8
Alternative Coins

Since the initial success of Bitcoin, many alternative currency projects have been launched. Bitcoin was released in 2009, and the first alternative coin project (named Namecoin) was introduced in 2011. In 2013 and 2014, the **alternative coins** (**altcoin**) market grew exponentially, and many different types of alternative coin project were started.

A few of those became a success, whereas many were unpopular due to less interest and as a result, they did not succeed. A few were *pump and dump* scams that surfaced for some time but soon disappeared. Alternative approaches to bitcoin can be divided broadly into two categories, based on the primary purpose of their development. If the primary goal is to build a decentralized blockchain platform, they are called alternative chains; if the sole purpose of the alternative project is to introduce a new virtual currency, it is called an altcoin.

Alternative blockchains will be discussed in detail in `Chapter 16`, *Alternative Blockchains.*

This chapter is mainly dedicated to altcoins whose primary purpose is to introduce a new virtual currency (coin) although some material will also be presented on the topic of alternative protocols built on top of bitcoin to provide various services. These include concepts such as Namecoin, where the primary purpose is to provide decentralized naming and identity services instead of currency.

Currently, as of late 2018, there are hundreds of altcoins on the market, and they hold some monetary value such as Namecoin, Zcash, Primecoin, and many others. We will examine some of these later in this chapter. Zcash is a more successful altcoin introduced in 2016. On the other hand, Primecoin did not gain much popularity however it is still used. Many of these alternative projects are direct forks of Bitcoin source code although some of those have been written from scratch. Some altcoins set out to address Bitcoin limitations such as privacy. Some others offer different types of mining, changes in block times, and distribution schemes.

By definition, an altcoin is generated in the case of a hard fork. If bitcoin has a hard fork then the other, older chain is effectively considered another coin. However, there is no established rule as to which chain becomes the altcoin. This has happened with Ethereum, where a hard fork caused a new currency **Ethereum Classic** (**ETC**) to come into existence in addition to the **Ethereum** (**ETH**) currency. Ethereum classic is the old chain and Ether is the new chain after the fork. Such a contentious hard fork is not desirable for some reasons. First it is against the true spirit of decentralization as the Ethereum foundation, a central entity, decided to go ahead with the hard fork even though not everyone agreed to the proposition; second, it also splits the user community due to disagreement over the hard fork. Although a hard fork, in theory, generates an altcoin, it is limited in what it can offer because, even if the change results in a hard fork, usually there are no drastic changes around the fundamental parameters of the coin. They typically remain the same. For this reason, it is desirable to either write a new coin from scratch or fork the bitcoin (or another coin's source code) to create a new currency with the desired parameters and features.

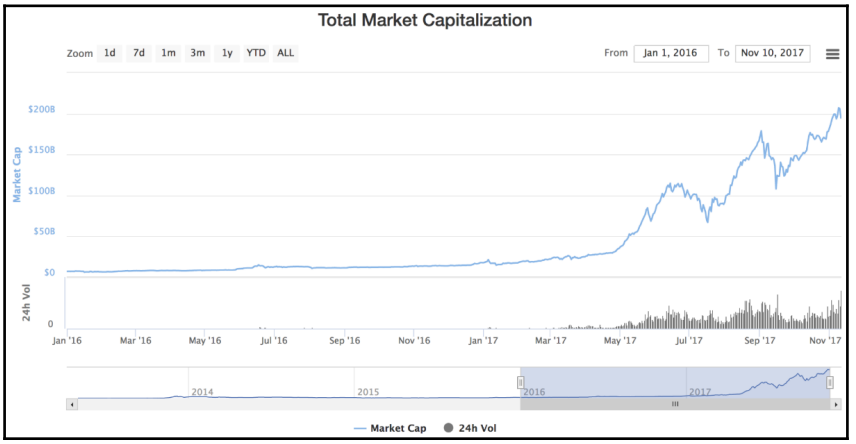
Altcoins must be able to attract new users, trades, and miners otherwise the currency will have no value. Currency gains its value, especially in the virtual currency space, due to the network effect and its acceptability by the community. If a coin fails to attract enough users then soon it will be forgotten. Users can be attracted by providing an initial amount of coins and can be achieved by using various methods. There is, however, a risk that if the new coin does not perform well than their initial investment may be lost. Methods of providing an initial number of altcoins are discussed as follows:

- **Create a new blockchain**: Altcoins can create a new blockchain and allocate coins to initial miners, but this approach is now unpopular due to many scam schemes or *pump and dump* schemes where initial miners made a profit with the launch of a new currency and then disappeared.

- **Proof of Burn (PoB)**: Another approach to allocating initial funds to a new altcoin is PoB, also called a one-way peg or price ceiling. In this method users permanently destroy a certain quantity of bitcoins in proportion to the quantity of altcoins to be claimed. For example, if ten bitcoins were destroyed then altcoins can have a value no greater than some bitcoins destroyed. This means that bitcoins are being converted into altcoins by burning them.
- **Proof of ownership**: Instead of permanently destroying bitcoins, an alternative method is to prove that users own a certain number of bitcoins. This proof of ownership can be used to claim altcoins by tethering altcoin blocks to Bitcoin blocks. For example, this can be achieved by merged mining in which effectively bitcoin miners can mine altcoin blocks while mining for bitcoin without any extra work. Merged mining is explained later in the chapter.
- **Pegged sidechains**: Sidechains, as the name suggests, are blockchains separate from the bitcoin network but bitcoins can be transferred to them. Altcoins can also be transferred back to the bitcoin network. This concept is called a **two-way peg**.

Investing and trading these alternative coins is also a big business, albeit not as big as bitcoin but enough to attract new investors and traders and provide liquidity to the market. Combined altcoin market capitalization is shown as follows:

The graph is generated from `https://coinmarketcap.com/`.



This graph shows that at the time of writing the Combined Altcoin Market Capitalization is more than 200 billion US Dollars

Current market cap (as of March, 2018) of the top 10 coins is shown as follows:

Name	**Market Cap**	**Price USD**
Bitcoin	$151,388,873,045	$8,951.83
Ethereum	$68,472,253,587	$697.94
Ripple	$31,340,920,806	$0.801723
Bitcoin Cash	$17,182,167,856	$1,010.08
Litecoin	$9,952,905,688	$179.11
NEO	$5,638,100,000	$86.74
Cardano	$5,450,310,987	$0.210217
Stellar	$5,438,720,268	$0.294010
EOS	$4,347,501,290	$6.04
Monero	$4,211,690,257	$266.40

The data is taken from `https://coinmarketcap.com/`.

There are various factors and new concepts introduced with alternative coins. Many concepts were invented even before bitcoin but with bitcoin were not only new concepts, such as a solution to the Byzantine Generals' Problem, introduced but also previous ideas such as hashcash and **Proof of Work** (**PoW**) were used ingeniously and came into the limelight.

Since then, with the introduction of alternative coin projects, various new techniques and concepts have been developed and introduced. To appreciate the current landscape of alternative cryptocurrencies, it is essential to understand some theoretical concepts first.

Theoretical foundations

In this section, various theoretical concepts are introduced to the reader that has been developed with the introduction of different altcoins in the past few years.

Alternatives to Proof of Work

The PoW scheme in the context of cryptocurrency was first used in Bitcoin and served as a mechanism to provide assurance that a miner had completed the required amount of work to find a block. This process in return provided decentralization, security, and stability for the blockchain. This is the primary vehicle in Bitcoin for providing decentralized distributed consensus. PoW schemes are required to have a much-desired property called **progress freeness**, which means that the reward for consuming computational resources should be random and proportional to the contribution made by the miners. In this case, some chance of winning the block reward is given to even those miners who have comparatively less computational power.

The term progress freeness was introduced by Arvind Narayanan and others in the book *Bitcoin and Cryptocurrency Technologies*. Other requirements for mining computational puzzles include adjustable difficulty and quick verification. Adjustable difficulty ensures that the difficulty target for mining on the blockchain is regulated in response to increased hashing power and the number of users.

Quick verification is a property which means that mining computational puzzles should be easy and quick to verify. Another aspect of the PoW scheme, especially the one used in Bitcoin (Double SHA-256), is that since the introduction of ASICs the power is shifting towards miners or mining pools who can afford to operate large-scale ASIC farms. This power shift challenges the core philosophy of the decentralization of Bitcoin.

There are a few alternatives that have been proposed such as ASIC-resistant puzzles and are designed in such a way that building ASICs for solving this puzzle is infeasible and does not result in a major performance gain over commodity hardware. A common technique used for this purpose is to apply a class of computationally hard problems called **memory hard computational puzzles**. The core idea behind this method is that as puzzle solving requires a large amount of memory, it is not feasible to be implemented on ASIC-based systems.

This technique was initially used in Litecoin and Tenebrix where the Scrypt hash function was used as an ASIC-resistant PoW scheme. Even though this scheme was initially advertised as ASIC resistant, recently Scrypt ASICs have now become available, disproving the original claim by Litecoin. This happened because Scrypt is a memory intensive mechanism and initially it was thought that building ASICs with large memories is difficult due to technical and cost limitations. This is no longer the case, because memory is increasingly becoming cheaper and with the ability to produce nanometer scale circuits it is possible to build ASICs that can run Scrypt algorithm.

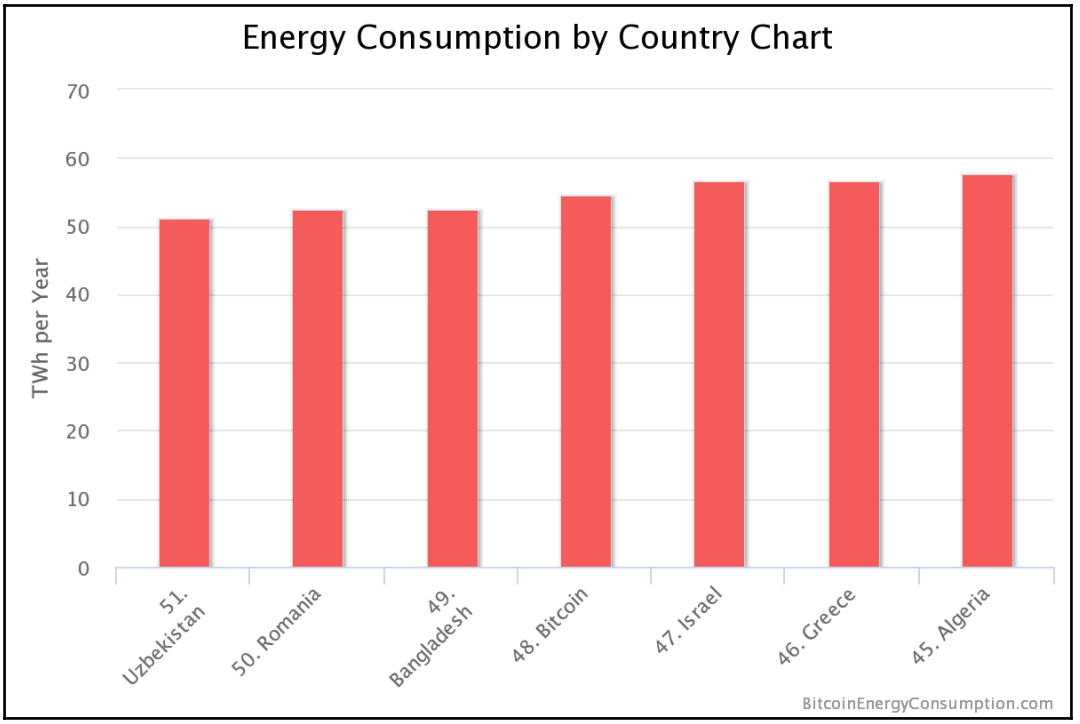
Another approach to ASIC resistance is where multiple hash functions are required to be calculated to provide PoW. This is also called a **chained hashing scheme**. The rationale behind this idea is that designing multiple hash functions on an ASIC is not very feasible. The most common example is the X11 memory hard function implemented in Dash. X11 comprises 11 SHA-3 contestants where one algorithm outputs the calculated hash to the next algorithm until all 11 algorithms are used in a sequence. These algorithms include BLAKE, BMW, Groestl, JH, Keccak, Skein, Luffa, CubeHash, SHAvite, SIMD, and ECHO.

