THE BOOK OF SATOSHI

The Collected Writings of Bitcoin Creator

Satoshi Nakamoto

PHIL CHAMPAGNE

FOREWORD BY JEFF BERWICK

THE BOOK OF SATOSHI

THE BOOK OF SATOSHI

<<<< >>>>

The Collected Writings of Bitcoin Creator
Satoshi Nakamoto

PHIL CHAMPAGNE

E53 PUBLISHING LLC

Published in the United States of America by e53 Publishing LLC

ISBN 978-0-9960613-0-8 Hardcover
ISBN 978-0-9960613-1-5 Softcover

e53 Publishing LLC
e53publishing.com

Cover illustration by Lisa Weichel
Editing by Mary Graybeal
Cover and text design and composition by John Reinhardt Book Design

This book is also available in eBook format.
To get a free copy, please go to: BookOfSatoshi.com

Contents

About the Cover Picture

CREDIT FOR THE IMAGE on the front cover goes to Lisa Weichel (user id *lisa_aw on flickr.com*). The photo was taken at Cueva de las Manos (Cave of Hands) in the province of Santa Cruz in Argentina. Cueva de las Manos is a series of caves famous for the various paintings of human hands covering its walls. The paintings, the earliest of which date from around 13,000 years and the latest from about 9,000 years ago, were left there by multiple generations.

I selected it as this book's cover image because it seems to me to embody many of the concepts underlying Bitcoin—many individuals participating and cooperating to attain, over time, a common goal and yet maintaining their own individuality and uniqueness. Bitcoin differs from the cave paintings of Cueva de las Manos in scale, however.

Although these paintings were produced by multiple generations of individuals over several thousands of years, the number of these artists can't compare in size to the millions who now and will in the future use Bitcoin. Moreover, Bitcoin's users are geographically dispersed, collaborating over a decentralized system. Finally, whereas Cueva de las Manos was the work of one or more distinct tribes of humans, Bitcoin, open to anyone to use and adapt, transcends nationality and has the potential to become a true world currency.

Acknowledgments

I WOULD LIKE TO EXTEND my profound appreciation to the following individuals for their contribution to this work:

Dustin Trammell (dustintrammell.com) for sharing email exchanges he had with Satoshi Nakamoto,

Gavin Andersen, Lead Core Developer of the Bitcoin project, for his contribution to Bitcoin and also for sharing his email exchanges with Satoshi Nakamoto,

Jeff Berwick of DollarVigilante.com for writing the foreword and for being an advocate of freedom and liberty.

For their support, expertise, input, and contributions, I would like to thank my son, Samuel, my daughter Vivianne and my wife, Marie Gagnon. And finally, I would like to thank all the people who helped me put this book together in particular Mary Graybeal, our editor who did a tremendous job and John Reinhardt who came up with this great design for the book.

And, lastly, a sincere thanks to Satoshi Nakamoto. Without him, how long would we have had to wait before such a revolutionary concept as Bitcoin was discovered and shared?

Who This Book Is Intended For

THIS BOOK CONTAINS most of the writings of Satoshi Nakamoto, creator of Bitcoin, published in emails and forum posts during the span of a little over two years during which Bitcoin was launched and became established. Anyone interested in learning about Bitcoin and, more specifically, about the thought processes of its creator will appreciate this book. Its content will be an easy read for anyone having a background in computer software. However, economists and investors without a background in information technology may also be interested in Satoshi's writings, some of which concern economic concepts. Depending on background and interest, certain readers may be interested in only certain chapters.

To enable readers to derive maximum benefit from Satoshi's writings, we've included a chapter entitled "How and Why Bitcoin Works" that provides an introduction to the key concepts of Bitcoin and the fundamental principles on which it is based. This should help the reader gain sufficient understanding to comprehend the majority of the chapters which follow. Chapters are presented in chronological order, from the earliest post in which Satoshi presents the germinal idea of Bitcoin to the most recent, which marks his withdrawal from public life.

Part of this book's content comes from various Internet forums: *p2pfoundation.org*, *bitcointalk.org*, and the cryptography mail archive.

You can visit the website *TheBookOfSatoshi.com* for easy references to the URL web links referenced in the book. They are listed per chapter.

