# Technical Basis of Digital Currencies

**Simon Sprankel**
**02 August 2013**

TECHNISCHE
UNIVERSITÄT
DARMSTADT

# Contents

# List of Figures

# 1 Introduction

Nowadays, more and more things are digitalised and their analogue counterparts become less important. E-books, digital music, digital newspapers, online lexica et cetera become much more important (cf. Figure 1.1 and Figure 1.2). Hence, it seems only a matter of time when money also becomes completely digitalised. How the digital counterpart will actually be called is not clear at all – each combination of the adjectives digital, virtual or electronic and the nouns currency, money or cash seem to be used. For this paper, we will mostly use the term digital currency for two reasons: First, because the most important system, Bitcoin, uses this term. And second, because the systems we are interested in allow users to exchange money from their digital system into traditional currencies like euro or dollar and vice-versa. This possibility of exchanging money justifies the term currency.

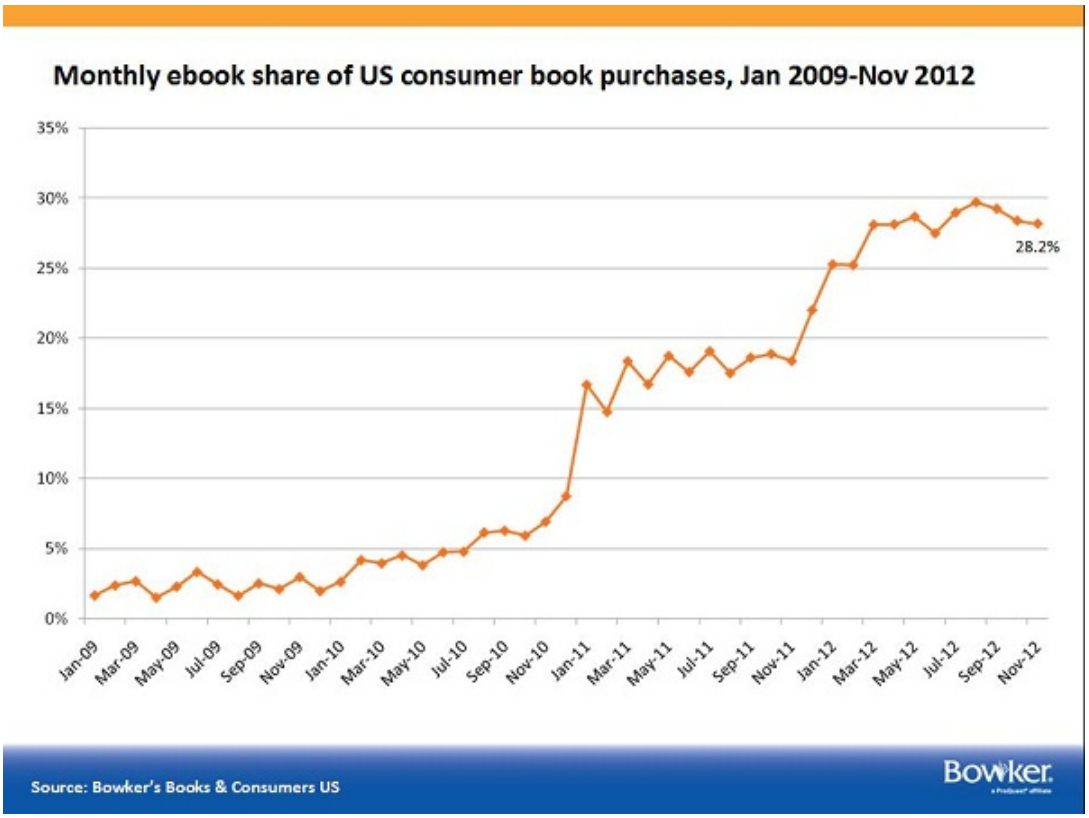


**Figure 1.1.:** Monthly E-book Share of US Consumer Book Purchases from Jan 2009 to Nov 2012

Although there is no clear naming convention, there is a categorisation of digital currencies. Guo and Chow came up with a quadrinomial categorisation in 2008 (cf. Figure 1.3) [GC08]. Type-1 virtual money is only spendable for virtual goods and services. Type-2 virtual money extends type-1 money in that it is also spendable for real goods and services. Type-3 virtual money extends type-2 money in that it is purchasable by real-world money. Type-4 virtual money extends type-3 money in that it is also sellable for real-world money. The European Central Bank invented another categorisation in their document about virtual currency schemes [Ban12]. Though it is just a slight modification and simplification of Guo's and Chow's categorisation, so that we will stick to theirs.

Type-4 digital currencies have multiple advantages over traditional currencies. Some of the mentioned benefits apply to all digital currencies, whereas many only apply to distributed ones like Bitcoin on which we focus. First of all, digital currencies have great accessibility. In order to get a credit card or something alike, people are normally in need of a bank account. In order to pay or sell with a digital currency, all what is needed is an internet access and the appropriate software. This is especially helpful for people from developing countries where the internet infrastructure may develop faster than the banking infrastructure. Another advantage is that one is not restricted by banks' opening hours or long transaction delays caused by them. Distributed digital currencies are furthermore nearly impossible to destroy. Of course a traditional bank has multiple backups and enormous protection, but it is still possible that all backups are destroyed.

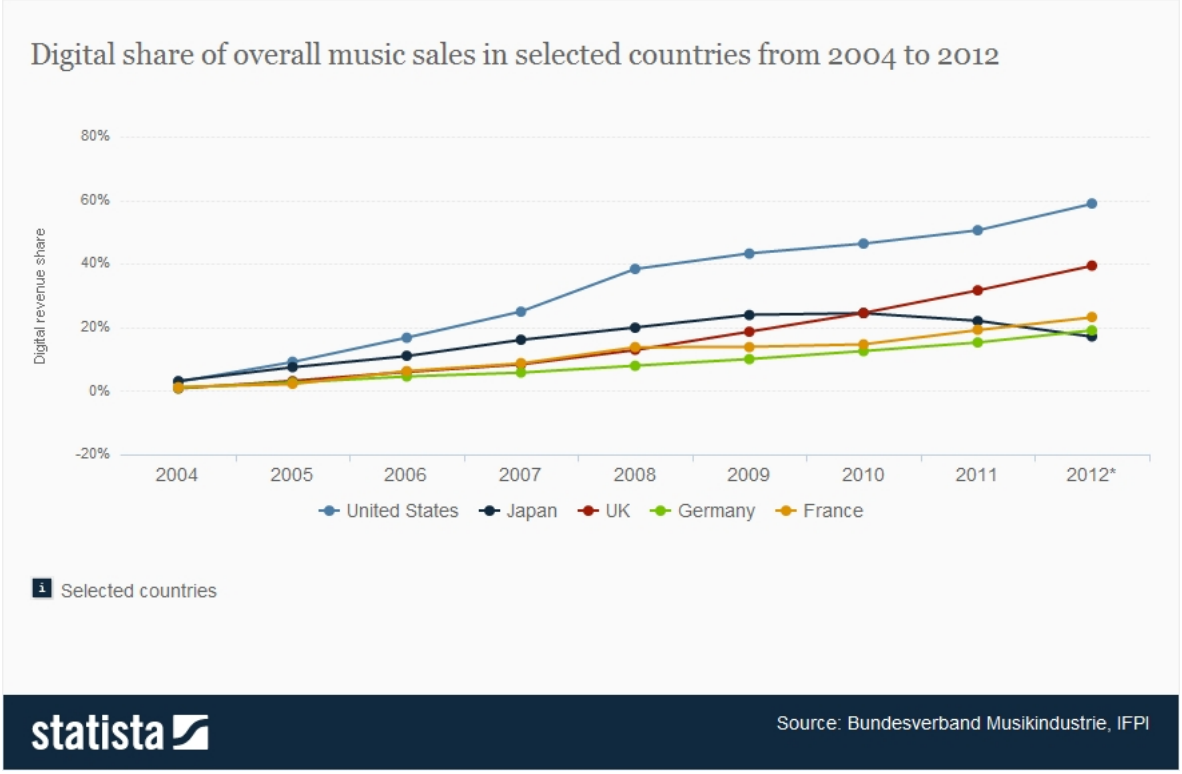


**Figure 1.2.:** Digital Share of Overall Music Sales in Selected Countries from 2004 to 2012

Compared to millions of distributed network nodes, the backups of a bank could be destroyed relatively easy. In addition, users are anonymous, so that the user's privacy is sustained. Users do not have to give any details about themselves – they act via a pseudonym. Another often mentioned advantage is irreversibility. Transactions, once executed, can not be charged back. This is great for merchants who may have not been able to start their business because of fraud problems.