This approach did provide some resistance to ASIC development initially, but now ASIC miners are available commercially and support mining of X11 and similar schemes. A recent example is ASIC Baikal Miner, which supports X11, X13, X14, and X15 mining. Other examples include miners such as iBeLink DM384M X11 miner and PinIdea X11 ASIC miner.

Perhaps another approach could be to design self-mutating puzzles that intelligently or randomly change the PoW scheme or its requirements as a function of time. This strategy will make it almost impossible to be implemented in ASICs as it will require multiple ASICs to be designed for each function and also randomly changing schemes would be almost impossible to handle in ASICs. At the moment, it is unclear how this can be achieved practically.

PoW does have various drawbacks, and the biggest of all is energy consumption. It is estimated that the total electricity consumed by Bitcoin miners currently is more than that of Bangladesh at 54.69 **Terawatt hash** (**TWh**). This is huge, and all that power is in a way wasted; in fact, no useful purpose is served except mining. Environmentalists have raised real concerns about this situation. In addition to electricity consumption, the carbon footprint is also very high currently estimated at around 387 kg of CO_2 per transaction.

The following graph shows the scale of Bitcoin energy consumption as compared to other countries. This is only expected to grow and it is estimated that by the end of year 2018, the energy consumption can reach approximately 125 TWh per year.



Energy consumption by country

The preceding graph shown is taken from the website which tracks this subject. It is available at `https://digiconomist.net/bitcoin-energy-consumption`.

It has been proposed that PoW puzzles can be designed in such a way that they serve two purposes. First, their primary purpose is in consensus mechanisms and second to perform some useful scientific computation. This way not only can the schemes be used in mining but they can also help to solve other scientific problems too potentially. This proof of useful work has been recently put into practice by Primecoin where the requirement is to find special prime number chains known as Cunningham chains and bi-twin chains. As the study of prime number distribution has special significance in scientific disciplines such as physics, by mining Primecoin miners not only achieve the block reward but also help in finding the special prime numbers.

Proof of Storage

Also known as proof of retrievability, this is another type of proof of useful work that requires storage of a large amount of data. Introduced by Microsoft Research, this scheme provides a useful benefit of distributed storage of archival data. Miners are required to store a pseudo, randomly-selected subset of large data to perform mining.

Proof of Stake (PoS)

This proof is also called virtual mining. This is another type of mining puzzle that has been proposed as an alternative to traditional PoW schemes. It was first proposed in Peercoin in August 2012. In this scheme, the idea is that users are required to demonstrate possession of a certain amount of currency (coins) thus proving that they have a stake in the coin. The simplest form of the stake is where mining is made comparatively easier for those users who demonstrably own larger amounts of digital currency. The benefits of this scheme are twofold; first acquiring large amounts of digital currency is relatively difficult as compared to buying high-end ASIC devices and second it results in saving computational resources. Various forms of stake have been proposed and are briefly discussed in the following subsection.

Various stake types

Different type of stakes will now be introduced in the following subsections.

Proof of coinage

The age of a coin is the time since the coins were last used or held. This is a different approach from the usual form of PoS where mining is made easier for users who have the highest stake in the altcoin. In the coin-age-based approach, the age of the coin (coinage) is reset every time a block is mined. The miner is rewarded for holding and not spending coins for a period of time. This mechanism has been implemented in Peercoin combined with PoW in a creative way.

The difficulty of mining puzzles (PoW) is inversely proportional to the coinage, meaning that if miners consume some coinage using coin-stake transactions, then the PoW requirements are relieved.

Proof of Deposit (PoD)

The core idea behind this scheme is that newly minted blocks by miners are made unspendable for a certain period. More precisely the coins get locked for a set number of blocks during the mining operation. The scheme works by allowing miners to perform mining at the cost of freezing a certain number of coins for some time. This is a type of PoS.

Proof of Burn

As an alternate expenditure to computing power, PoB, in fact, destroys a certain number of bitcoins to get equivalent altcoins. This is commonly used when starting up a new coin projects as a means to provide a fair initial distribution. This can be considered an alternative mining scheme where the value of the new coins comes from the fact that previously a certain number of coins have been destroyed.

Proof of Activity (PoA)

This scheme is a hybrid of PoW and PoS. In this scheme, blocks are initially produced using PoW, but then each block randomly assigns three stakeholders that are required to digitally sign it. The validity of subsequent blocks is dependent on the successful signing of previously randomly chosen blocks.

There is, however, a possible issue known as the nothing at stake problem where it would be trivial to create a fork of the blockchain. This is possible because in PoW appropriate computational resources are required to mine whereas in PoS there is no such requirement; as a result, an attacker can try to mine on multiple chains using the same coin.

Nonoutsourceable puzzles

The key motivation behind this puzzle is to develop resistance again the development of mining pools. Mining pools as previously discussed offer rewards to all participants in proportion to the computing power they consume. However, in this model the mining pool operator is a central authority to whom all the rewards go and who can enforce specific rules. Also, in this model, all miners only trust each other because they are working towards a common goal together in the hope of the pool manager getting the reward. Nonoutsourceable puzzles are a scheme that allows miners to claim rewards for themselves; consequently, pool formation becomes unlikely due to inherent mistrust between anonymous miners.

There are also various other alternatives to PoW, some of which have been described in `Chapter 1`, *Blockchain 101* and some will be explained later in this book in `Chapter 15`, *Hyperledger* and `Chapter 18`, *Scalability and Other Challenges*. As this is an ongoing area of research, new alternatives will keep emerging as blockchain technology grows.

Difficulty adjustment and retargeting algorithms

Another concept that has been introduced with the advent of bitcoin and altcoins is difficulty in retargeting algorithms. In bitcoin, a difficulty target is calculated simply by the following equation; however other coins have either developed their algorithms or implemented modified versions of the bitcoin difficulty algorithm:

*T = Time previous * time actual / 2016 * 10 min*

The idea behind difficulty regulation in bitcoin is that a generation of 2016 blocks should take roughly around two weeks (inter-block time should be around 10 minutes). If it takes longer than two weeks to mine 2016 blocks, then the difficulty is decreased, and if it takes less than two weeks to mine 2016 blocks, then the difficulty is increased. When ASICs were introduced due to a high block generation rate, the difficulty increased exponentially, and that is one drawback of PoW algorithms that are not ASIC resistant. This leads to mining power centralization.

This also poses another problem; if a new coin starts now with the same PoW based on SHA-256 as bitcoin uses, then it would be easy for a malicious user to just simply use an ASIC miner and control the entire network. This attack would be more practical if there is less interest in the new altcoin and someone decides to take over the network by consuming adequately high computing resources. This may not be a possible attack if other miners with similar computing power also join the altcoin network because then miners will be competing with each other.

Also, multipools pose a more significant threat where a group of miners can automatically switch to the currency that is becoming profitable. This phenomenon is known as **pool hopping** and can adversely affect a blockchain, and consequently the growth of the altcoin. Pool hopping impacts the network adversely because pool hoppers join the network only when the difficulty is low and they can gain quick rewards; the moment difficulty goes up (or is readjusted) they hop off and then come back again when the difficulty is adjusted back.

For example, if a multipool consumes its resources in quickly mining a new coin, the difficulty will increase very quickly; when the multipool leaves the currency network; it becomes almost unusable because of the fact that now the difficulty has increased to such a level that it is no longer profitable for solo miners and can no longer be maintained. The only fix for this problem is to initiate a hard fork which is usually undesirable for the community.

There are a few algorithms that have come into existence to address this issue and are discussed later in this chapter. All these algorithms are based on the idea of readjusting various parameters in response to hash rate changes; these parameters include the number of previous blocks, the difficulty of previous blocks, the ratio of adjustment, and the number by which the difficulty can be readjusted back or up.

In the following section, readers will be introduced to the few difficulty algorithms being used in and proposed for various altcoins.

Kimoto Gravity Well

This algorithm is used in various altcoins to regulate difficulty. This method was first introduced in Megacoin and used to adjust the difficulty of the network every block adaptively. The logic of the algorithm is shown as follows:

*KGW = 1 + (0.7084 * pow((double(PastBlocksMass)/double(144)), -1.228))*

The algorithm runs in a loop that goes through a set of predetermined blocks (*PastBlockMass*) and calculates a new readjustment value. The core idea behind this algorithm is to develop an adaptive difficulty regulation mechanism that can readjust the difficulty in response to rapid spikes in hash rates. **Kimoto Gravity Well** (**KGW**) ensures that the time between blocks remains approximately the same. In Bitcoin, the difficulty is adjusted every 2016 blocks, but in KGW the difficulty is adjusted at every block.

This algorithm is vulnerable to time warp attacks, which, allow an attacker to enjoy less difficulty in creating new blocks temporarily. This attack allows a time window where the difficulty becomes low, and the attacker can quickly generate many coins at a fast rate.

More information can be found at the link `https://cryptofrenzy.wordpress.com/2014/02/09/multipools-vs-gravity-well/`.

Dark Gravity Wave

Dark Gravity Wave (**DGW**) is a new algorithm designed to address certain flaws such as the time warp attack in the KGW algorithm. This concept was first introduced in Dash, previously known as Darkcoin. It makes use of multiple exponential moving averages and simple move averages to achieve a smoother readjustment mechanism. The formula is shown as follows:

2222222/ (((Difficulty+2600)/9)^2)

This formula is implemented in Dash coin, Bitcoin SegWit2X and various other altcoins as a mechanism to readjust difficulty.

DGW version 3.0 is the latest implementation of DGW algorithm and allows improved difficulty retargeting as compared to KGW.

More information can be found at `https://dashpay.atlassian.net/wiki/spaces/DOC/pages/1146926/Dark+Gravity+Wave`.

DigiShield

This is another difficulty retargeting algorithm that has recently been used in Zcash with slight variations and after adequate experimentation. This algorithm works by going through a fixed number of previous blocks to calculate the time they took to be generated and then readjusts the difficulty to the difficulty of the previous block by dividing the actual time span by averaging the target time. In this scheme, the retargeting is calculated much more rapidly, and also the recovery from a sudden increase or decrease in hash rate is quick. This algorithm protects against multipools, which can result in rapid hash rate increases.

The network difficulty is readjusted every block or every minute depending on the implementation. The key innovation is faster readjusting times as compared to KGW.

Zcash uses DigiShield v3.0 which uses the following formula for difficulty adjustment:

(New difficulty) = (previous difficulty) x SQRT [(150 seconds) / (last solve time)

There is detailed discussion available on it at `https://github.com/zcash/zcash/issues/147#issuecomment-245140908`.

MIDAS

Multi-Interval Difficulty Adjustment System (**MIDAS**) is an algorithm that is comparatively more complex than the algorithms discussed previously due to number of parameters it uses. This method responds much more rapidly to abrupt changes in hash rates. This algorithm also protects against time warp attacks.

The original post about this is now available via web archive at `https://web.archive.org/web/20161005171345/http://dillingers.com/blog/2015/04/21/altcoin-difficulty-adjustment-with-midas/`.

The interested readers can read more about this at the preceding location.

This concludes our introduction to various difficulty adjustment algorithms.

Many alternative cryptocurrencies and protocols have emerged as an attempt to address various limitations in Bitcoin.

