256 taking on the order of several hundred milliseconds and "large" amounts of memory on a standard computer.

Significantly, this raises the hardware requirements to mine Gridcoin encouraging participation in BOINC because it is an easier computational task which can provide an equivalent financial reward, and discouraging the use of SHA256 application specific integrated circuits from crowding out average users.

[4]For a cryptocurrency, this is evaluated as (number of coins in circulation) × (market value of a coin)

2.1.3 How it Works

There are two ways to mine a block:

- GPU Mining consists of a Scrypt-based proof of work
- CPU Mining consists of providing clock cycle credits of work to a whitelisted BOINC project

with the restriction that CPU mining cannot be greater than 50% of all generated blocks. However, all miners (even GPU miners) must contribute at least 100 RAC to BOINC, where a RAC or Recent Average Credit is a variable unit of measurement for how much work is done on BOINC projects which depends on how much work nodes on the system have recently done on a project [11].

For a CPU miner, the reward for successfully mining a block is variable and equals

$$\text{reward} = \frac{\text{Node's Project RAC}}{\text{Individual Project's RAC}} \cdot 150 \text{ GRC}$$

with the condition that this value not fall under 5 GRC or exceed 150 GRC and a maximum of 168 million gridcoins created by the system [17]. With the exception of these differences, the Gridcoin network works very much in the same way as the Bitcoin network does with signed transactions and the blockchain.

2.1.4 What Gridcoin is Not

Even though Gridcoin marks a promising idea to utilize proof of work computation for a computational task with other applications, it is **not** an effective way to hire distributed computation.

- Although it is very easy to create a BOINC project, projects must be whitelisted to be integrated into Gridcoin which presents a serious difficulty to entry and a third party reviewing a computational task before it can be outsourced to the system.
- Because the monetary reward comes from newly "minted" coins at a predetermined rate, project involvement by the network is essentially decided by what projects individuals wish to contribute to, and computation cannot be paid for.
- There are no cryptographic guarantees as to the correctness of computation results (even though the security offered is sufficient for BOINC projects which do not work with sensitive information by nature).

These are not necessarily shortcomings of Gridcoin, but illustrate the scope for which Gridcoin is useful, leaving the door open for future innovation in the form of new cryptocurrencies.

2.2 Ethereum

Although it has not yet been released at the time of writing of this paper[5], Ethereum claims to be, "a platform and a programming language that makes it possible for any developer to build and publish next-generation distributed applications" [6].

2.2.1 Purpose

Ethereum developers recognize the undesirable nature of centralized authorities that require trust in order to function and may ultimately abuse this trust. Using the distributed consensus idea of the blockchain, they are developing a platform to easily create systems that are secure as well as non-centralized. In this sense, Ethereum is different from all the other alt-coins because it is not just an alt-coin (even though it does have a coin, the Ether), but a tool to develop distributed networks (such as cryptocurrencies) on top of the Ether.

[5]Release for Ethereum is planned for sometime during the fourth quarter of 2014

2.2.2 How it Works

Although it is too early to talk about the particulars of Ethereum because it hasn't been made public yet, the developers have published a proof of concept which provides insight into the workings of Ethereum. By allowing developers access to a Turing-complete language on top of Ethreum to create "smart contracts", Ethereum could in theory be used to create any distributed network with a lot less code than would be required to start from scratch.

Ethereum creates two different types of accounts:

- **An Externally Owned Account** — This is the familiar type of account one would see in a typical cryptocurrency.
- **A Contract Account** — This account is not owned by anyone, and its behavior is completely dictated by the account's *contract code*. These are the accounts that allow Ethereum to create smart contracts and gain functionality from developers.

Each account has a nonce which is used to avoid double spending and an ether balance. It may also have storage, and if it is a contract account, will have contract code.

The Ethereum whitepaper outlines three important differences between Bitcoin transactions and their Ethereum counterpart, *messages* [7]:

1. Messages can originate from any type of account, meaning that not only individuals, but also contract code is capable of sending messages.
2. Ethereum messages have the, "explicit option. . . to contain data" [7].
3. If a contract account receives a message, it has the option of a response.