Mind that some of the mentioned advantages can also be disadvantages in certain situations – especially the last two. The anonymity definitely attracts criminals to use the platform. Irreversibility can also be a problem for buyers who prepaid their order, but who are not supplied by the merchant. Hence, one always has to balance what is more important for oneself.

We are interested in full replacements of traditional money – in the so-called type-4 currencies. They can be used for purchasing digital as well as real goods and services and can be exchanged in both directions. Hence, systems like AceBucks, QQ Coins and WoW Gold will not be covered here. Additionally, we are dependent on open source systems and on the written technical information the inventors give us. This is not easy and often leads to non-scientific references. It furthermore excludes some systems which may be interesting due to their popularity like Ripple, Ukash, Ven or Linden dollars. These systems are all not open source and all lack enough technical information, so that they will not be covered. Additionally, most of these currencies still have a central authority which actually relativises some of their advantages over traditional currencies. In the end, only distributed cryptographic currencies, which are all based on Bitcoin, remain. The systems introduced here are basically the ones with the highest market capitalisation: Bitcoin (Chapter 3), Litecoin (Chapter 4), Peercoin (Chapter 5) and Primecoin (Chapter 6) [coi]. Namecoin, which is in third place, is not covered in this paper since it is mainly a distributed domain name system and has not been designed as a digital currency [Dot]. Primecoin is only in seventh place of the market capitalisation ranking, but introduces a new, innovative concept, so that the system is included in this paper.

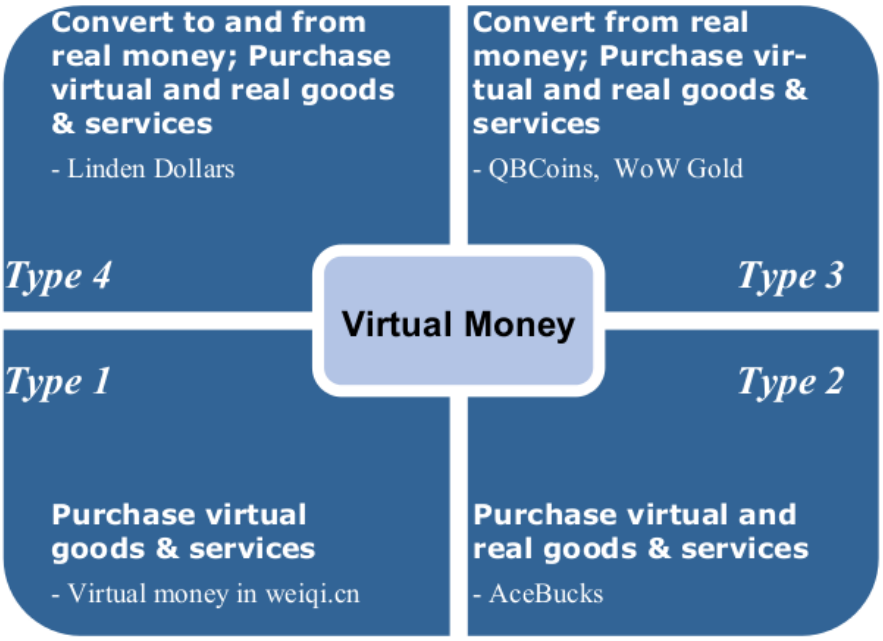


**Figure 1.3.:** Digital Currency Categories of Guo and Chow Exemplified [GC08]

# 2 Technical Background

Digital currencies make extensive use of basic cryptographic features such as public key infrastructures, digital signatures or hashing. Since these techniques are quite basic and popular, they are not covered in this paper. Therefore, rather unknown techniques are introduced in this chapter.

## 2.1 Proof-of-Work

The idea behind proof-of-work is that one can trust parties which prove that they completed some computational work in order to send a specific request. It has been developed in order to prevent denial-of-service attacks and e-mail spam, but can be used for other purposes as well. The proof has to require some computational work and the verification of the proof has to be rather easy. If work in the meaning of computational power is needed in order to send some form of request, an attacker cannot send an arbitrary number of requests. Hence, if the work has been proven, the receiver has a signal that the sender is potentially trustworthy. Anyway, this is not proof that the sender actually is trustworthy – it may just be used as an indicator.

Different proof-of-work systems mainly differ from their cost-functions; the functions which are actually responsible for the work. Cost-functions should be expensive to compute and efficient to verify. Ideally, the expensiveness of the computation is configurable via a parameter.

A popular proof-of-work system which is also used in Bitcoin is Hashcash [Bac02]. It was invented by Adam Back in 1997. The heart of the Hashcash cost-function is some kind of hash function. Back suggested SHA1 and MD5 in his paper, but more modern hash functions may be used nowadays – for instance, Bitcoin uses SHA256. The hash function just has to be efficiently computable. The idea of Hashcash is that the sender has to compute a hash with specific properties. The hash has to start with a configurable, but fixed number of zeros. In order to compute such a hash, a random number $x$, often called nonce, is added to the string to be hashed. If the resulting hash does not have the claimed requirements, $x$ is incremented and another hash is computed. The sender than sends the string as well as the computed number $x$ to the receiver which just has to check whether the hash of the string concatenated with the number fulfils the requirements.

## 2.2 Merkle Trees

Although Merkle trees are as old as 1987, they still play an important role in both – applications such as Apache Cassandra [The] or BitTorrent [Bak09] as well as in research [WLL+13, JO13, Nak08]. The idea of Merkle trees, which are also referred to as hash trees, is to store data blocks in the leaf nodes and only hash values in the non-leaf nodes [Mer88]. The hashes are calculated by a conventional encryption function such as DES. The hash values of the leaf nodes containing the data are the respective hashes of the data. The hash values of the other nodes are calculated by hashing the hash value of the left child concatenated with the hash value of the right child.

The big advantage of this model is that one can discard hashes and data blocks without breaking the hash of the root node. Considering the example from Figure 2.1, it is possible to remove H_00, H_01 as well as DATA_000 and DATA_001 without breaking the hash of the root node. This is extremely helpful if new data is added to the right of the tree and the old data is not that important. Bitcoin makes extensive use of this technique.

## 2.3 Cunningham Prime Chains and Bi-twin Chains

The three prime chains introduced in this section are important for the Primecoin system which is discussed in Section 6. Prime chains are basically just sequences of primes $p_i$ with a specific chain length $n$: $p_1, p_2, \ldots, p_n$.