Foreword

BITCOIN HAS CHANGED EVERYTHING. Its importance as an evolution in money and banking cannot be overstated. Notice I don't use the word "revolution" here because I consider Bitcoin to be a complete "evolution" from the anachronistic money and banking systems that humanity has been using—and been forced by government dictate to use—for at least the last hundred years.

One of the biggest issues that newcomers to Bitcoin have is that it is "shrouded in mystery".

This is not totally true, as this important book shows. While the true identity of Satoshi Nakamoto may never be known for certain—despite those like Dorian Nakamoto, whom the mainstream media say is Satoshi—what we do know, in very prolific and historical detail, are the underpinnings and design of Bitcoin from its earliest days.

Very detailed conversations were held between top cryptographic and programming experts since the very first day Bitcoin was introduced… a day that may go down in history and possibly be celebrated by generations to come. November 1, 2008.

The first words posted by Satoshi Nakamoto were eloquent in their simplicity as he announced his creation, which would go on to change the world, "I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party."

He then put a link to a white paper he had written on the subject. The rest, as they say, is history.

These discussions, taking place publicly on the *bitcointalk.org* forum, went on until December 12th, 2010. After that, Satoshi went dark.

Amongst the Bitcoin community, these posts are well known, but your average person would need hours to scour through it all and make sense of it. In this important book, Phil Champagne has gone through each post and identified the most important ones… and given the context for the time of the post as to why it is important. This creates a logical timeline of Bitcoin's evolution straight from the keyboard of Satoshi Nakamoto and could be described as Bitcoin's autobiography.

As I write, in March 2014, Bitcoin's future is unknowable. It could go on to change the world dramatically, freeing us from the oppression of central banks and the gargantuan governments that feed off their free money. Or, it could go down in smoke and flames due to any number of possible events.

No matter what happens from here, however, the impact of Bitcoin ***is*** knowable. Its most core concept has and will change how we think about contracts, trust, and transactions no matter what happens to Bitcoin itself. Already thousands of applications have been built off the platform, and these have expanded it outside the world of financial transactions.

Phil Champagne has put into an easy-to-read format the fomenting of one of the most important technological innovations of our time… a completely decentralized platform to perform payment transaction without the need for a trusted third party. Its importance is only surpassed by the Internet itself as an evolution in communications. Chapter 2 provides readers unfamiliar with Bitcoin a great overview of its technological and philosophical foundation and of how it operates.

Decades from now many will look back at this innovation the way we currently look back at the Internet or the Gutenberg press as being epochal moments in the history of civilization. And this collection of Satoshi's posts and correspondences forms a logical timeline and will

be one of the easiest ways for future historians to understand just how it began and evolved.

Jeff Berwick,
Editor in chief, *The Dollar Vigilante*
http://DollarVigilante.com

1

Introduction

WE HAVE SEEN many amazing technological revolutions throughout human history. The Guttenberg press helped bring books to the masses. The telegraph has enabled crude but rapid communication across great distances. More recently, personal computers have vastly increased human productivity, leading to the creation of the Internet, digital communications, and the advent of citizen journalism as photos of major events are almost instantly uploaded to Twitter and other social networks via smartphone, which are small computers in their own right. Until fairly recently, however, the monetary system has remained somewhat untouched by a major breakthrough.

Bitcoin is run by software whose blueprint (source code) is freely available for anyone to see and even adapt for his or her own use. It currently runs on multiple computers connected over the Internet via a common networking protocol defined by this same software.

Existing within this software and existing because of it is a digital currency known as *bitcoin*, spelled with a lower case b and abbreviated BTC.

Bitcoin, both a virtual currency and a payment system, represents a revolutionary concept whose significance quickly becomes apparent with a first transaction. A buyer making a purchase in BTCs has only to provide the merchant with personal information relevant to the purchase, for example, the shipping or email address, to pay. Compare this with a credit card purchase, which necessitates the buyer giving enough personal information to enable another party bent on fraud, a hacker or dishonest employee, to make fraudulent purchases with it.