In Ethereum, a message contains a couple of parts to it:

- The recipient
- A digital signature of the sender
- What is being sent (amount of ethers, and data)
- STARTGAS and GASPRICE — Because every message to a contract account could in theory create another message somewhere else, we could easily see this process falling into an infinite loop or just a long sequence of steps. To prevent abuse of the network in this way, the sender must stipulate the limit on the number of computational steps his messageg will spawn (STARTGAS) and the price he is wiling to pay for each of these steps (GASPRICE). If the STARTGAS value is exceeded, all changes to the state of a network because of a message are reversed and the message stops running.

The basic principles of how Ethereum functions can be simply represented by a depiction included in the whitepaper of a transaction changing states of accounts:

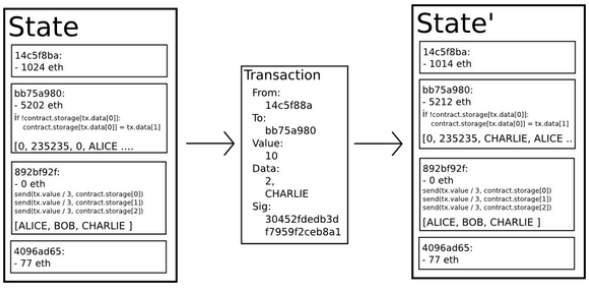


Figure 7: Representation of Ethereum Transaction [7]

2.2.3 Downsides?

Ethereum attempts an ambitious undertaking which if successful could greatly reduce the time and effort necessary to develop future decentralized networks. Ethereum has big shoes to fill with all the potential advertised for it and only time will tell whether it indeed fills these shoes. Notably however, the cryptocurrency community has already expressed discontent at Ethereum's *pre-mining* [16]. This means that the Ethereum developers will take some of the cryptocurrency for themselves **before** the network is online for everyone else, and is upsetting for obvious reasons, even if there is debate on whether this is a merited reward or an abuse of power.

3 Proposed Cryptocurrencies

Having studied Bitcoin's technology along with the innovation provided by some of the hundreds of new cryptocurrencies that have emerged, we consider the concept of a new currency: NDCoin.

3.1 NDCoin

Most new cryptocurrencies still serve the purpose of a fiat currency, albeit with bells and whistles over Bitcoin. In most cases the coin lacks intrinsic value (even in Gridcoin, in which *mining* the coin has value contributing to BOINC, but the coin itself is still just a fiat currency). We consider a new cryptocurrency who's value stems from the network's computation power. We could spend this coin to have the network do distributed, nondeterministic (ND) computation for us at a competitive price. Quoting the description in the abstract of the whitepaper, NDCoin functions as,

> "a cryptocurrency platform that can also be used as a trustworthy distributed marketplace for robustly carrying out high-performance computing tasks, such as, in particular, for solving instances of difficult (e.g., NP-complete, or apparently exponentially hard) problem classes that are in the complexity class NP (nondeterministic polynomial time) in massively-parallel fashion, or, more generally, for harnessing the network to efficiently contract out the execution of any desired nondeterministic (ND) algorithms including, in an extended version, algorithms that do not essentially rely on nondeterminism." [9]

3.1.1 Cryptographic Technologies

The development of NDCoin will require some additional technologies to those used in Bitcoin, among them:

- **Bit Commitment** — Suppose that a party wishes to prove that they had possession of some sort of information at some time in the past without revealing the information until later. The act of proving the possession of information without revealing its content is called *bit commitment*.

  To do this the party can hash the information (say using SHA256 or Scrypt) and digitally sign it (with ECDS for example) then publish the result. At a later date, the information can be revealed and anyone else can hash the information to check that the party indeed had it when they claimed they did.
- **zk-SNARK** — A **S**uccinct **N**on-interactive **Arg**ument is called a *SNARG* for short. A **SNAR**G of **K**nowledge in turn is called a *SNARK*, and if this SNARK produces **z**ero **k**nowledge, then we call it a *zk-SNARK* [1].