Bitcoin limitations

Various limitations in Bitcoin have also sparked some interest in altcoins, which were developed specifically to address limitations in Bitcoin. The most prominent and widely discussed limitation is the lack of anonymity in Bitcoin. We will now discuss some of the limitations of Bitcoin.

Privacy and anonymity

As the blockchain is a public ledger of all transactions and is openly available, it becomes trivial to analyze it. Combined with traffic analyses, transactions can be linked back to their source IP addresses, thus possibly revealing a transaction's originator. This is a big concern from a privacy point of view.

Even though in Bitcoin it is a recommended and common practice to generate a new address for every transaction, thus allowing some level of unlinkability, this is not enough, and various techniques have been developed and successfully used to trace the flow of transactions throughout the network and link them back to their originator. These techniques analyze blockchains by using transaction graphs, address graphs and entity graphs which facilitate linking users back to the transactions, thus raising privacy concerns. The techniques mentioned earlier in the preceding analysis can be further enriched by using publicly available information about transactions and linking them to the actual users. There are open source block parsers available that can be used to extract transaction information, balances, and scripts from the blockchain database.

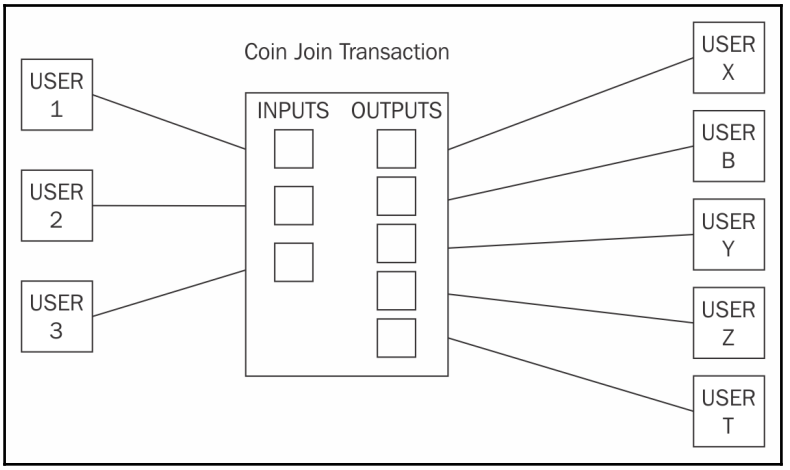
A parser available at `https://github.com/mikispag/rusty-blockparser` is written in Rust language and provides advanced blockchain analysis capabilities.

Various proposals have been made to address the privacy issue in Bitcoin. These proposals fall into three categories: mixing protocols, third-party mixing networks, and inherent anonymity.

A brief discussion of each category is presented as follows.

Mixing protocols

These schemes are used to provide anonymity to bitcoin transactions. In this model, a mixing service provider (an intermediary or a shared wallet) is used. Users send coins to this shared wallet as a deposit, and then, the shared wallet can send some other coins (of the same value deposited by some other users) to the destination. Users can also receive coins that were sent by others via this intermediary. This way the link between outputs and inputs is no longer there and transaction graph analysis will not be able to reveal the actual relationship between senders and receivers.



CoinJoin transaction with three users joining their transaction into a single larger CoinJoin transaction

CoinJoin is one example of mixing protocols, where two transactions are joined together to form a single transaction while keeping the inputs and outputs unchanged. The core idea behind CoinJoin is to build a shared transaction that is signed by all participants. This technique improves privacy for all participants involved in the transactions.

Third-party mixing protocols

Various third-party mixing services are available, but if the service is centralized, then it poses the threat of tracing the mapping between senders and receivers because the mixing service knows about all inputs and outputs. In addition to this, fully centralized miners even pose the risk of the administrators of the service stealing the coins.

Various services, with varying degrees of complexity, such as CoinShuffle, Coinmux, and Darksend in Dash (coin) are available that are based on the idea of CoinJoin (mixing) transactions. CoinShuffle is a decentralized alternative to traditional mixing services as it does not require a trusted third party.

CoinJoin-based schemes, however, have some weaknesses, most prominently the possibility of launching a denial of service attack by users who committed to signing the transactions initially but now are not providing their signature, thus delaying or stopping joint transaction altogether.

Inherent anonymity

This category includes coins that support privacy inherently and is built into the design of the currency. The most popular is Zcash, which uses **Zero-Knowledge Proofs** (**ZKP**) to achieve anonymity. It is discussed in detail later in the chapter. Other examples include Monero, which makes use of ring signatures to provide anonymous services.

The next section introduces various enhancements that have been made or are proposed, to extend the Bitcoin protocol.

Extended protocols on top of Bitcoin

Several protocols, discussed in the following sections, have been proposed and implemented on top of Bitcoin to enhance and extend the Bitcoin protocol and use for various other purposes instead of just as a virtual currency.

Colored coins

Colored coins are a set of methods that have been developed to represent digital assets on the Bitcoin blockchain. Coloring a bitcoin refers colloquially to updating it with some metadata representing a digital asset (smart property). The coin still works and operates as a bitcoin but additionally carries some metadata that represents some assets. This can be some information related to the asset, some calculations related to transactions or any arbitrary data. This mechanism allows issuing and tracking specific bitcoins. Metadata can be recorded using the bitcoins `OP_RETURN` opcode or optionally in multisignature addresses. This metadata can also be encrypted if required to address any privacy concerns. Some implementations also support storage of metadata on publicly available torrent networks which means that virtually unlimited amounts of metadata can be stored. Usually these are JSON objects representing various attributes of the colored coin. Moreover, smart contracts are also supported. One example of such implementation is Colu, which can be found at, `http://colu.co/`.

Colored coins can be used to represent a multitude of assets including, but not limited to commodities, certificates, shares, bonds, and voting. It should also be noted that to work with colored coins, a wallet that interprets colored coins is necessary and normal Bitcoin wallets will not work. Normal Bitcoin wallets will not work, because they cannot differentiate between *colored coins* and *not colored coins*.

Colored coin wallets can be set up online using a service available at `https://www.coinprism.com/`. By using this service, any digital asset can be created and issued via a colored coin.

The idea of colored coins is very appealing as it does not require any modification to the existing Bitcoin protocol and can make use of the already existing secure Bitcoin network. In addition to the traditional representation of digital assets, there is also the possibility of creating smart assets that behave according to the parameters and conditions defined for them. These parameters include time validation, restrictions on transferability, and fees. This opens the possibility of creating smart contracts which we will discussed in chapter 9, *Smart Contracts*.

A significant use case can be the issuance of financial instruments on the blockchain. This will ensure low transaction fees, valid and mathematically secure proof of ownership, fast transferability without requiring some intermediary, and instant dividend payouts to the investors.

A rich API is available for colored coins at `http://coloredcoins.org/`.

Counterparty

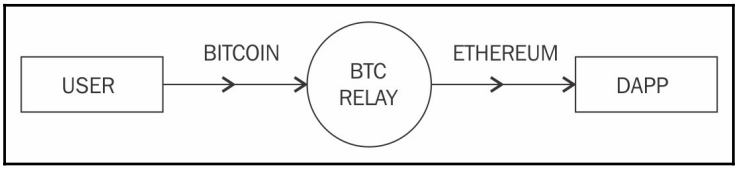
This is another service that can be used to create custom tokens that act as a cryptocurrency and can be used for various purposes such as issuing digital assets on top of bitcoin blockchain. This is quite a robust platform and runs on bitcoin blockchains at their core but has developed its client and other components to support issuing digital assets. The architecture consists of a counterparty server, counter block, counter wallet, and `armory_utxsvr`. Counterparty works based on the same idea as colored coins by embedding data into regular bitcoin transactions but provides a much more productive library and set of powerful tools to support the handling of digital assets. This embedding is also called **embedded consensus** because the counterparty transactions are embedded within bitcoin transactions. The method of embedding the data is by using `OP_RETURN` opcode in bitcoin.

The currency produced and used by Counterparty is known as XCP and is used by smart contracts as the fee for running the contract. At the time of writing its price is 2.78 USD. XCPs were created by using the PoB method discussed previously.

Counterparty allows the development of smart contracts on Ethereum using solidity language and allows interaction with bitcoin blockchain. To achieve this, BTC Relay is used as a means to provide interoperability between Ethereum and Bitcoin. This is a clever concept where Ethereum contracts can talk to bitcoin blockchain and transactions through BTC Relay. The relayers (nodes that are running BTC Relay) fetch the bitcoin block headers and relay them to a smart contract on the Ethereum network that verifies the PoW. This process verifies that a transaction has occurred on the bitcoin network.

This is available at `http://btcrelay.org/`.

Technically, this is an Ethereum contract that is capable of storing and verifying bitcoin block headers just like bitcoin simple payment verification lightweight clients do by using bloom filters. SPV clients were discussed in detail in the previous chapter. The idea can be visualized with the following diagram:



BTC relay concept

Counterparty is available at `http://counterparty.io/`.

Development of altcoins

Altcoin projects can be started very quickly from a coding point of view by simply forking the bitcoin or another coin's source code, but this probably is not enough. When a new coin project is started, several things need to be considered to ensure a successful launch and the coin's longevity. Usually, the code base is written in C++ as was the case with bitcoin, but almost any language can be used to develop coin projects, for example, Golang or Rust.

Writing code or forking the code for an existing coin is the trivial part, the challenging issue is how to start a new currency so that investors and users can be attracted to it. Generally, the following steps are taken in order to start a new coin project.

From a technical point of view, in the case of forking the code of another coin, for example, bitcoin, there are various parameters that can be changed to effectively create a new coin. These parameters are required to be tweaked or introduced in order to create a new coin. These parameters can include but are not limited to the following.

Consensus algorithms

There is a choice of consensus algorithms available, for example PoW used in Bitcoin or PoS, used in Peercoin. There are also other algorithms available such as **Proof of Capacity** (**PoC**) and few others, but PoW and PoS are the most common choices.

Hashing algorithms

This is either SHA-256, Scrypt, X11, X13, X15, or any other hashing algorithm that is adequate for use as a consensus algorithm.

Difficulty adjustment algorithms

Various options are available in this category to provide difficulty retargeting mechanisms. The most prominent examples are KGW, DGW, Nite's Gravity Wave, and DigiShield. Also, all these algorithms can be tweaked based on requirements to produce different results; therefore, many variants are possible.

Inter-block time

This is the time elapsed between the generation of each block. For bitcoin the blocks are generated every 10 minutes, for litecoin it's 2.5 minutes. Any value can be used but an appropriate value is usually between a few minutes; if the generation time is too fast it might destabilize the blockchain, if it's too slow it may not attract many users.

Block rewards

A block reward is for the miner who solves the mining puzzle and is allowed to have a coinbase transaction that contains the reward. This used to be 50 coins in bitcoin initially and now many altcoins set this parameter to a very high number; for example, in Dogecoin it is 10,000, currently.

Reward halving rate

This is another important factor; in bitcoin, it is halved every 4 years and now is set to 12.5 bitcoins. It's a variable number that can be set to any time period or none at all depending on the requirements.

Block size and transaction size

This is another important factor that determines how high or low the transaction rate can be on the network. Block sizes in bitcoin are limited to 1 MB but in altcoins, it can vary depending on the requirements.

Interest rate

This property applies only to PoS systems where the owner of the coins can earn interest at a rate defined by the network in return for some coins that are held on the network as a stake to protect the network. This interest rate keeps inflation under control. If interest rate is too low then it can cause hyperinflation.

Coinage

This parameter defines how long the coin has to remain unspent in order for it to become eligible to be considered stake worthy.

Total supply of coins

This number sets the total limit of the coins that can ever be generated. For example, in Bitcoin the limit is 21 million, whereas in Dogecoin it's unlimited. This limit is fixed by the block reward and halving schedule discussed earlier.