A Cunningham chain of first kind is a prime chain with the following property:

$$\forall i \in \mathbb{N} \;\; 1 \leq i < n \rightarrow p_{i+1} = 2p_i + 1$$

The so-called origin of such a chain is $p_1 + 1$. For instance, $2, 5, 11, 23, 47$ is a Cunningham chain of first kind with length 5 and origin 6.

A Cunningham chain of second kind is a prime chain with the following property:

$$\forall i \in \mathbb{N} \;\; 1 \leq i < n \rightarrow p_{i+1} = 2p_i - 1$$

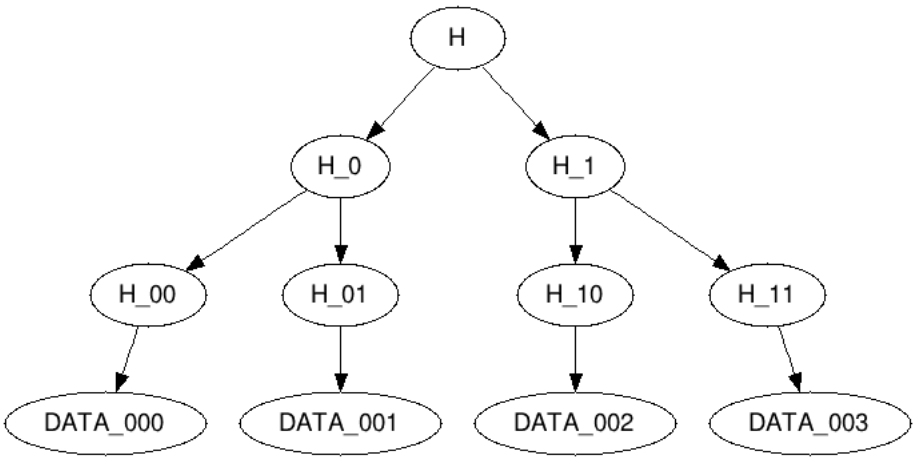


**Figure 2.1.:** Example Merkle Tree

The origin of such a chain is $p_1 - 1$. For instance, $2, 3, 5$ is a Cunningham chain of second kind with length 3 and origin 1.

Twin primes are pairs of primes where $p$ as well as $p+2$ are primes. Bi-twin chains are chains of these twin primes with the property that the doubled average of a twin prime is the average of the next twin prime in the chain. So assuming a prime chain of length $2n$ with $p_{1,1}, p_{1,2}, \ldots, p_{n,1}, p_{n,2}$, the property for a bi-twin chain is:

$$\forall i \in \mathbb{N} \;\; 1 \leq i < n \rightarrow p_{i,1} + p_{i,2} = \frac{p_{i+1,1} + p_{i+1,2}}{2}$$

The origin of a bi-twin chain is the middle of the first twin prime, which is actually $p_{1,1} + 1$. For instance, $5, 7, 11, 13$ is a bi-twin chain of length 4 and origin 6.

## 2.4 Fermat Primality Test

The classical Fermat test and the Euler-Lagrange-Lifchitz test are used in Primecoin, whereby the former one is much more important and therefore introduced here. The Fermat primality test checks whether a number is probably a prime number. Numbers which pass the Fermat test and are not prime numbers are called pseudoprimes.

The Fermat primality test is based on Fermat's little theorem, which states that if $p$ is a prime and $1 \leq a < p$, then:

$$a^{p-1} \equiv 1 \;\; (mod p)$$

Hence, the test just picks random numbers $a$ with $1 \leq a < p$ and checks the equality. If the equality does not hold for one of the numbers, then $p$ is not a prime. If the equality holds for many numbers, then $p$ is probable prime.

# 3 Bitcoin (BTC)

Bitcoin is a decentralised type-4 P2P currency based on cryptography [Nak08]. It has been invented by a person with the pseudonym Satoshi Nakamoto in 2008. It is plainly *the* digital currency, because it was already invented in 2008 and has, with around 30,000 to 70,000 transactions per day and a market capitalisation of about 1.2 billion USD, a massive dissemination nowadays [blo, coi]. There is a vast number of other cryptocurrencies, though they all have not been developed from scratch, but are based on Bitcoin [Wik]. It is even possible to use the Bitcoin system for other purposes than digital currencies: Namecoin is based on Bitcoin and uses the system in order to build a decentralised domain name system [Dot].

Bitcoin is based on public key cryptography. Hence, each Bitcoin user has a secret private key and the appropriate public key, which is known to all other users[1]. There are three concepts which are useful in order to get a quick overview and which are essential to understand the Bitcoin system: transactions, blocks and the block chain. A transaction is a dataset which describes a money sending process from a sender to a receiver. The sender signs the transaction, so that everyone can verify that the message is authentic. The receiver's public key is included into the transaction dataset, so that everyone can check who the new owner of the transferred bitcoins is. A block is a dataset which contains multiple transactions. It includes, besides the transactions, a proof-of-work as well as a reference to the previous block. Since it contains a proof-of-work, a new valid proof has to be found in order to create a new block. The difficulty for the proof-of-work is adjusted every 14 days, so that on average, every ten minutes a new block is generated. Why new blocks have to be generated is discussed in the following section. The block chain is the chain of all blocks ever generated and therefore contains the whole transaction history. Each block includes the hash of the previous block in order to build this chain. The first block in this chain is the so-called genesis block. Bitcoin's genesis block has been established on 03 January 2009.

It may help to read the Bitcoin glossary in the Appendix in order to get used to the other Bitcoin terms as well.

## 3.1 How to Get Bitcoins

First of all, it should be clarified what a coin in the Bitcoin-world actually is. A coin is nothing else than a chain of digital signatures which represents a transaction history. Hence, a proof that a coin belongs to someone is not a proof of some knowledge or property, but just a proof that you got this coin at sometime from someone. In other words, a coin does not exist through a physical or virtual representation, but only through its history. This is nothing else than the logical implication of electronic money. Even traditional money nowadays often only exists in a virtual form. The user does not have all money in cash, but only sees the balance on the (digital) bank account. Since the whole transaction history is stored on each node in the Bitcoin network, every node is able to verify the current owner of a coin. But how do new coins come into the system? Each time a Bitcoin user solves a proof-of-work and thus creates a new block, he is rewarded with a specific number of bitcoins. This is an incentive for the users to utilise their computational work and solve the proof-of-work puzzles. The whole process of trying to solve proof-of-work puzzles and thus creating new blocks is called mining.

Although a new block is created every ten minutes, the total number of coins is strictly limited to 21 million. In order to satisfy this limitation, the reward a user gets for creating a new block is roughly halved every four years (exactly once every 210,000 blocks). Figure 3.1 illustrates that the reward for creating new blocks rapidly tends to zero. In 2140, when all 21 million bitcoins are mined, the reward will be less than the smallest unit of bitcoins (which is one Satoshi and equals 0.00000001 BTC), so that no more coins are created. The number of available bitcoins over time are analogous and are shown in Figure 3.2. Even when all bitcoins are mined, there is still another incentive for users to solve the proof-of-work puzzles. The user who mines a block gets the transaction fees of all included blocks. What transaction fees there are and what they are good for is described in the following section.

But of course there is another possibility of getting bitcoins apart from mining. Users can also exchange traditional money like euros or dollars into bitcoins on specific exchange markets. The largest and most popular Bitcoin exchange platform is Mt.Gox[2].

[1] To be exact, users have a various number of keypairs, because a new keypair is used for each transaction. Though we will keep it a bit simpler here.