  Suppose that for simplicity, we have two nodes. One node is a computationally bounded worker, i.e. this node will perform computational tasks for the other node; call this node the *worker*. However, the other node (call it the *customer*) does not trust this worker, so the worker must prove it has correctly done the computational tasks according to three requirements:

  – The argument must provide zero knowledge to the customer.
  – The argument must be publicly verifiable meaning any third party can check it's validity.
  – The argument must be less complex than the calculation the worker is proving, else the customer would be no better off than performing the computation himself.

A zk-SNARK has three main functions:

1. **The Key Generator** — Unfortunately, this function requires expensive preprocessing in order to generate the public parameters necessary to use a zk-SNARK given a random input λ:
   - A Proving Key
   - A Verification Key

   Fortunately, public parameters may be used multiple times without a loss of security, lessening the cost of the initial computation over time.
2. **The Prover** — Given that the public parameters have already been generated, the worker node can run the prover to generate a proof of work on whatever computational task it wishes. A regrettable downside of running the prover is that the run-time of the computational task with the prover is larger than the run-time of the computational task without the prover by a factor proportional to the logarithm of the number of steps in the computational task.
3. **The Verifier** — After the prover has run, the verifier can be run by anyone given the inputs of the proof of work and the public parameters to check that the calculations were indeed performed. The verifier will not given any information as to the result of the calculations that were performed, and is "succinct", meaning checking isn't nearly as hard as performing the actual calculation.

- **Homomorphic Encryption** — We suppose that a node has a calculation that it wishes to hire out to another worker node. However, this calculation may involve sensitive information that should be kept secret. While it may initially seem that the first node that wants to keep the calculation private must execute it itself, *homomorphic encryption* allows for this calculation to still be hired out without revealing any information. Formally, homomorphic encryption allows certain operations to be performed on ciphertext which correspond to other operations on the plaintext.

- **Indistinguishability Obfuscation** — Although this technology is still in its early stages, it is theoretically possible to encrypt a computer program in such a way that its code remains impossible to recover, yet a third party can run it [10]. The application of this possibility to the concept of NDCoin is self evident, but the process of encryption makes the new program much longer than the original, comparable to an, "unwieldy albatross" [10]. So, while this would be inconvenient to immediately incorporate into NDCoin, it remains a promising development for the future.

3.1.2 The NDCoin Concept

NDCoin would in principle work very much like Bitcoin, especially with a practically identical blockchain implementation except for possibly the use of Scrypt instead of SHA256. However, the way transaction requests work would be modified for an important difference. Bitcoin transactions are complete as they are requested (i.e. they have all the information that they need to be executed when first put into the system). However, nothing prevents us from creating a system in which transactions do not necessarily have to be complete, particularly in the send to address field. We consider the necessary fields of NDCoin transaction which is used to pay for computation:

- A non-deterministic computation which the customer wishes to pay for successful completion.
- quantity of NDCoins — size of the bounty for successfully completing this computation.
- time limit — the maximum amount of time that is allowed to pass before someone claims the bounty. If this time limit is exceeded, the bounty is returned to the customer.
- number of bounties — the number of different solutions that the customer is willing to pay for.
- a digital signature — proof that the transaction request is coming from someone with possession of the private key of the funds.

If a worker node wishes to claim the reward, it must:

1. Successfully run the computation and generate a proof, which can take the form of a witness string or a zk-SNARK depending on its complexity.

2. Post a bit commitment of the proof to the blockchain.
3. Reveal proof to the network and specify send to address

In order for a transaction to take place, the network will

- Check the signature on the posted transaction to ensure its legitimacy.
- Either:
  - Check that the time limit has been exceeded and deliver funds back to the same address they originate from.
  - Check the witness string or zk-SNARK for the first bit commitment posted to the blockchain, and if valid, transfer funds to address specified by worker node.