There are two options to create your own virtual currency: forking existing established cryptocurrency source code or writing a new one from scratch. The latter option is less popular but the first option is easier and has allowed the creation of many virtual currencies over the last few years. Fundamentally, the idea is that first a cryptocurrency source code is forked and then appropriate changes are made at different strategic locations in the source code to effectively create a new currency. NEM coin is one of the newly created coins that have their code written entirely from scratch.

In the next section, readers are introduced to some altcoin projects. It is not possible to cover all alternative currencies in this chapter, but a few selected coins are discussed in the following section. Selection is based on longevity, market cap, and innovation. Each coin is discussed from different angles such as theoretical foundations, trading, and mining.

Namecoin

Namecoin is the first fork of the Bitcoin source code. The key idea behind Namecoin is not to produce an altcoin but instead to provide improved decentralization, censorship resistance, privacy, security, and faster-decentralized naming. Decentralized naming services are intended to respond to inherent limitations such as slowness and centralized control in the traditional **Domain Name System** (**DNS**) protocols used on the internet. Namecoin is also the first solution to Zooko's triangle, which was briefly discussed in `Chapter 1`, *Blockchain 101.*

Namecoin is used to essentially provide a service to register a key/value pair. One major use case of Namecoin is that it can provide a decentralized **Transport Layer Security** (**TLS**) certificate validation mechanism, driven by blockchain-based distributed and decentralized consensus.

It is based on the same technology introduced with bitcoin, but with its own blockchain and wallet software.

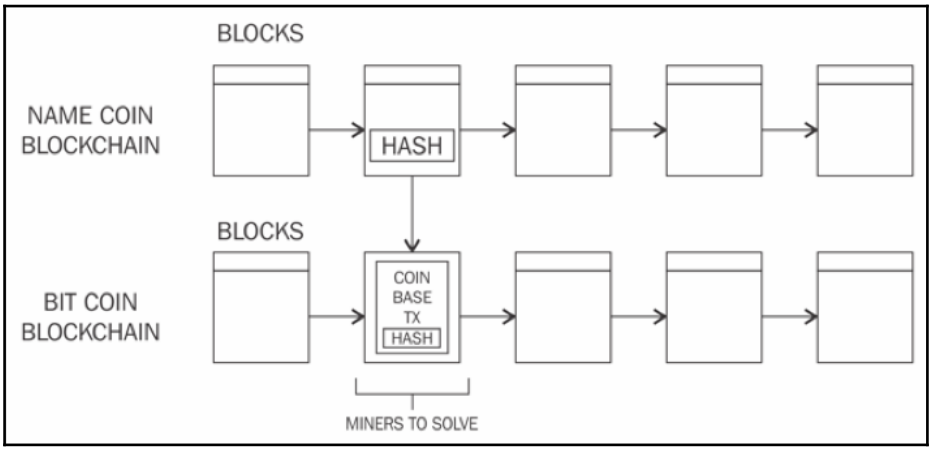
The source code for the Namecoin core is available at `https://github.com/namecoin/namecoin-core`.

In summary, Namecoin provides the following three services:

- Secure storage and transfer of names (keys)
- Attachment of some value to the names by attaching up to 520 bytes of data
- Production of a digital currency (Namecoin)

Namecoin also for the first time introduced merged mining, which allows a miner to mine on more than one chain simultaneously. The idea is simple but very effective: miners create a Namecoin block and produce a hash of that block. Then the hash is added to a Bitcoin block and miners solve that block at equal to or greater than the Namecoin block difficulty to prove that enough work has been contributed towards solving the Namecoin block.

The coinbase transaction is used to include the hash of the transactions from Namecoin (or any other altcoin if merged mining with that coin). The mining task is to solve Bitcoin blocks whose coinbase `scriptSig` contains a hash pointer to Namecoin (or any other altcoin) block. This is shown in the following diagram:



Merged mining visualization

If a miner manages to solve a hash at the bitcoin blockchain difficulty level, the bitcoin block is built and becomes part of the Bitcoin network. In this case, the Namecoin hash is ignored by the bitcoin blockchain. On the other hand, if a miner solves a block at Namecoin blockchain difficulty level a new block is created in the Namecoin blockchain. The core benefit of this scheme is that all the computational power spent by the miners contributes towards securing both Namecoin and Bitcoin.

Trading Namecoins

The current market cap of Namecoin is $29,143,884 USD as per `https://coinmarketcap.com/` in March, 2018. It can be bought and sold at various exchanges such as:

- `https://cryptonit.net/`
- `https://bisq.network`
- `https://www.evonax.com`
- `https://bter.com`

Obtaining Namecoins

Even though Namecoins can be mined independently, they are usually mined as part of bitcoin mining by utilizing the merged mining technique as explained earlier. This way Namecoin can be mined as a by-product of bitcoin mining. Solo mining is no longer profitable as is evident from the following difficulty graph; instead, it is recommended to use merged mining, use a mining pool, or even use a cryptocurrency exchange to buy Namecoin.

Namecoin difficulty as shown at: https://bitinfocharts.com/comparison/difficulty-nmc.html (since December, 2016)

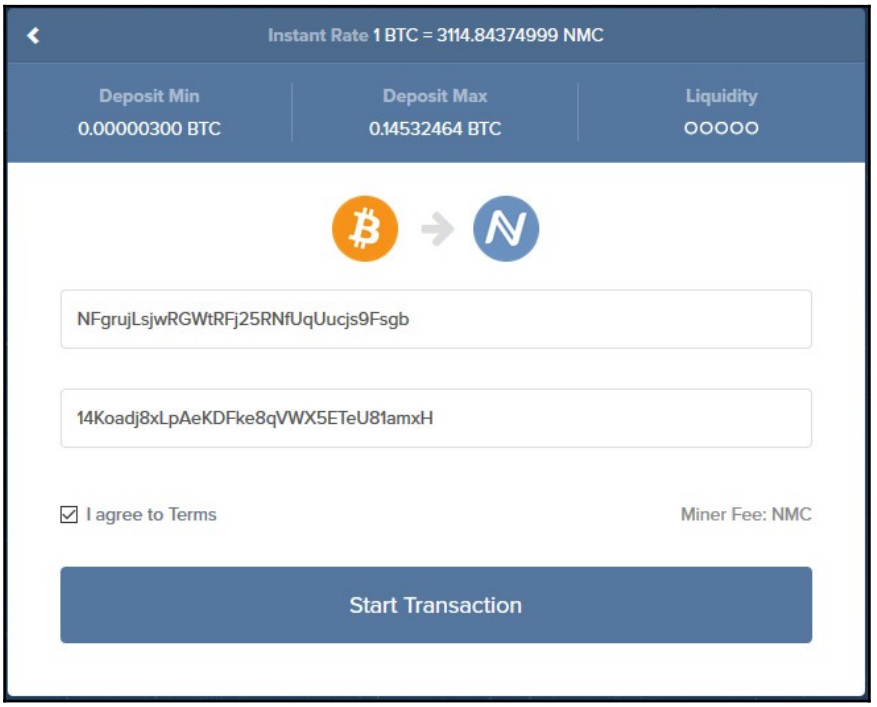
Various mining pools such as `https://slushpool.com` also offer the option of merged mining. This allows a miner to mine primarily bitcoin but also, as a result, earn Namecoin too.

Another method that can be used to quickly get some Namecoins is to swap your existing coins with Namecoins, for example, if you already have some bitcoins or another cryptocurrency that can be used to exchange with Namecoin.

An online service, `https://shapeshift.io/`, is available that provides this service. This service allows conversion from one cryptocurrency to another, using a simple user-friendly interface.

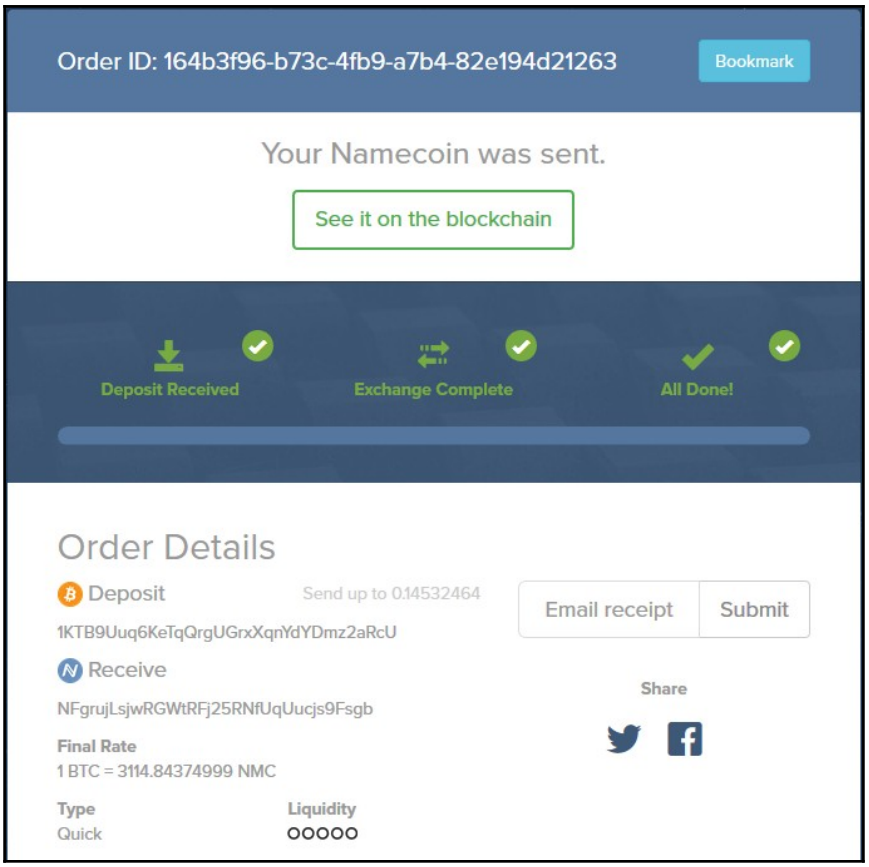
For example, paying BTC to receive NMC is shown as follows:

1. First the deposit coin is selected, which in this case is bitcoin and coin to be received is selected, which is Namecoin in this case. In the top editbox, the Namecoin address where you want to receive the exchanged Namecoin is entered. In the second editbox, at the bottom the bitcoin refund address is entered, where the coins will be returned to in case the transaction fails for any reason.
2. The exchange rate and miner fee are calculated instantly as soon as the deposit and exchange currency are chosen. Exchange rate is driven by the market conditions whereas miner fee is calculated algorithmically based on the target currency chosen and what the target network's miner would charge.