[2] `https://www.mtgox.com/`, accessed 01 August 2013

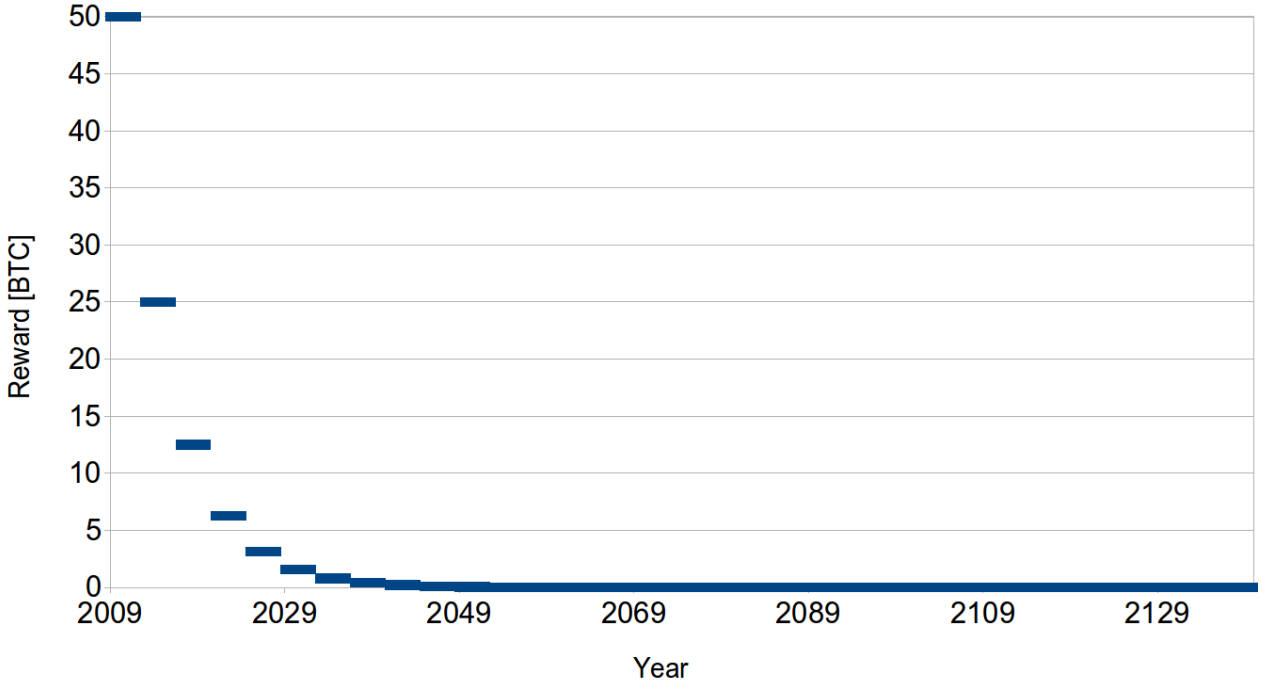


**Figure 3.1.:** Reward for Creating a New Bitcoin Block Subject to Time

## 3.2 The Transaction Process

Assumed Alice wants to send 50 BTC to Bob and previously got 30, 15 and 10 BTC from Charlie, Dan and Erin respectively. She will then create a transaction record similar to the one from 3.1. Inputs are the three transactions which prove that she has enough money. To be exact, these are hashes of the previous transactions as well as a respective index, which references a specific output from the corresponding transaction. The signatures which verify that Alice is authorised to spend the funds are also included in the inputs. The outputs contain the public key from Bob and the amount which should be transferred to that key. If the sum of the transaction input amounts do not match the amount she wants to send, she can create an additional transaction output that sends the remainder back to her own public key (3 BTC in the example). If the sum of the amounts of all transaction inputs and outputs still does not match, the remainder is considered to be a transaction fee (2 BTC in the example). This fee is an incentive for other users to confirm the transaction. The dots in 3.1 indicate that there are some other, less important fields in a transaction record.

$$\left( \underbrace{\begin{pmatrix} txHash_{CA} & txOutIndex_{CA} & sig_{CA} \\ txHash_{DA} & txOutIndex_{DA} & sig_{DA} \\ txHash_{EA} & txOutIndex_{EA} & sig_{EA} \end{pmatrix}}_{\text{Inputs}}, \underbrace{\begin{pmatrix} pk_B & 50 \\ pk_A & 3 \end{pmatrix}}_{\text{Outputs}}, \dots \right) \tag{3.1}$$

A transaction is verified if a user who mined a block by finding a correct proof-of-work has included this transaction in his block. He will then get the transaction fees of all included transactions. Since new blocks are created every ten minutes on an average, the transaction verification can also last ten minutes and even longer. One can increase the probability for a faster verification by providing a higher transaction fee. Then it is more likely that the transaction is included in the next created block.

The first transaction in a block is a special one. It includes the reward for the user who mined this block. It is a transaction with no inputs. As an output, it includes the current reward amount together with the public key of the miner.

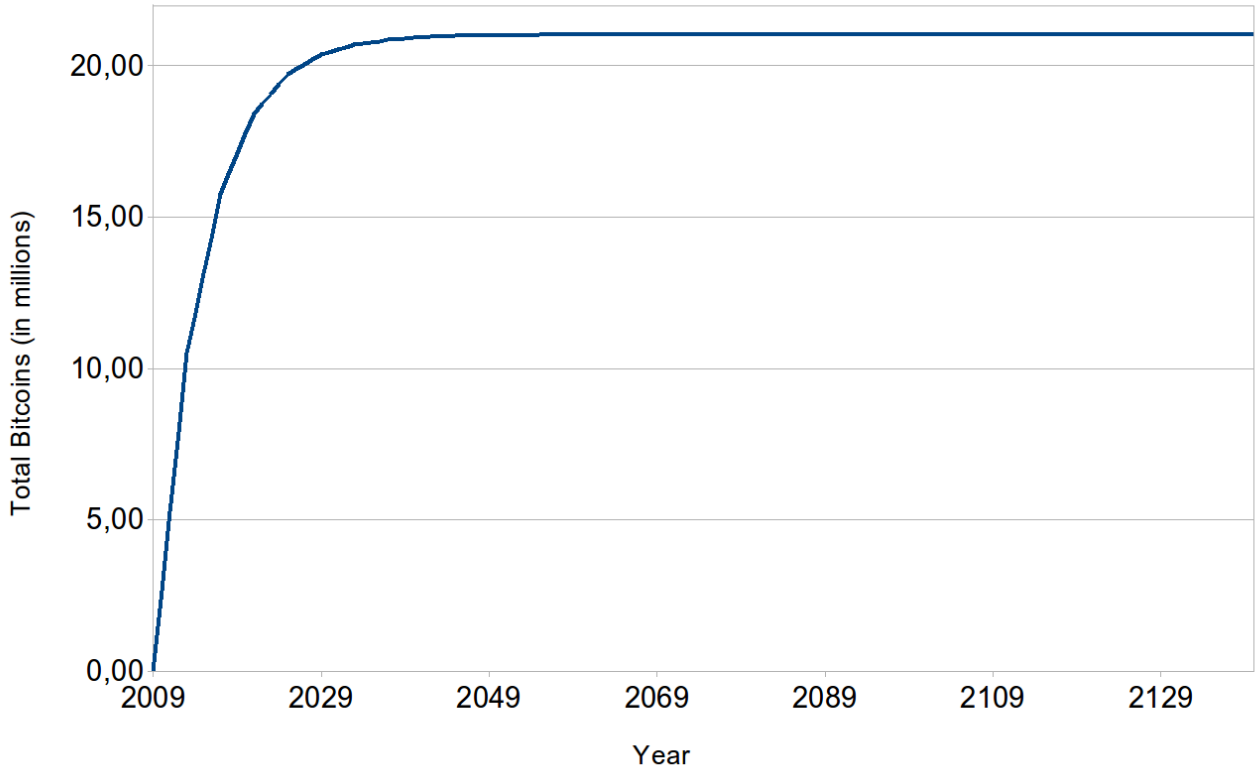


**Figure 3.2.:** Total Bitcoins Subject to Time

## 3.3 Double Spending Prevention

Dishonest users could use the same transactions as input for multiple new transactions. In other words, they could try to double-spend the money they eventually got. Actually this would work in the very first moment, but all transactions except the first one will not be verified and would therefore not be executed. When transactions are included into blocks, they are checked by the block creator as well as by all users receiving the new block. Since the current block chain which includes all successful transactions is broadcasted to all users, everyone has the whole transaction history. A user receiving or creating a new block checks the input transactions of all included transactions. If the transaction has already been used as an input in another transaction, the transaction and the whole block will be rejected.