Bitcoin's significance is not limited to the simplicity of the payment system, however. The supply of Bitcoin currency is defined by the software and its underlying protocol. Only 21 million bitcoins will ever come into existence, with about 12 million so far having been created. The last bitcoin is expected to be created around the year 2140. This very specific, limited money supply has led to many controversies, some of which have more to do with lack of understanding of the protocol or the economics than with the software itself. Although 21 million BTC might seem insufficient with a global population of 7 billion people, the bitcoin currency is highly divisible. The smallest denomination allowed by the current software is 0.00000001 BTC (10^{-8} BTC), which has been defined as 1 *satoshi* and was named after the software's putative creator, Satoshi Nakamoto. There are therefore 100 million satoshis in a single bitcoin, and thus the maximum supply of 21 million BTC will be equal to 2.1 quadrillion satoshis or, if you prefer, 2,100 trillion satoshis.

Bitcoin was created by an anonymous person (or group of persons) known as Satoshi Nakamoto. At the time Nakamoto made his first public post announcing his paper on Bitcoin, he was just another anonymous user like millions of others posting on Internet forums. His new software was then still in the early phase of development, and

Bitcoin was only an experiment in its early stages. Satoshi's interaction was limited to email exchanges only and for a brief duration of a little over 2 years. Since then, we haven't heard from him. Around the time of his last post, Bitcoin's value was soaring, and the media were starting to take notice. Just when Bitcoin appeared poised to take off and was beginning to attract serious interest, Satoshi Nakamoto retreated from the public eye.

A few years later, Satoshi has become something of an iconic figure, and his retreat has only served to amplify the mystery surrounding him. His identity is irrelevant to the well-being of Bitcoin, as the code is open source and is, in fact, being constantly upgraded and improved upon even as we speak. However, gaining an understanding of the mindset of the mysterious person (or group of persons) behind this marvelous new technology would certainly prove interesting.

Satoshi's two-year "public life" overlapping Bitcoin's development and launch began with the publication of his paper "Bitcoin: A Peer-to-Peer Electronic Cash System", which he announced on November 1st, 2008, on the Cryptography Mailing List. At that time, this paper could be downloaded at domain name *bitcoin.org,* which had been registered a few months earlier on August 18th, 2008, through *anonymousspeech.com*. On November 9th, 2008, the Bitcoin project was registered on *SourceForge.net* and, at the beginning of 2009, the genesis block was created. To understand the genesis block, imagine a bookkeeping ledger that adds new pages (blocks) daily and contains a record of all bitcoin transactions ever made. The very first page of this book is called the genesis block, which will be explained in more detail in the following chapter. Satoshi incorporated this interesting quote into the genesis block in reference to the bank bailouts occurring at the time:

The Times 03/Jan/2009
Chancellor on brink of second bailout for banks

Bank bailouts were and still are extremely unwelcome occurrences, particular to libertarians, who caricaturized our political and economic environment with this quote: "Privatize the gains and socialize the losses".

Six days later, on January 9th, 2009, Nakamoto published the source code of Bitcoin version 0.01 on *SourceForge.net*. As of this writing (March 2014), Bitcoin v. 0.8.6 is the latest version.

Satoshi's last post was published on the *bitcointalk.org* forum on December 12th, 2010. His last known communication is a private email sent a few months later to Gavin Andresen, current Lead Core Developer of the Bitcoin project.

Below is a chart of the public trade data from *bitcoinmarket.com*, the first Bitcoin exchange, which is no longer in business. As can be seen, the value of one bitcoin went from 10 cents to a dollar in a very short time. At the time of Satoshi's last post on the forum, it was trading around 25 cents and was approaching 30 cents per bitcoin.