Bitcoin to Namecoin exchange

3. Once **Start Transaction** is clicked, the transaction starts and instructs the user to send the bitcoins to a specific bitcoin address. When the user sends the required amount, the conversion process starts as shown in the following screenshot. This whole process takes few minutes:

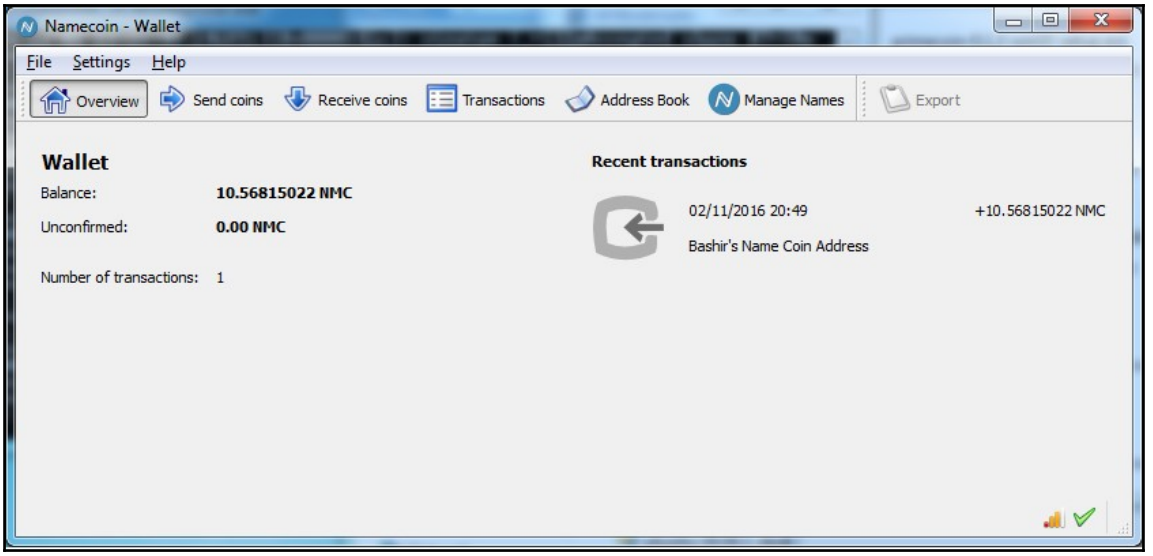


Notification of Namecoin delivery

The preceding screenshot shows that after sending the deposit the exchange occurs and finally **All Done!** message is displayed indicating that the exchange has been successful.

A few other order details are displayed on the page such as what currency was deposited and what was received after exchange. In this case it is Bitcoin to Namecoin exchange. It's also worth noting that relevant addresses are also displayed under each coin icon. There are few other option such **Email receipt** which can be invoked to receive an email receipt of the transaction.

When the process completes, the transactions can be viewed in the Namecoin wallet as shown here:



Namecoin wallet

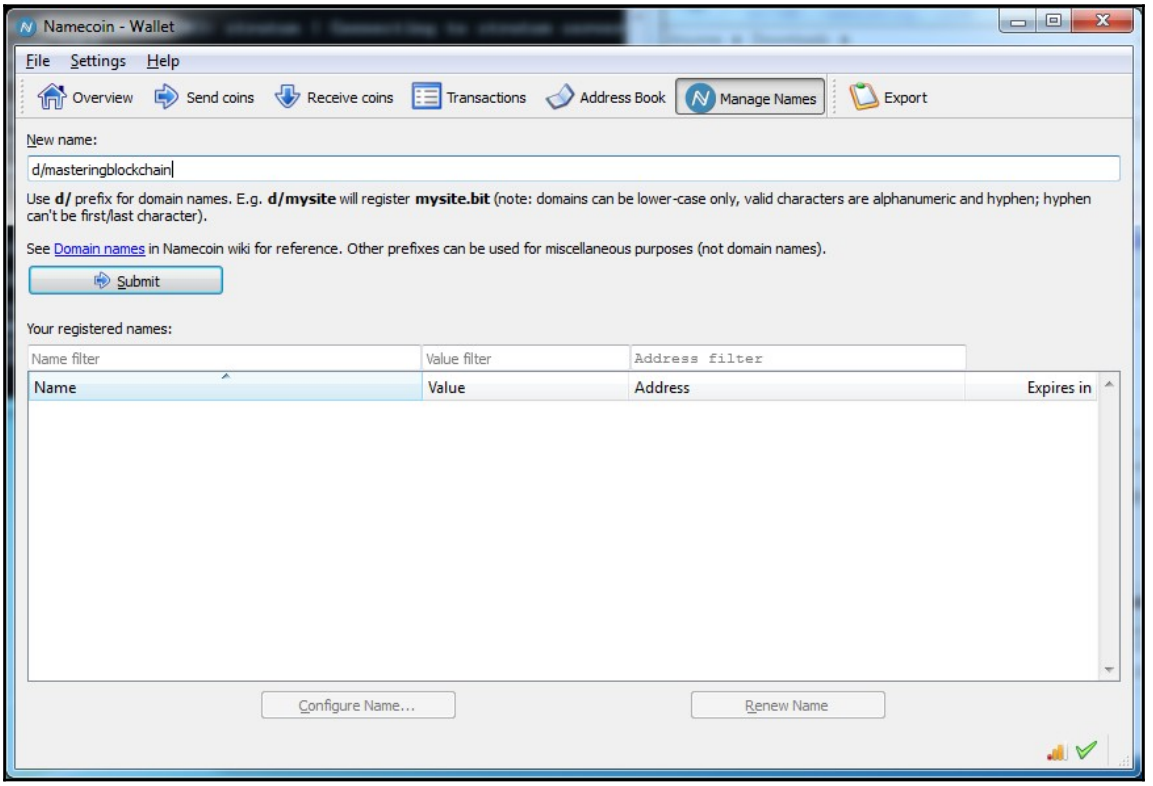
It may take some time (usually around 1 hour) to confirm the transactions; until that time, it is not possible to use the Namecoins to manage names. Once Namecoins are available in the wallet, the **Manage Names** option can be used to generate Namecoin records.

Generating Namecoin records

Namecoin records are in the form of key and value pairs. A name is a lowercase string of the form `d/examplename` whereas a value is a case-sensitive, UTF-8 encoded JSON object with a maximum of 520 bytes. The name should be RFC1035 (`https://tools.ietf.org/html/rfc1035`)-compliant.

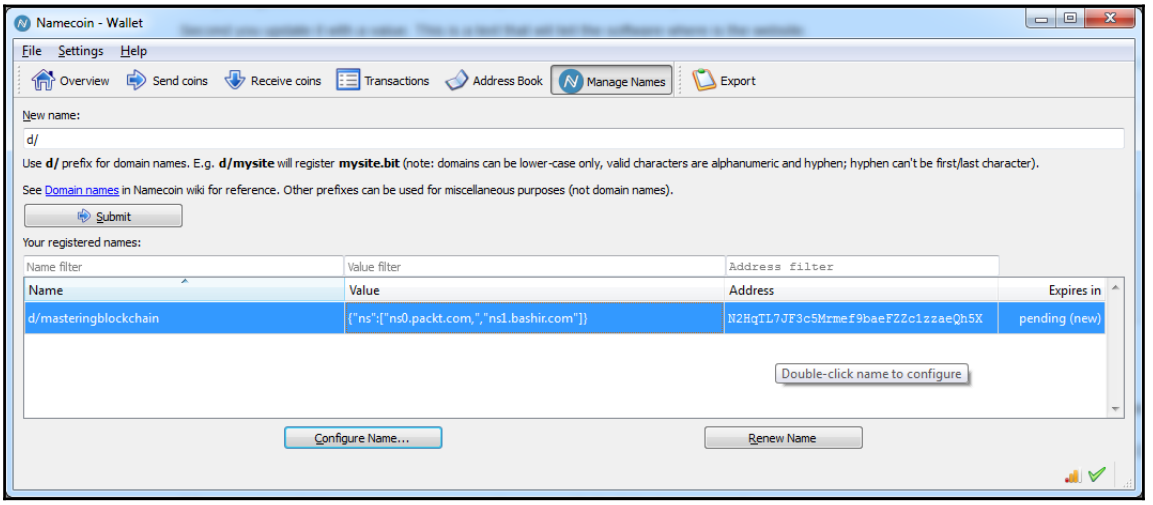
A general Namecoin name can be an arbitrary binary string up to 255 bytes long with, 1024-bits of associated identifying information. A record on a Namecoin chain is only valid for around 200 days or 36,000 blocks after which it needs to be renewed. Namecoin also introduced `.bit` top-level domains that can be registered using Namecoin and can be browsed using specialized Namecoin-enabled resolvers. Namecoin wallet software as shown in the following screenshot can be used to register `.bit` domain names.

The name is entered and, after the **Submit** button is pressed, it will ask for configuration information such as DNS, IP, or identity:



Namecoin wallet: domain name configuration

As shown in the following screenshot, `masteringblockchain` will register as `masteringblockchain.bit` on the Namecoin blockchain:



Namecoin wallet: showing registered name

Litecoin

Litecoin is a fork of the bitcoin source code released in 2011. It uses Scrypt as PoW, originally introduced in the Tenebrix coin. Litecoin allows for faster transactions as compared to bitcoin due to its faster block generation time of 2.5 minutes. Also, difficulty readjustment is achieved every 3.5 days roughly due to faster block generation time. The total coin supply is 84 million.

Scrypt is a sequentially memory hard function that is the first alternative to the SHA-256-based PoW algorithm. It was originally proposed as a **Password-Based Key Derivation Function** (**PBKDF**). The key idea is that if the function requires a significant amount of memory to run then custom hardware such as ASICs will require more VLSI area, which would be infeasible to build. The Scrypt algorithm requires a large array of pseudorandom bits to be held in memory and a key is derived from this in a pseudorandom fashion.

The algorithm is based on a phenomenon called **Time-Memory Trade-Off** (**TMTO**). If memory requirements are relaxed, then it results in increased computational cost. Put another way, TMTO shortens the running time of a program if more memory is given to it. This trade-off makes it unfeasible for an attacker to gain more memory because it is expensive and difficult to implement on custom hardware, or if the attacker chooses not to increase memory, then it results in the algorithm running slowly due to high processing requirements. This means that ASICs are difficult to build for this algorithm.

Scrypt uses the following parameters to generate a derived key (*Kd*):

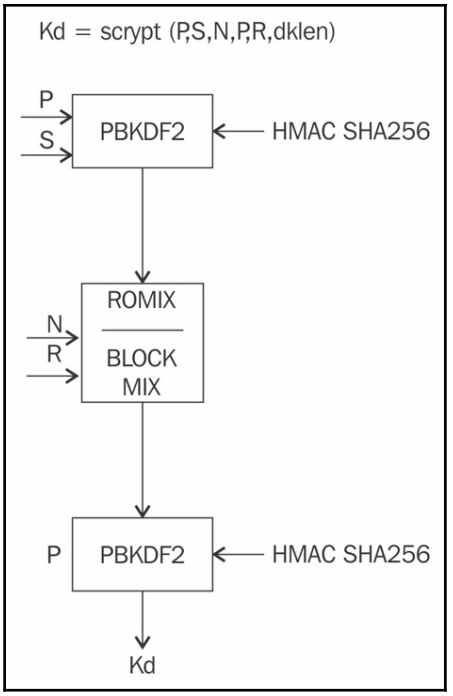
- **Passphrase**: This is a string of characters to hash
- **Salt**: This is a random string that is provided to Scrypt functions (generally all hash functions) in order to provide a defense against brute-force dictionary attacks using rainbow tables
- **N**: This is a memory/CPU cost parameter that must be a power of 2 > 1
- **P**: This is the parallelization parameter
- **R**: This is the block size parameter
- **dkLen**: This is the intended length of the derived key in bytes

Formally, this function can be written as follows:

$$Kd = scrypt\ (P, S, N, P, R, dkLen)$$

Before applying the core Scrypt function, the algorithm takes *P* and *S* as input and applies PBKDF2 and SHA-256-based HMAC. Then the output is fed to an algorithm called ROMix, which internally uses the Blockmix algorithm using the Salsa20/8 core stream cipher to fill up the memory which requires large memory to operate, thus enforcing the sequentially memory hard property.