Another possibility to double spend money would be to edit the whole block chain in order to cheat the other users. The dishonest user could edit an old block by removing the transaction where he already spent his money. Anyway, removing or editing a transaction would result in a new block hash, so that the proof-of-work of this block would have to be solved again. Since all blocks are arranged in a chain, the dishonest node would also have to solve the proof-of-work of all newer blocks than the manipulated block. And since users always accept the longest chain and reject all others, the dishonest user would have to control at least half of the network's computational power in order to build a longer chain than the chain of the honest nodes. But if a dishonest user controls more than half of the computational power, he could manipulate transactions and double-spend his money.

## 3.4 Problems and Possible Solutions

The fact that a completely new and well-known digital currency has not been invented since the launch of Bitcoin in 2008 shows that it is a brilliant system. Though there are known problems which are discussed in this section. Some problems have been solved in Bitcoin forks and some of these forks will be introduced in the next chapters.

- **energy consumption**
  Bitcoin's security heavily depends on the computational power needed for a valid proof-of-work. This is not very energy-efficient. No matter how the current difficulty is, the creation of a new block will take approximately ten

minutes and will be done over and over again. Hence, each Bitcoin user's CPUs/GPUs or specialised processors run all the time and waste a significant amount of energy. Even if the devices doing the calculations are rather energy-efficient, this of course is not a good design from an ecological point of view.
Peercoin suggests an addition to the proof-of-work system called proof-of-stake, which ensures energy-efficiency in the long run. Peercoin is discussed in Chapter 5.

- **anonymity**
  People may think that Bitcoin is completely anonymous, because it uses pseudonyms and uses new addresses for each transaction. Unfortunately, it is still possible to reveal identity information in certain cases. The main problem is that the whole transaction history is public. Assumed an attacker runs an exchange service as well as a trap or just sells some products or services. If a user buys bitcoins from this exchange service and uses the same transaction for buying a product from the attacker, the attacker can link the identity from the exchange service to the order. The common solution to this problem is to use so-called mixer services. Users can send money to these mixers. The mixers will do a few transactions with the money and will refund it to the sender afterwards. Subsequently, it is not possible to link the identity of the user to the transaction. But of course there is another problem that there may be untrusted mixers which do not refund the sent money.
  Therefore, Barber et al. propose another solution based on a fair exchange protocol [BBSU12]. Since it is not implemented in a known system yet, it is not further discussed here.

- **history-revision attack / double spending**
  The double spending problem has also been discussed in Section 3.3. But it has been admitted, that such an attack is possible when an attacker controls more than half of the network's computational power. Barber et al. propose a solution to that problem based on checkpointing [BBSU12]. The idea is that users should not trust block chains in which rather old transactions have been altered. Since each user gets new transactions and blocks through broadcasting, they have first-hand information about the block chain. They could make regular checkpoints of the block chain and if these older parts stored in the block chain are updated by some user, they should be quite sceptical. They could then require a high number of other users to verify this chain before they accept it. Something similar is also included in the current protocol, but the checkpointing is done by software developers and included in the software download. This is of course again a security problem, so that Barber et al. have suggested to base this mechanism on the private, first-hand transaction history. Unfortunately, this idea has not been implemented in any known system.

- **theft or loss of bitcoins**
  Users sign their transactions with their private keys and authenticate the transaction thereby. Hence, if private keys are stolen, the coins of the user can be transferred to another user. If private keys are lost, the user cannot authenticate any transaction with his coins, so that the coins are useless.
  In order to solve the theft problem, Barber et al. propose threshold cryptography and super-wallets [BBSU12]. Threshold cryptography means that the private keys are not stored on one device, but rather are split and stored on multiple devices. This makes a malware attack more difficult, but also reduces the usability of the system. An addition to that idea and a solution to the usability problem are super-wallets. The super-wallet would be implemented with the mentioned threshold cryptography and only a sub-wallet with a small amount of coins would be on the user's smartphone, so that money can be spent easily. If the user loses his smartphone or if the smartphone is attacked, he would only loose the small amount of money and his super-wallet with all his money would not be threatened.
  The solution ideas for the loss of bitcoins proposed by Barber et al. are rather straightforward like backups, encryption and trusted paths and are therefore not covered here. Encrypting the wallet file, which stores all private keys, is supported in the current client, but the user has to opt-in [Bit]. The threshold cryptography idea has not yet been implemented in a known system.

- **scalability**
  Scalability is potentially an issue for clients with limited hardware or network resources like clients with smartphones. It is rather difficult to download and store the whole block chain, which as the time of writing is approximately 8.7 GB big, on the smartphone [blo]. Barber et al. propose a subscription-based filtering service as a solution [BBSU12]. The idea is that clients who do not verify any transactions are not interested in all new transactions and blocks, but only the ones relevant to them. They could subscribe to a filtering service which filters the updates and only forwards the ones the user is interested in. Bitcoin supports such a filtering service and the feature is also implemented in the current Android Bitcoin wallet [HC12, Sch13].

## 3.5 Overview of Bitcoin Forks

There is a huge number of Bitcoin forks, which enhance it in some way. The following table gives an overview of the forks, which are introduced in the next chapters.

| Currency | Proof-of-work Algorithm | Mining Time | Current Reward Per Block | Maximum Volume | Current Market Capitalisation [coi] |
|---|---|---|---|---|---|
| **Bitcoin** | SHA256 | 10 minutes | 25 BTC | 21 million | ≈ 1.254 billion USD |
| **Litecoin** | scrypt | 2.5 minutes | 50 LTC | 84 million | ≈ 59 million USD |
| **Peercoin** | SHA256 | 10 minutes | 50 PPC | no limit | ≈ 3 million USD |
| **Primecoin** | different approach | 1 minute | 50 XPM | no limit | ≈ 864,814 USD |

# 4 Litecoin (LTC)

Litecoin is a Bitcoin fork whose genesis block was mined on 13 October 2011. It basically alters two parameters of the Bitcoin system. First, it quarters the block mining time from ten minutes to two and a half minutes. Second, it quadruples the number of blocks after which the reward for mining new blocks is halved from 210,000 to 840,000. Additionally, it uses another proof-of-work function. Instead of SHA256 used by Bitcoin, it uses scrypt – a password-based key derivation function invented by Colin Percival in 2009 [Per09].

The smaller block rate and the higher number of blocks after which the reward is halved lead to a greater maximum number of coins. Where 21 million is the maximum number of possible bitcoins, the maximum volume of litecoins is 84 million. Bitcoin and Litecoin have the same reward distribution (cf. Figure 3.1). The two adjustments cancel each other out in this point. Only the years on the x-axis are shifted by two years, because the Litecoin genesis block was mined two years later than the one from Bitcoin. The function showing the litecoins over time has the same characteristics as the one from Bitcoin. But it is also shifted by two years and the value to which the function tends is four times higher (cf. Figure 4.1).