3.2 Vote Coin

Although we would like to believe in the legitimacy of our elections, in reality, we have no strict guarantee against fraud. Indeed, in developing countries, blatantly fraudulent elections are an all too common event. However, fraud is only possible because vote counting is done by a centralized institution and and can be improved by a distributed consensus system.

3.2.1 Cryptographic Technologies

An implementation of VoteCoin would only require one more technology than the ones we have already familiarized ourselves with:

- **Blind signatures** — Similar to digital signatures, a blind signature proves that a message was approved by some private key. However, in contrast to digital signatures (which blind signatures use), a blind signature proves that a third party approves a message another is sending without being able to read or modify it.

  We consider how Alice could send a message signed by Bob without letting Bob modify of read it. First, Alice encrypts her message:

  $$f(\text{blind}, \text{message}) = \text{blindedMessage}$$

  and sends the blinded message to Bob. Bob can digitally sign this blinded message, say using ECDS. Once the message has been signed, Alice can publish her blinding key and message. However, an important requirement in this process is that the blinded message should not result in multiple *valid* messages depending on the blinding key used.

  Although this completely describes the process of a blind signature technically, we can better understand it through an analogy: Alice still as a message she wants Bob to sign, so she puts it in an envelope with carbon paper. Bob then signs the envelope and indirectly signs the message without ever knowing what it was:

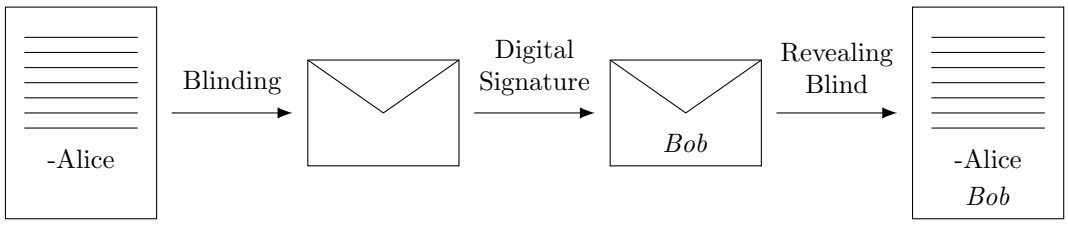


Figure 8: Blind Signature Representation

3.2.2 Votecoin Requirements

We would like to building a crypto"currency" to implement a voting procedure just as in a regular election, except with the following desirable properties [8]:

- Every voter can verify their vote was counted and can independently arrive at the published results of the election to confirm them.
- Votes are confidential
- Any possible cheating in the system can be proved

3.2.3 Votecoin Concept

Other than using a blockchain, Votecoin would have more differences than similarities with the way Bitcoin, which makes it easier to outline the procedure of holding an election rather than compare to Bitcoin [8]:

1. Registrar publishes public list of eligible voters (identified by some unique public ID)
2. The list is inspected by the public and corrections can be made as deemed appropriate.
3. Registrar digitally signs final version of the voter list.
4. Voters generate a "'fingerprint' code" using personal, possibly biometric, information and send it to the registrar.
5. The registrar then assigns every valid public ID a public address (although the network does not know the correspondence), with each key pair tied to a double hash of the fingerprint code.
6. Every voter is then given his private key with which to vote.
7. The voter posts a signed bit commitment to the network on his vote along with the hash of the fingerprint code.
8. Once all bit commitments have been taken into the blockchain, all the voters reveal their votes signed with the public address and they can be tallied up to determine the result of the election

4 Future Development

Whether Bitcoin continues its upward trend in value or crashes never to rise again remains a hotly contended question, but the future of decentralized networks remains much more secure. Because of their ability to provide services which have previously had no choice but to be centralized, distributed consensus networks have a huge market to competitively join with an edge over their centralized counterparts. The security produced by these networks which does not require trust of a single, powerful entity thus eliminating the "choke point" of most systems today may easily make these the technology of tomorrow which is just emerging. Already we have seen the potential of an application to currency, but potential in areas such as voting and cloud computing still remain largely untapped.