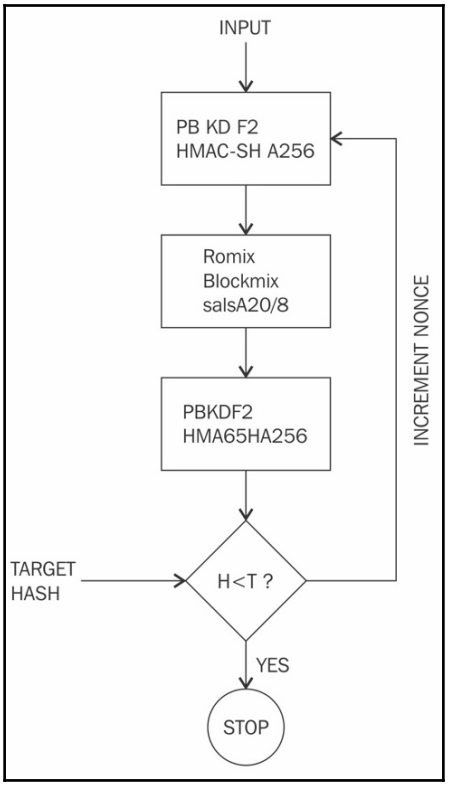
The output from this step of the algorithm is finally fed to the PBKDF2 function again in order to produce a derived key. This process is shown in the following diagram:



Scrypt algorithm

Scrypt is used in Litecoin mining with specific parameters where *N= 1024*, *R = 1*, *P=1*, and *S = random 80 bytes* producing a 256-bit output.

It appears that, due to the selection of these parameters, the development of ASICs for Scrypt for Litecoin mining turned out to be not very difficult. In an ASIC for Litecoin mining, a sequential logic can be developed that takes the data and nonce as input and applies the PBKDF2 algorithm with HMAC-SHA256; then the resultant bit stream is fed into the SALSA20/8 function which produces a hash that again is fed down to the PBKDF2 and HMAC-256 functions to produce a 256-bit hash output. As is the case with bitcoin PoW, in Scrypt also if the output hash is less than the target hash (already passed as input at the start, stored in memory, and checked with every iteration) then the function terminates; otherwise, the nonce is incremented and the process is repeated again until a hash is found that is lower than the difficulty target.



Scrypt ASIC design simplified flowchart

- **Trading litecoin**: As with other coins, trading litecoin is easily carried out on various online exchanges. The current market cap of litecoin is $10,448,974,615. The current price (as of March, 2018) of litecoin is $188.04/LTC.
- **Mining**: Litecoin mining can be carried out solo or in pools. At the moment, ASICs for Scrypt are available that are commonly used to mine Litecoin.

Litecoin mining on a CPU is no longer profitable as is the case with many other digital currencies now. There are online cloud mining providers and ASIC miners available that can be used to mine Litecoin. Litecoin mining started from the CPU, progressed through GPU mining rigs, and eventually now has reached a point where specialized ASIC miners, such as ASIC Scrypt Miner Wolf are available from Ehsminer are now required to be used in the hope of being able to make some coins. Generally, it is true that even with ASICs it is better to mine in pools instead of solo as solo mining is not as profitable as mining in pools due to the proportional rewards scheme used by mining pools. These miners are capable of producing a hashing rate of 2 Gh/s for Scrypt algorithms.

- **Software source code and wallet**: The source code for litecoin is available at `https://github.com/litecoin-project/litecoin`. The Litecoin wallet can be downloaded from `https://litecoin.org/` and can be used just like the Bitcoin core client software.

Primecoin

Primecoin is the first digital currency on the market that introduced a useful PoW, as opposed to Bitcoin's SHA256-based PoW. Primecoin uses searching prime numbers as a PoW. Not all types of prime number meet the requirements to be selected as PoW. Three types of prime numbers (known as Cunningham chain of the first kind, Cunningham chain of the second kind, and bi-twin chains) meet the requirements of a PoW algorithm to be used in cryptocurrencies.

The difficulty is dynamically adjusted via a continuous difficulty evaluation scheme in Primecoin blockchain. The efficient verification of PoW based on prime numbers is also of high importance, because if verification is slow, then PoW is not suitable. Therefore, prime chains are selected as a PoW because finding prime chains gets difficult as the chain increases in length whereas verification remains quick enough to warrant being used as an efficient PoW algorithm.

It is also important that once a PoW has been verified on a block, it must not be reusable on another block. This is accomplished in Primecoin by a combination of PoW certificates and hashing it with the header of the parent block in the child block.

The PoW certificate is produced by linking the prime chain to the block header hash. It also requires that the block header's origin be divisible by the block header hash. If it is, it is divided and after division, the quotient is used as a PoW certificate. Another property of the adjustable difficulty of PoW algorithms is met by introducing difficulty adjustment every block instead of every 2,016, as is the case with bitcoin. This is a smoother approach as compared to bitcoin and allows readjustment in the case of sudden increases in hash power. Also, the total number of coins generated is community-driven, and there is no definite limit on the number of coins Primecoin can generate.

Trading Primecoin

Primecoins can be traded on major virtual currency trading exchanges. The current market cap of Primecoin is $17,482,507 at the time of writing (March, 2018). It is not very large but, because Primecoin is based on a novel idea and there is a dedicated community behind it, this continues to hold some market share.

A graph, showing some statistics related to Primecoin

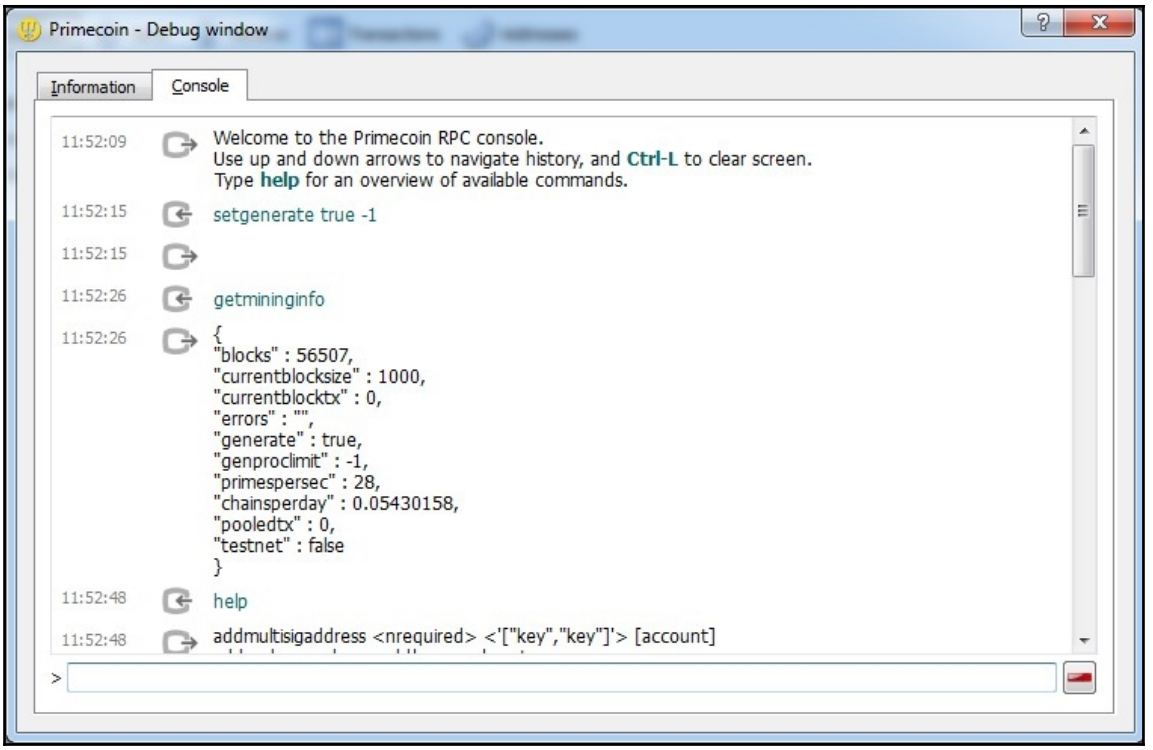
Data sourced from: https://coinmarketcap.com/currencies/primecoin/

Mining guide

The first step is to download a wallet. Primecoin supports native mining within the wallet, just like original Bitcoin clients, but also can be mined on the cloud via various online cloud service providers.

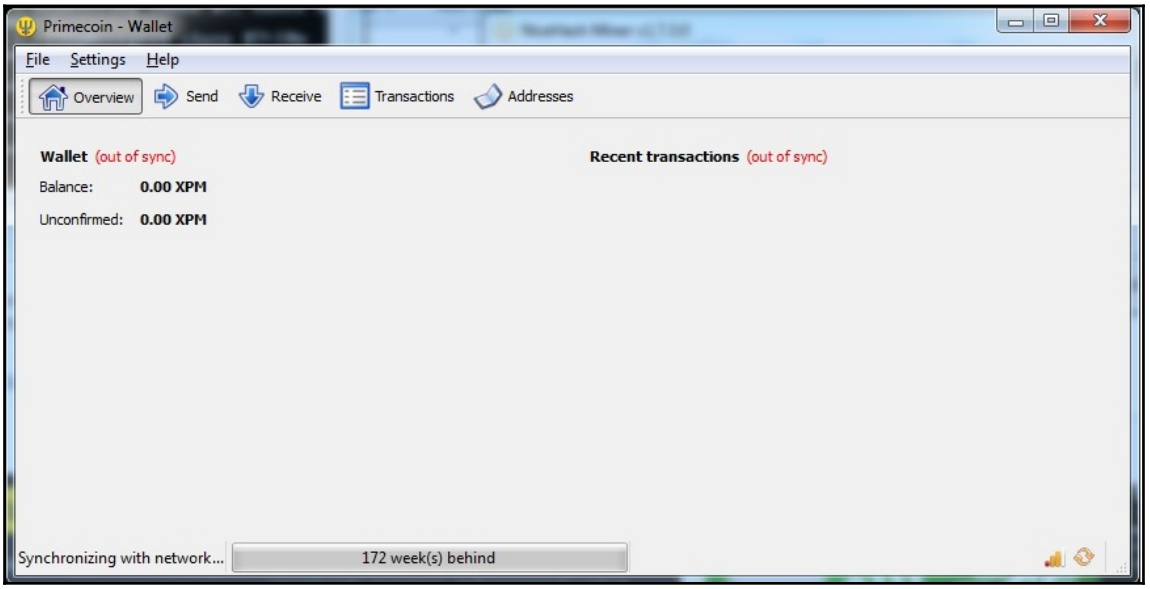
A quick Windows guide is presented as follows, Linux client is also available at `http://primecoin.io/downloads.php`.

1. The first step is to download the Primecoin wallet from `http://primecoin.io/index.php`.
2. Once the wallet is installed and synced with the network, mining can be started by following the next step. A debug window can be opened in the Primecoin wallet by clicking on the **Help menu** and selecting the **Debug window** menu item. Additional help can be invoked through typing `help` in the **Console** window of **Debug window** used to enable the Primecoin mining function:



Primecoin mining

3. Once the preceding commands are successfully executed mining will start in solo mode. This may not be very fast and profitable if you have an entry level PC with slower CPU but as this is a CPU mined cryptocurrency the miner can use PCs with powerful CPUs. As an alternative cloud services can be used which host powerful server hardware:



Primecoin wallet software, synching with the network

The Primecoin source code is available at `https://github.com/primecoin/primecoin`.

Primecoin is a novel concept and the PoW that it has introduced have great scientific significance. It is still in use with a market cap of $17,034,198 USD but it appears that no active development is being carried out to further develop Primecoin as is evident from GitHub inactivity.

Readers can further explore Primecoin by reading the Primecoin whitepaper by Sunny King (pseudonym) at: `http://primecoin.io/bin/primecoin-paper.pdf`.

Zcash

Zcash was launched on October 28, 2016. This is the first currency that uses a specific type of ZKPs known as **Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge** (**ZK-SNARKs**) to provide complete privacy to the user. These proofs are concise and easy to verify; however, setting up the initial public parameters is a complicated process. The latter include two keys: the proving key and verifying key. The process requires sampling some random numbers to construct the public parameters. The issue is that these random numbers, also called toxic waste, must be destroyed after the parameter generation in order to prevent counterfeiting of Zcash.