The usage of scrypt as a function for the proof-of-work means that it is more profitable to mine litecoins with home hardware like normal workstations. For Bitcoin mining, there is specialised hardware with which the mining process is much faster. Therefore, Bitcoin mining with home hardware as a normal user becomes much more inefficient, because the probability that people with specialised hardware find a valid proof-of-work is much higher. With Litecoin, the probability that one finds a proof-of-work with home hardware is much higher, because there are no competitors with much better preconditions.

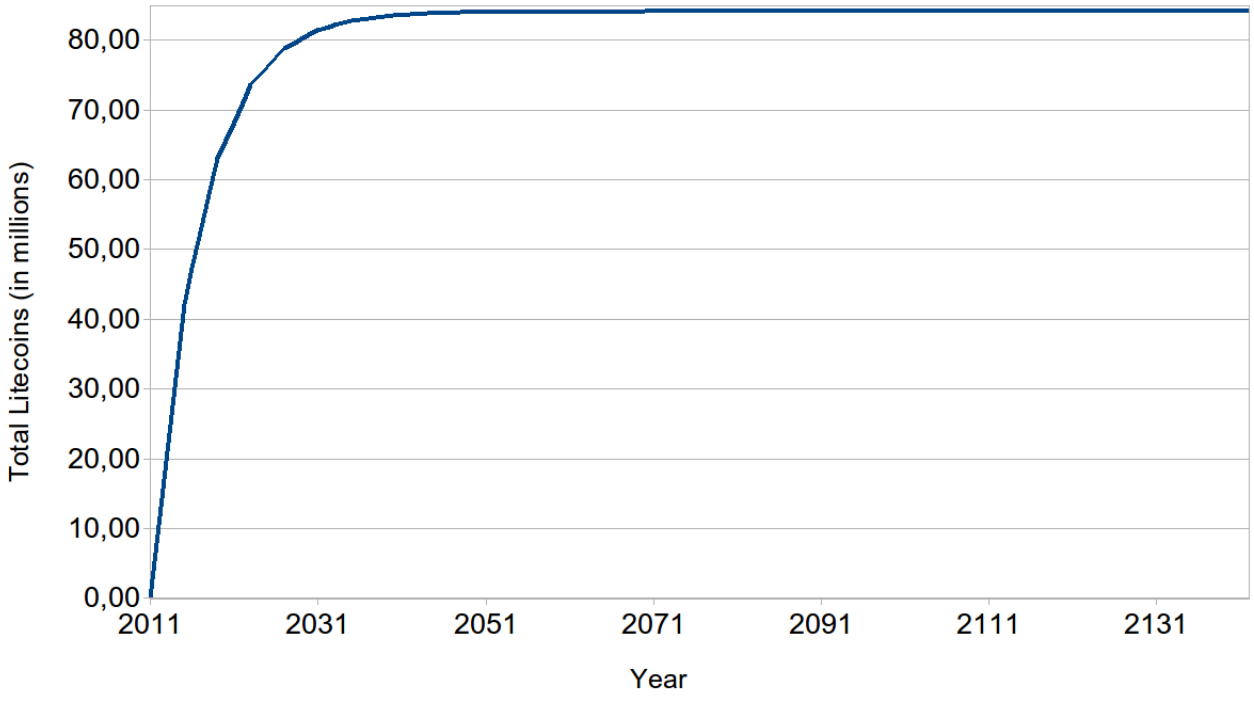


**Figure 4.1.:** Total Litecoins Subject to Time

# 5 Peercoin (PPC)

Peercoin, often referred to as Peer to Peer Coin or PPCoin, has been invented by two people with the pseudonyms Sunny King and Scott Nadal in 2012 [KN12]. It is another Bitcoin fork, which solves Bitcoin's energy consumption problem in the long run. Bitcoin's proof-of-work-based system depends on the computational power of all network nodes. Every node computes hashes over and over again in order to find a valid proof-of-work. Of course this is not very energy-efficient. Peercoin suggests a hybrid system which uses proof-of-work for the mining process, but a new concept called proof-of-stake for security purposes. Whereas Bitcoin nodes solve proof-of-work puzzles and therefore waste computational power, Peercoin mainly uses energy-efficient proof-of-stake in the long run.

Peercoin's proof-of-stake is based on the so-called coin age, which is defined as currency amount times holding period: $coinAge = currencyAmount \cdot holdingPeriod$. So the coin age can be computed, each transaction and each block has a timestamp. Peercoin has the traditional proof-of-work blocks known from Bitcoin as well as a new type of blocks: the so-called proof-of-stake blocks. Finding proof-of-stake blocks is also based on hash operations, but in contrast to the proof-of-work blocks, there is no nonce in the data which is hashed. The variable part of the data which is hashed is the timestamp field. This field only changes every second, so that users just calculate a hash every second and not millions of hashes per second like in Bitcoin. The probability for a user to find a valid proof-of-stake block depends on the total accumulated coin age. A proof-of-stake block is accepted if a user found a hash fulfilling his specific target – meaning that the hash is less than his current target. Since the current target is multiplied by the coin age the user accumulated, users with a higher coin age are more likely to find a valid proof-of-stake. One could actually pre-calculate the hashes with the respective timestamps, but this would not help for two reasons. First, the network only accepts blocks with a timestamp near to the current timestamp. Second, the hash of the previous block is also included in the header which has to be hashed, so that one would have to recalculate the hash as soon as another block has been mined. Hence, a pre-calculation does not make any sense, so that users will stick to the one-hash-per-second rule and the energy efficiency is maintained.

Of course users get a reward for mining proof-of-stake blocks. Transaction fees are indeed compulsory, but they are not declared to the miner. They are "destroyed" insofar as they are charged, but not declared to anyone. They act as a counterpart to the inflation caused by the so-called coinstake transactions. If a user mined a block by solving a proof-of-stake puzzle, the first transaction in this block will be the coinstake transaction. It is a transaction in which the user spends his coins and issues them back to himself. But he does not only get back the coins he spent, but is rewarded one cent per coin year on top. In other words, the reward can be calculated as

$$reward = coinDays \cdot \frac{1}{36,500} \text{ PPC}$$

Hence, assumed a user held 100 PPC for two years, accumulating $100 \cdot 365 \cdot 2 = 73,000$ coin days and finds a proof-of-stake block, he will be rewarded with $73,000 \cdot \frac{1}{36,500} = 2$ PPC. Since this reward is paid every time a user finds a proof-of-stake block, Peercoin does not have a fixed volume cap. There is a potentially unlimited number of coins, but the growth rate is actually limited by the block mining rate.

Another change compared to Bitcoin is that the proof-of-work and proof-of-stake hash target respectively the difficulty is adjusted continuously and not discretely every 14 days as in Bitcoin. This avoids sudden jumps of the difficulty. Additionally, since the whole system is based on the coin age, the definition of the longest chain has to be adjusted. This is important for the security model – if a node receives multiple chains, it has to rely on the longest chain. The longest chain in Peercoin is defined as the chain with the highest total consumed coin age. For ensuring that the correct chain prevails, Peercoin additionally has a checkpointing mechanism. Unfortunately, this mechanism is not distributed, but depends on a central authority. Whereby the developers claim that it was just an additional security feature for the beginning of the currency and can be removed soon [PPC].