References

[1] Ben-Sasson, Eli, et al. "SNARKs for C : Verifying Program Executions Succinctly and in Zero Knowledge." *International Association for Cryptological Research*. N.p., 7 Oct. 2013. Web. 10 July 2014. <https://eprint.iacr.org/2013/507.pdf>.

[2] *BOINC*. UC Berkeley, n.d. Web. 9 July 2014. <http://boinc.berkeley.edu/>.

[3] "The Case for Elliptic Curve Cryptography." *National Security Agency*. United States, 15 Jan. 2009. Web. 7 July 2014. <http://www.nsa.gov/business/programs/elliptic_curve.shtml>.

[4] "Crypto-Currency Market Capitalizations." *Coin Market Cap*. N.p., n.d. Web. 9 July 2014. <http://www.coinmarketcap.com/>.

[5] "Descriptions of SHA-256, SHA-384, and SHA-512." National Institute of Standards and Technology. U.S., n.d. Web. 11 July 2014. <http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf>.

[6] "Ethereum." *Ethereum*. N.p., n.d. Web. 9 July 2014. <https://www.ethereum.org/>.

[7] "Ethereum White Paper." *Github*. N.p., 24 Apr. 2014. Web. 9 July 2014. <https://github.com/ethereum/wiki/wiki/%5BEnglish%5D-White-Paper>.

[8] Frank, Mike. "Votecoin Notes 2." File sent on 10 July 2014. Word file.

[9] Frank, Mike P., and David Mondrus. "Introducing NDcoin." *NDCoin*. Cryptowerks, n.d. Web. 10 July 2014. <https://dl.dropboxusercontent.com/u/3133557/ Bitcoin/Introducing%20NDcoin.pdf>.

[10] Garg, Sanjam, et al. "Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits." International Association for Cryptological Research. N.p., 21 July 2013. Web. 14 July 2014. <http://eprint.iacr.org/2013/451.pdf>.

[11] Halford, Rob. "Gridcoin Whitepaper." *Gridcoin*. N.p., 23 May 2014. Web. 9 July 2014. <http://www.gridcoin.us/images/gridcoin-white-paper.pdf>.

[12] "Hashrate Distribution." *Blockchain*. Ed. Ben Reeves. N.p., 8 July 2014. Web. 8 July 2014. <https://blockchain.info/pools>.

[13] Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." *Carnegie Mellon University*. N.p., 1982. Web. 7 July 2014. <https://www.andrew.cmu.edu/course/15-749/READINGS/required/resilience/lamport82.pdf>.

[14] "Market Price." *Blockchain*. Ed. Ben Reeves. N.p., n.d. Web. 5 July 2014. <https://blockchain.info/charts/market-priceshowDataPoints=falsexpan=all&show_header=true&daysAverageString=1&scale=0&format=csv&address=>.

[15] Nakamoto, Satoshi. "Bitcoin: A Peer-to-Peer Electronic Cash System." *Bitcoin*. The Bitcoin Foundation, 2009. Web. 8 July 2014. <https://bitcoin.org/bitcoin.pdf>.

[16] Torpey, Kyle. "Ethereum Premine Leading to Forks of the Source Code." *Cryptocoins News*. N.p., 2 Mar. 2014. Web. 10 July 2014. <http://www.cryptocoinsnews.com/news/ethereum-premine-leading-forks-source-code/2014/02/03>.

[17] "What is Gridcoin?" *Gridcoin*. N.p., n.d. Web. 9 July 2014. <http://www.gridcoin.us/>.

[18] Yellin, Tai, Dominic Aratari, and Jose Pagliery. "What Is Bitcoin." *CNN*. N.p., n.d. Web. 5 July 2014. <http://money.cnn.com/infographic/technology/what-is-bitcoin/>.