For this purpose, the Zcash team came up with a multi-party computation protocol to generate the required public parameters collaboratively from independent locations to ensure that toxic waste is not created. Because these public parameters are required to be created by the Zcash team, it means that the participants in the ceremony are trusted. This is the reason why the ceremony was very open and conducted by making use of a multi-party computation mechanism.

This mechanism has a property whereby all of the participants in the ceremony will have to be compromised to compromise the final parameters. When the ceremony is completed all participants physically destroyed the equipment used for private key generation. This action eliminates any trace of the participants' part of the private key on the equipment.

ZK-SNARKs must satisfy the properties for completeness, soundness, succinctness, and non- interactivity. Completeness means that there is a definite strategy for a prover to satisfy a verifier that an assertion is true. On the other hand, soundness means that no prover can convince the verifier that a false statement is true. Succinctness means that messages passed between the prover and verifier are tiny in size.

Finally, the property non-interactive means that the verification of correctness of an assertion can be carried out without any interaction or very little interaction. Also, being a ZKP, the property of zero-knowledge (discussed in `Chapter 4`, *Public Key Cryptography)* needs to be met too.

Zcash developers have introduced the concept of a **Decentralized Anonymous Payment scheme** (**DAP scheme**) that is used in the Zcash network to enable direct and private payments. The transactions reveal no information about the origin, destination, and amount of the payments. There are two types of addresses available in Zcash, Z address and T address. Z addresses are based on ZKPs and provide privacy protection whereas T addresses are similar to those of bitcoin. A snapshot of various attributes of Zcash (after an initial slow start) is shown as follows:

Attribute	Value
Name	Zcash
Launch date	28/10/16
Main purpose	Currency
Currency Code	ZEC
Maximum coins	21 million
Block time	10 minutes
Consensus facilitation algorithm	Proof of Work (equihash)
Difficulty adjustment algorithm	DigiShield V3 (modified)
Mining hardware	CPU, GPU
Difficulty adjustment period	1 block

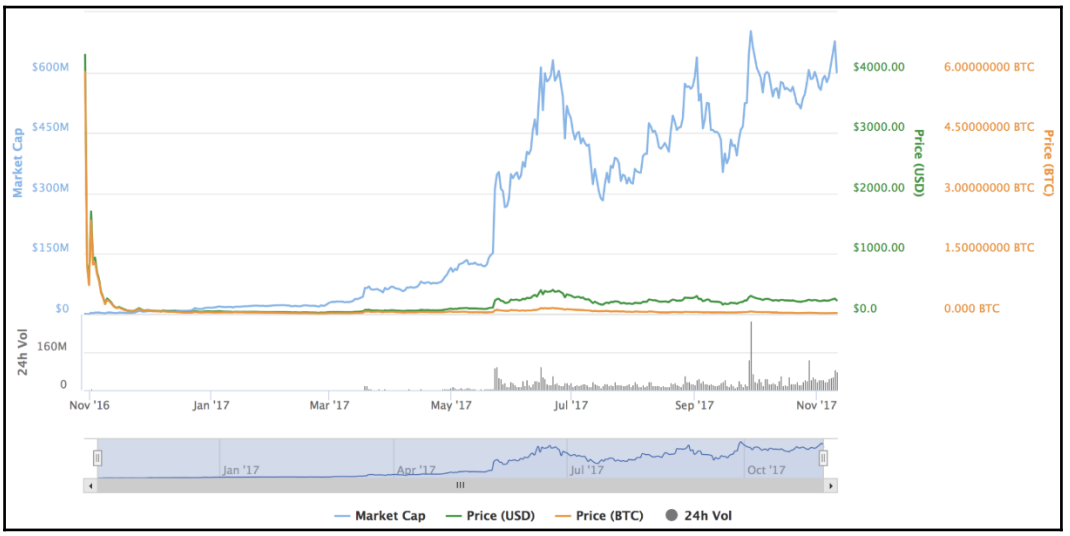
Zcash attributes summary

Zcash uses an efficient PoW scheme named asymmetric PoW (Equihash), which is based on the *Generalized Birthday Problem*. It allows very efficient verification. It is a memory-hard and ASIC-resistant function. A novel idea (initial slow mining) has been introduced with Zcash, which means that the block reward increases gradually over a period until it reaches the 20,000th block. This allows for initial scaling of the network and experimentation by early miners, and adjustment by Zcash developers if required. The slow start did have an impact on price due to scarcity as the price of ZEC on its first day of launch reached roughly 25,000 USD. A slightly modified version of the DigiShield difficulty adjustment algorithm has been implemented in Zcash. The formula is shown as follows:

(Next difficulty) = (last difficulty) x SQRT [(150 seconds) / (last solve time)]

Trading Zcash

Zcash can be bought on major digital currency sellers and exchanges such as CryptoGo (`https://cryptogo.com`). Another exchange where Zcash can be bought or sold is Crypto Robot 365 (`https://cryptorobot365.com`). When Zcash was introduced its price was very high. As shown in the following graph, the price soared as high as approximately ten bitcoins per Zcash. Some exchanges carried out orders as high as 2,500 BTC per ZEC. Price of ZEC is around 311 USD at the time writing (March, 2018):



Zcash market cap and price

Mining guide

There are multiple methods to mine Zcash. Currently, CPU and GPU mining are possible. Various commercial cloud mining pools also offer contracts for mining Zcash. To perform solo mining using a CPU, the following steps can be followed on Ubuntu Linux:

1. The first step is to install prerequisites using the following command:

```
$ sudo apt-get install \
  build-essential pkg-config libc6-dev m4 g++-multilib \
  autoconf libtool ncurses-dev unzip git python \
  zlib1g-dev wget bsdmainutils automake
```

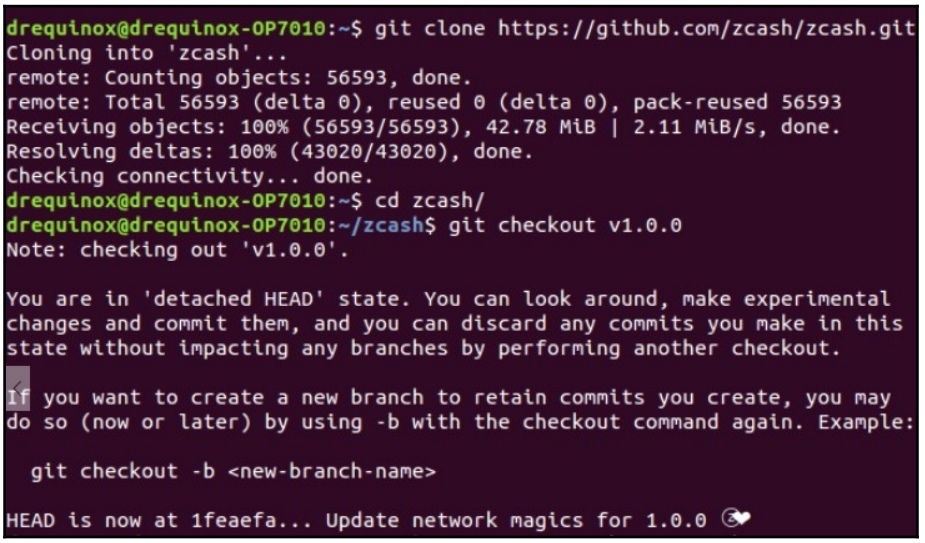
If the prerequisites are already installed, a message will display indicating that components are already the newest version. If not already installed or older than the latest package, then the installation will continue, the required packages will be downloaded, and the installation will be completed.

2. Next, run the commands to clone Zcash from Git as shown in the following screenshot:

```
$ git clone https://github.com/zcash/zcash.git
```

Note that if you are running `git` for the first time then you may have to accept a few configuration changes, which will automatically be done for you, but you may have to do this interactively.

This command will clone the Zcash Git repository locally. The output is shown in the following screenshot:



Cloning the Zcash Git repository

3. The next step is to download proving and verifying keys, by using the following commands:

```
$ ./zcutil/fetch-param.sh
```

This command will produce the output similar to the one shown here:

```
drequinox@drequinox-OP7010:~/zcash$ ./zcutil/fetch-params.sh
Zcash - fetch-params.sh

This script will fetch the Zcash zkSNARK parameters and verify their
integrity with sha256sum.

The parameters are currently just under 911MB in size, so plan accordingly
for your bandwidth constraints. If the files are already present and
have the correct sha256sum, no networking is used.

Creating params directory. For details about this directory, see:
/home/drequinox/.zcash-params/README

Retrieving: https://z.cash/downloads/sprout-proving.key
--2016-10-28 21:46:21--  https://z.cash/downloads/sprout-proving.key
Resolving z.cash (z.cash)... 104.236.171.172
Connecting to z.cash (z.cash)|104.236.171.172|:443... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: https://s3.amazonaws.com/zcashfinalmpc/sprout-proving.key [following]
--2016-10-28 21:46:22--  https://s3.amazonaws.com/zcashfinalmpc/sprout-proving.key
Resolving s3.amazonaws.com (s3.amazonaws.com)... 54.231.40.114
Connecting to s3.amazonaws.com (s3.amazonaws.com)|54.231.40.114|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 910173851 (868M) [application/octet-stream]
Saving to: '/home/drequinox/.zcash-params/sprout-proving.key.dl'

     0K ........ ........ ........ ........  3% 2.71M 5m8s
 32768K ........ ........ ........ ........  7% 3.58M 4m20s
 65536K ........ ........ ........ ........ 11% 2.53M 4m28s
 98304K ........ ........ ........ ........ 14% 1.75M 4m59s
131072K ........ ........ ........
```

Zcash setup fetching ZK-SNARK parameters

4. Once this command runs, it will download around 911 MBs of keys into the `~/.zcash-params/` directory. The directory contains files for proving and verifying keys:

```
$ pwd
/home/drequinox/.zcash-params
$ ls -ltr
sprout-verifying.key
sprout-proving.key
```

5. Once the preceding commands are completed successfully, the source code can be built using the following command:

```
$ ./zcutil/build.sh -j$(nproc)
```

   This will produce a very long output; if everything goes well it will produce a `zcashd` binary file. Note that this command takes `nproc` as the parameter, which is basically a command that finds the number of cores or processors in the system and displays that number. If you don't have that command then replace `nproc` with the number of processors in your system.

6. Once the build is completed, the next step is to configure Zcash. This is achieved by creating a configuration file with the name `zcash.conf` in the `~/.zcash/` directory.

   A sample configuration file is shown as follows:

```
addnode=mainnet.z.cash
rpcuser=drequinox
rpcpassword=xxxxxxoJNo4o5c+F6E+J4P2C1D5izlzIKPZJhTzdW5A=
gen=1
genproclimit=8
equihashsolver=tromp
```

   The preceding configuration enables various features. The first line adds the mainnet node and enables mainnet connectivity. `rpcuser` and `rpcpassword` are the username and password for the RPC interface. `gen = 1` is used to enable mining. `genproclimit` is the number of processors that can be used for mining. The last line enables a faster mining solver; this is not required if you want to use standard CPU mining.

7. Now Zcash can be started using the following command:

```
$ ./zcashd --daemon
```

Once started this will allow interaction with the RPC interface via the `zcash-cli` command-line interface. This is almost the same as the bitcoin command-line interface. Once the Zcash daemon is up-and-running, various commands can be run to query different attributes of Zcash. Transactions can be viewed locally by using the CLI or via a blockchain explorer.

A blockchain explorer for Zcash is available at:
`https://explorer.zcha.in/`.