# 6 Primecoin (XPM)

Primecoin is a quite innovative, Bitcoin-based currency which was invented in 2013 by someone with the pseudonym Sunny King, who also invented Peercoin [Kin13]. It is the first cryptographic currency whose proof-of-work function is not based on Hashcash and does not waste energy just in order to ensure the security of the currency. It invests the energy for a scientific purpose – for finding prime chains. Unfortunately, there is not much scientific literature regarding prime chains. We did not find any scientific evidence that prime chains are important for modern science. Hence, the claim that finding prime chains has a scientific value stands a claim for the moment. There is an explanation on the Primecoin FAQ, though it lacks external references which prove the claims [Pri]:

> *The distribution of primes has been one of the most important discoveries in arithmetic, and the study of prime chains traces its lineage to the work of Riemann and prime number theorem, with connections to the deeper nature of the seemingly random pattern of prime distribution. Prime distribution is not just an abstract interest of mathematicians. Riemann's study revealed connections between Riemann zeta function and prime distribution, whereas later on Riemann zeta function has been shown to be highly relevant in other scientific disciplines such as physics, thus the study of prime distribution is an important part of the foundation of modern sciences.*

As stated above, Primecoin uses another proof-of-work function than Bitcoin. Such a function basically has to meet four requirements:

1. It has to be hard to compute, so that one can not create a proof in no time.
2. The difficulty needed in order to solve the proof has to be adjustable in a linear way. Otherwise, the function could not adapt to a changing amount of computational power.
3. It has to be easy to verify, so that everyone can efficiently check if a proof is valid.
4. There has to be a possibility to prevent that proofs are used multiple times. Hence, it has to be possible to link some input of the function to the hash of the block header.

Primecoin's proof-of-work based on prime numbers achieves that in the following way:

1. Prime chains are hard to compute, so the calculation of prime chains is used as a proof-of-work algorithm.
2. The chain length does not result in a linear difficulty curve and is hence not well suited. Instead, pseudoprimes are allowed and the remainder of the Fermat test is used to construct an appropriate difficulty function. If $k$ is the prime chain length, $p_0, \ldots, p_{k-1}$ the prime chain and $r$ the remainder of the Fermat test, $d = k + 1 - \frac{r}{p_k}$ is the difficulty. As the author states, "the distribution of $\frac{r}{p_k}$ is not strictly uniform, but experiments have shown that the difficulty adjustment is reasonably good in practice" [Kin13].
3. In order to verify the proof-of-work, one only has to perform the Fermat test with every prime chain element $p_0, \ldots, p_{k-1}$. If all elements are primes or pseudoprimes, one performs the Fermat test with $p_k$ and plugs in the remainder into the equation shown above. The complexity for this verification is, compared to the computation of such a proof, relatively low. Primecoin only uses $a = 2$ for the Fermat test.
4. In order to link the input of the proof-of-work function to the hash of the block header, it is necessary that the origin of the prime chain is a multiple of the block header hash.

A miner therefore takes multiples of the block header hash and checks if it is possible to construct a Cunningham chain of first or second kind or a bi-twin chain with this value as an origin and with a specific required length. The required chain length is the integer value of the current difficulty (9 as the time of writing, because $\lfloor 9.27 \rfloor = 9$). If a chain with the given origin and the required length is found, the miner computes the difficulty of the found chain. If this difficulty is greater than the current network difficulty, the miner found a valid block and can broadcast it to all other nodes in the network. The algorithm for finding the prime chains is not further explained, but as a look into the code reveals, it seems to be based on the Sieve of Eratosthenes.

Another difference to Bitcoin is that blocks are approximately mined each minute and therefore ten times more often than in Bitcoin. This also means that transactions are confirmed ten times faster. Additionally, the total volume of coins

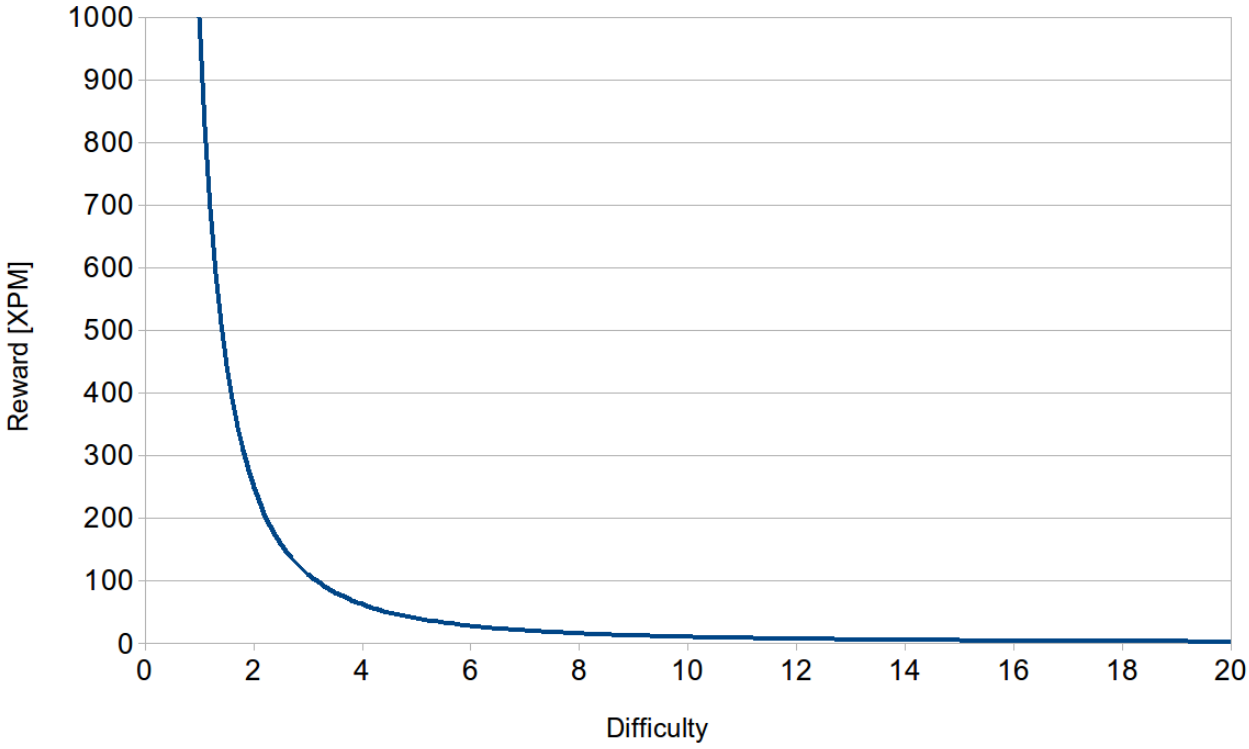


**Figure 6.1.:** Reward for Creating a New Primecoin Block Subject to Time

is not limited. The reward for mining a new block is always equal to 999 divided by the square of the current difficulty. Hence, as the difficulty increases, the reward will converge to a specific value (cf. Figure 6.1). As the time of writing, the difficulty is roughly 9.27 and the according reward for creating a new block is about 11.63 XPM.

# 7 Conclusion and Future Work

The digital currency market is very turbulent. New digital currencies are released very often. Even during writing this paper, Primecoin, which has meanwhile become an important currency, had been released. Though there is not much innovation on this market. All new systems are based on Bitcoin and often alter it just slightly. The only systems which are really innovative since the launch of Bitcoin in 2008 are Peercoin and Primecoin. Peercoin, because it introduces the new concept of energy-efficient proof-of-stake and Primecoin, because it introduces the new concept of proof-of-work with a scientific purpose. More innovation on the market would be desirable. The high movement of the market shows that it is not yet a stable market. People who invest in this market should be quite careful. People with experiences in digital currencies often state that one should only invest money which one can loose without severe consequences.