Address generation

New Z addresses can be generated using the following command:

```
$ ./zcash-cli z_getnewaddress
zcPDBKuuwHJ4gqT5Q59zAMXDHhFoihyTC1aLE5Kz4GwgUXfCBWG6SDr45SFLUsZhpcdvHt7nFmC
3iQcn37rKBcVRa93DYrA
```

Running the `zcash-cli` command with the `getinfo` parameter produces the output shown in the following screenshot. It displays valuable information such as `blocks`, `difficulty`, and `balance`:

```
drequinox@drequinox-OP7010:~/zcash/src$ ./zcash-cli getinfo
{
    "version" : 1000050,
    "protocolversion" : 170002,
    "walletversion" : 60000,
    "balance" : 0.00000000,
    "blocks" : 601,
    "timeoffset" : 0,
    "connections" : 8,
    "proxy" : "",
    "difficulty" : 13748.56014152,
    "testnet" : false,
    "keypoololdest" : 1477688856,
    "keypoolsize" : 101,
    "paytxfee" : 0.00000000,
    "relayfee" : 0.00005000,
    "errors" : "WARNING: abnormally high number of blocks generated, 190 blocks received in the last 4 hours (96 expected)"
}
drequinox@drequinox-OP7010:~/zcash/src$
```

Screenshot displaying the output of getinfo

New T addresses can be generated using the following command:

```
$ ./zcash-cli getnewaddress
t1XRCGMAw36yPVCcxDUrxv2csAAuGdS8Nny
```

GPU mining

Other than CPU mining, a GPU mining option is also available. There is no official GPU miner yet; however open source developers have produced various proofs of concepts and working miners. The Zcash Company held an open competition to encourage developers to build and submit CPU and GPU miners. No winning entry has been announced as of the time of writing.

Readers can get more information by visiting the website, `https://zcashminers.org/`.

There is another mining: using cloud mining contracts available from various online cloud mining providers. The cloud mining service providers perform mining on the customers' behalf. In addition to cloud mining contracts, miners can use their own equipment to mine via mining pools using stratum or other protocols. One key example is Zcash pool by NiceHash available at: `https://www.nicehash.com`. Using this pool, miners can sell their hash power.

An example of building and using a CPU miner on a Zcash mining pool is shown as follows.

Downloading and compiling nheqminer

The following steps can be used to download and compile `nheqminer` on an Ubuntu Linux distribution:

```
$ sudo apt-get install cmake build-essential libboost-all-dev git clone
https://github.com/nicehash/nheqminer.git
$ cd nheqminer/nheqminer
$ mkdir build
$ cd build
$ cmake .. make
```

Once all the steps are completed successfully, `nhequminer` can be run using the following command:

```
$ ./nhequminer -l eu -u <btc address> -t <number of threads>
```

`nhequminer` releases are available for Windows and Linux at the following link:
`https://github.com/nicehash/nheqminer/releases`

`nheqminer` takes several parameters such as location (`-l`), username (`-u`), and the number of threads to be used for mining (`-t`).

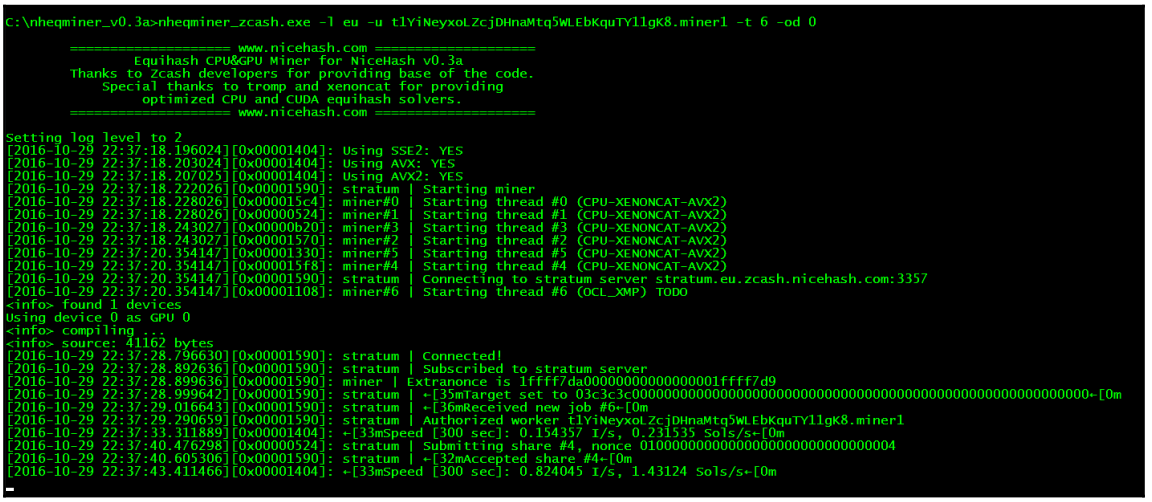
A sample run of Linux miner `nheqminer` for Zcash is shown as follows. In this screenshot the payout is being made to a Bitcoin address for selling hash power:

```
drequinox@drequinox-OP7010:~/nheqminer/nheqminer/build$ ./nheqminer -l eu -u 1PL6gsm49xCFMvrXqgGcee5cdrG119GoWN.worker1 -t 6 -od 0
Equihash CPU Miner for NiceHash v0.1c
Thanks to Zcash developers for providing most of the code
Special thanks to tromp for providing optimized CPU equihash solver

Setting log level to 2
[09:28:53][0x00007f51009cd700] stratum | Connecting to stratum server equihash.eu.nicehash.com:3357
[09:28:53][0x00007f51009cd700] stratum | Connected!
[09:28:53][0x00007f51009cd700] stratum | Starting miner
[09:28:53][0x00007f50fafce700] miner#1 | Starting thread #1
[09:28:53][0x00007f50fb7cf700] miner#0 | Starting thread #0
[09:28:53][0x00007f50f8fca700] miner#5 | Starting thread #5
[09:28:53][0x00007f50fa7cd700] miner#2 | Starting thread #2
[09:28:53][0x00007f50f97cb700] miner#4 | Starting thread #4
[09:28:53][0x00007f50f9fcc700] miner#3 | Starting thread #3
[09:28:54][0x00007f51009cd700] stratum | Subscribed to stratum server
[09:28:54][0x00007f51009cd700] miner | Extranonce is 5000e5b8000000000000005000e5b9ab
[09:28:54][0x00007f51009cd700] stratum | Authorized worker 1PL6gsm49xCFMvrXqgGcee5cdrG119GoWN.worker1
[09:28:54][0x00007f51009cd700] stratum | Target set to 01e1e1e1e0000000000000000000000000000000000000000000000000000000
[09:28:54][0x00007f51009cd700] stratum | Received new job #000000329b82d287
[09:28:55][0x00007f50fa7cd700] stratum | Submitting share #4, nonce 0200000000000000000000000000000
[09:28:55][0x00007f51009cd700] stratum | Accepted share #4
[09:28:55][0x00007f51009cd700] stratum | Ignoring non-clean job #000000329b82d2cc
[09:28:57][0x00007f50fafce700] stratum | Submitting share #5, nonce 0100000000000000000000000000001
[09:28:57][0x00007f51009cd700] stratum | Accepted share #5
[09:28:59][0x00007f50f97cb700] stratum | Submitting share #6, nonce 0400000000000000000000000000005
```

Using the BTC address to receive payouts for selling hash power

The screenshot, shown here, shows a sample run of `nheqminer` on Windows with payouts being made to a Zcash T address for selling hash power:



Using Zcash T address to receive payouts for selling hash power

Zcash has used ZKPs in an innovative way and they pave the way for future applications that require inherent privacy, such as banking, medicine, or the law.

This section completes the introduction to Zcash; readers can explore more about Zcash online at `https://z.cash`.

Initial Coin Offerings (ICOs)

ICOs are comparable to the **Initial Public Offering** (**IPO**). Just as an IPO is launched to raise capital by a firm similarly, ICOs are launched to generate money for a start-up project. The critical difference is that IPOs are regulated and fall under the umbrella of securities market (shares in the company) whereas ICOs are unregulated and do not fall under any strict category of already established market structures.

However, there are few suggestions that ICOs should be treated as securities in the light of some scam ICO schemes launched in the last few months and growing concerns around investor protection. Recently the **Securities and Exchange Commission** (**SEC**) suggested that all coins, ICOS, digital assets fall under the definition of *security*. This means that same laws would be applicable to ICOs, Bitcoin and other digital coins that are in applicable to securities. Also, an introduction of formal **Know Your Customer** (**KYC**) and **Anti Money Laundering** (**AML**) is also being recommended to addresses issues related to money laundering. Experts are recommending *Howey Test* as some criteria for any ICO to be considered a security.

Another difference is that ICOs by design usually require investors to invest using cryptocurrencies and payouts are paid using cryptocurrencies, most commonly this is the new token (a new cryptocurrency) introduced by the ICO. This can also be Fiat currency, but most commonly cryptocurrency is used. For example, in the Ethereum crowdfunding campaign a new token, Ether was introduced. The name token sale for crowdfunding is also quite popular and both terms are used interchangeably. ICO are also called crowd sales.

When a new blockchain based application or organization is launched, a new token can be launched with it as a token to access and use the application and also to gain incentives that are paid in the very same token that has been introduced by the ICO. This token is released to the public in exchange of some already established cryptocurrency (for example, Bitcoin or Ethereum) or Fiat currency. The advantage is that when the usage of the application or product launched increases the value of the new token also increases with it. This way the investors who invested initially gain a good incentive.

In the year 2017, ICOs have become a leading tool for raising capital for new start-ups. The first successful ICO was that of Ethereum which raised 18 million USD in 2014. A recent success is Tezos which made 232 million USD in a few weeks' time. Another example is Filecoin which raised more than 250 million USD.

The process of creating a new token has been standardized on Ethereum blockchain thus making it relatively easy to launch an ICO and issue new tokens in exchange of Ether, Bitcoin or some other cryptocurrency. This standard is called ECR20 and is described in the next section. It's worth noting that using ECR20 is not a requirement and a completely new cryptocurrency can be invented on a new blockchain to start an ICO, but ERC20 has been used in various ICOs recently and provides a comparatively easier way to build a token for an ICO.

Recently ICOs have also been offered via platforms other than Ethereum, such as NEM (`https://nem.io`) and Stellar (`https://www.stellar.org`).

ERC20 tokens

ERC20 token is an interface which defines various functions dictating the requirements of the token. It does not, however, provide implementation details and has been left to the implementer to decide. ERC is basically an abbreviation of **Ethereum Request for Comments** which is equivalent to Bitcoin's BIPs for suggesting improvements in Ethereum blockchain.

This is defined under EIP 20, which you can read more about here `https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md`.

Ethereum is becoming a platform for choice for ICOs due to its ability to create new tokens and with ERC20 standard, it has become even more accessible.

ECR20 token standard defines various functions which describe various properties, rules, and attributes of the new token. These include total supply of the coins, total balance of holders, transfer function, approval and allowance functions.

There are other standards such as ERC223, ERC777 and extension of ERC20 called ERC827 are also under development.
You can refer to the followings links to learn more:

- `https://github.com/ethereum/EIPs/issues/827`
- `https://github.com/ethereum/EIPs/issues/223`
- `https://github.com/ethereum/EIPs/issues/777`

Summary

In this chapter, we introduced you to the overall cryptocurrency landscape. We discussed a number of altcoins in detail, especially Zcash and Namecoin. Cryptocurrencies are a very active area for research, especially around scalability, privacy, and security aspects. Some research has also been conducted to invent new difficulty retargeting algorithms to thwart the threat of centralization in cryptocurrencies.

Further research needs to be carried out in the areas of privacy and especially scalability of blockchain.

Now you should be able to appreciate the concept of altcoins and various motivations behind them. We also discussed few practical aspects, such as mining and starting a new currency project, which hopefully will give you a strong foundation, enabling you to explore these areas further. Altcoins are a fascinating field of research and they open many possibilities for a decentralized future.

In the next chapter, we will see what smart contracts are and discuss relevant ideas and concepts that are essential to fully understand blockchain technology.