It remains the question whether Bitcoin is the only possibility to implement a decentralised digital currency. Maybe there are also other concepts which are suited for developing such a currency. Additionally, even if this is the only possibility, which other concepts could replace the rather useless default proof-of-work function? Are there other scientific concepts which could be used, so that the energy is not wasted that useless? Or are there possibilities to replace the proof-of-work mechanism with another one which is energy-efficient and also ensures the network security? These are open questions which would be worth working on. It would be desirable that the topic of digital currencies would gain more scientific attention. Currently, most of the publications in this area are not scientific.

To conclude this paper, we want to pose the question whether it is really good to have a currency which can by definition not be regulated in any way. Assuming that traditional currencies will be replaced by a few digital ones: What will happen if an error is found in these new currencies – either in their protocol definition or in their implementation? Or if some attacker accomplishes to create a longer block chain or such a chain emerges through a bug? Answers to these question are to be found before these systems replace our traditional currencies.

# A Appendix

## A.1 Bitcoin Glossary

block
: record of multiple transactions. Contains a reference to the previous block, a timestamp, the difficulty in the sense of the number of leading zeros, the nonce proving the work, the transactions and some less important data.

block chain
: chain of blocks in which all transactions ever executed are stored. The blocks are chained by storing the hash of the respective previous block.

BTC
: the currency code for Bitcoin.

cBTC
: abbreviation for centibitoin or bitcent, which is equal to 0.01 BTC.

coin
: see electronic coin.

difficulty
: measure of how difficult it is to solve a proof-of-work puzzle. Is negatively correlated to the target. As the size and overall power of the Bitcoin network grows, the difficulty is increased, so that an appropriate hash is found each ten minutes on average.

double spending
: problem that dishonest nodes may try to spend their coins multiple times.

electronic coin
: a chain of digital signatures.

genesis block
: the first block which is created; the first block in the block chain.

longest chain
: the “longest” block chain in the network. “Longest” means that it is the chain in which the most proof-of-work is included.

mBTC
: abbreviation for millibitoin or mbit or millibit, which is equal to 0.001 BTC.

microBTC
: abbreviation for microbitoin or ubit or microbit, which is equal to 0.000 001 BTC.

mining
: is the creation of new blocks. Hence, it describes the solving of proof-of-work puzzles and the creation of new coins.

nonce
: is the number with which the hash outcome is altered. In the proof-of-work procedure, different nonces are added to the hash input until the resulting hash fulfils the current difficulty.

proof-of-work
: mathematically-based proof that some work in the sense of computational power has been needed in order to find an appropriate value. It is very easy to check that this work has actually been done, but in order to redo the work, the whole calculation has to be repeated. See Section 2.1 for a more detailed explanation.

Satoshi
: the smallest fractional amount of a BTC (named after the pseudonym of the Bitcoin inventor Satoshi Nakamoto), which is equal to 0.000 000 01 BTC.

target
: 256 bit hash. The hash has to be less than this target in order to be accepted as a proof-of-work. The lower this number is, the higher the average computational power is needed in order to find a hash fulfilling the requirements..

transaction
: signed record of data. The included data contains the inputs as well as the outputs. Inputs are the transactions which prove that the user earned the money which he wants to spend. Outputs define the receivers of the money. They may contain a reference to the sender itself in order to define the change the sender gets. The input amount normally does not match the output amount. The difference is the transaction fee and an incentive for the user who confirms the transaction. Multiple transactions are bundled into blocks.

transaction fee
: a typically small amount of money, which the sender of a transaction offers the block creator who includes the transaction into his block. The block creator gets the transaction fees of all included transactions.

# Bibliography


[Bac02] Adam Back. Hashcash – a denial of service counter-measure, August 2002. http://www.hashcash.org/papers/hashcash.pdf, accessed 01 August 2013.

[Bak09] Arno Bakker. Merkle hash torrent extension, August 2009. http://www.bittorrent.org/beps/bep_0030.html, accessed 01 August 2013.

[Ban12] European Central Bank. Virtual currency schemes, October 2012. https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf, accessed 01 August 2013.

[BBSU12] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. Bitter to better — how to make bitcoin a better currency. In Angelos D. Keromytis, editor, *Financial Cryptography and Data Security*, volume 7397 of *Lecture Notes in Computer Science*, pages 399–414. Springer Berlin Heidelberg, 2012.

[Bit] Bitcoin Repository on Github. Encrypt wallet – issue #3. https://github.com/bitcoin/bitcoin/issues/3#issuecomment-1764850, accessed 01 August 2013.

[blo] blockchain.info. Blockchain size. http://blockchain.info/charts/blocks-size, accessed 01 August 2013.

[coi] coinmarketcap.com. Crypto-currency market capitalizations. http://coinmarketcap.com/, accessed 01 August 2013.

[Dot] DotBIT Project. Namecoin dns. http://dot-bit.org/Main_Page, accessed 01 August 2013.

[GC08] Jingzhi Guo and Angelina Chow. Virtual money systems: a phenomenal analysis. In *Proceedings of the 10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, pages 267–272, 2008.

[HC12] Mike Hearn and Matt Corallo. Bitcoin improvement proposal 0037, October 2012. https://en.bitcoin.it/wiki/BIP_0037, accessed 01 August 2013.

[JO13] Ari Juels and Alina Oprea. New approaches to security and availability for cloud data. *Communications of the ACM*, 56(2):64–73, February 2013.

[Kin13] Sunny King. Primecoin: Cryptocurrency with prime number proof-of-work, July 2013. http://ppcoin.org/static/primecoin-paper.pdf, accessed 01 August 2013.

[KN12] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake, August 2012. http://ppcoin.org/static/ppcoin-paper.pdf, accessed 01 August 2013.

[Mer88] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *A Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology*, CRYPTO '87, pages 369–378. Springer Berlin Heidelberg, 1988.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. http://bitcoin.org/bitcoin.pdf, accessed 01 August 2013.

[Per09] Colin Percival. Stronger key derivation via sequential memory-hard functions, May 2009. https://www.tarsnap.com/scrypt/scrypt.pdf, accessed 01 August 2013.

[PPC] PPCoin GitHub Wiki. Faq – why do you need central checkpointing? https://github.com/ppcoin/ppcoin/wiki/FAQ#why-do-you-need-central-checkpointing, accessed 01 August 2013.

[Pri] Primecoin GitHub Wiki. Faq – what is the scientific value behind primecoin's work? https://github.com/primecoin/primecoin/wiki/FAQ#what-is-the-scientific-value-behind-primecoins-work, accessed 01 August 2013.

[Sch13] Andreas Schildbach. Bitcoin wallet 2.4.1 released, February 2013. https://bitcointalk.org/index.php?topic=146552.0, accessed 01 August 2013.

[The] The Apache Software Foundation. Antientropy – cassandra wiki. https://wiki.apache.org/cassandra/AntiEntropy, accessed 01 August 2013.

[Wik] Wikipedia. List of cryptocurrencies. https://en.wikipedia.org/wiki/List_of_cryptocurrencies, accessed 01 August 2013.

[WLL+13] Yuanfeng Wen, JongHyuk Lee, Ziyi Liu, Qingji Zheng, Weidong Shi, Shouhuai Xu, and Taeweon Suh. Multiprocessor architectural support for protecting virtual machine privacy in untrusted cloud environment. In *Proceedings of the ACM International Conference on Computing Frontiers*, CF '13, pages 25:1–25:10. ACM, May 2013.