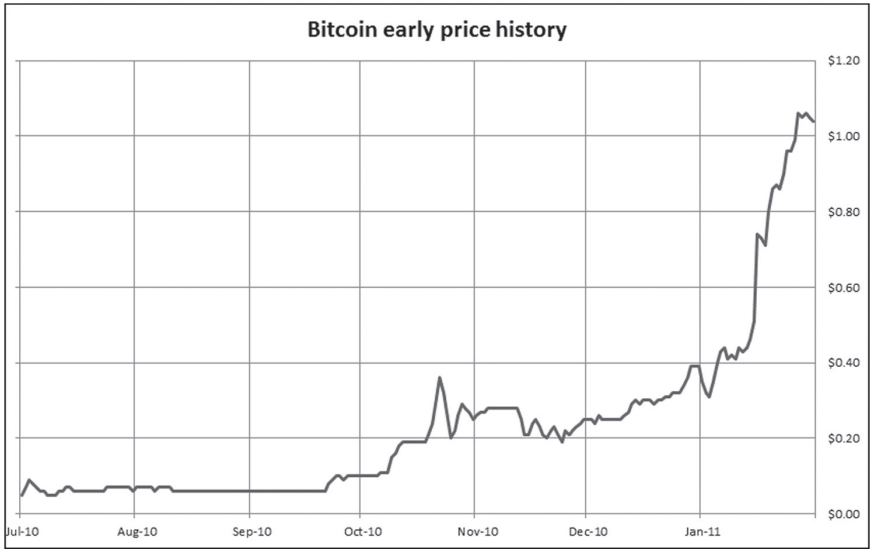


FIGURE 1 - EARLY CHART OF BITCOIN PRICED IN USD

This book is a collection of the postings and writings published under Satoshi's name on various forums and included in email exchanges. I have chosen to exclude posts of a technical nature, such as those related to coding, software compilation, and the detailed technical operation of the Bitcoin software. You will notice a few interesting subjects are discussed; one in particular involves the Byzantine Generals Problem, heretofore considered unsolvable, which describes the challenge of communicating in an unreliable environment. Some of Satoshi's comments relate to the news coverage that developed as Bitcoin started to attract media attention. One such event was when PayPal stopped processing payments for WikiLeaks, a journalistic non-profit organization dedicated to publishing selected secret and classified information provided by anonymous sources. A subsequent article published in *PC World* magazine conjectured how WikiLeaks could benefit from Bitcoin.

Satoshi's post seems to indicate that he was not comfortable with Bitcoin getting this kind of attention and was not ready for such a relationship, at least not yet:

> It would have been nice to get this attention in any other context. WikiLeaks has kicked the hornet's nest, and the swarm is headed towards us.

How much this event influenced his decision to "retire" from Bitcoin's development is unknown, but the timing is interesting, to say the least. Significantly, this post was written just nineteen hours before his last post on the forum, the announcement of the release of Bitcoin version 0.3.19.

Many journalists and researchers have tried to identify who could be the person behind Satoshi Nakamoto. So far, at least three attempts at identifying him have been made. Typical choices have been known scientists in the field of cryptography, none of whose real names are

Satoshi Nakamoto. All have been proven false, and all denied being Satoshi Nakamoto as well. However, very recently, a newspaper claimed to have identified a Californian, an engineer with actual name Dorian Satoshi Nakamoto, as the Bitcoin Satoshi Nakamoto. Dorian Nakamoto has denied this, and I tend to believe him. For one thing, Dorian Nakamoto does not demonstrate the proficiency in English that the Bitcoin Satoshi Nakamoto has shown through his writing. What is most relevant to this book concerning this episode is that it apparently caused Bitcoin's Satoshi Nakamoto to break his silence and post this message on the *p2pfoundation* forum on Friday March 7th, 2014:

I am not Dorian Nakamoto.

As you will see in the book, Satoshi's replies addressed many of the most commonly asked questions and criticisms regarding Bitcoin and are still pertinent. I suspect that, were he still involved in Bitcoin's development and were he to be interviewed, the writings contained in this book would reflect the type of answers Satoshi would give.

Whatever eventually happens to Bitcoin itself, that the software has opened the mind of the world to a new concept is indisputable. As an open source code, it allowed a myriad of other distributed digital currencies to enter the scene. While most of them do not represent any significant innovations—only varying the number of coins, the transaction confirmation speed (in Bitcoin termed *block creation*), or the computer encryption algorithm—a few new ones which incorporate significant new features or new concepts are being developed. One such is "Truthcoin", described as a trustless, decentralized, censorship-proof, incentive-compatible, scalable bitcoin prediction marketplace. Ethereum (see *ethereum.org*) is another digital currency that, according to its creator, will allow users to encode advanced transaction types, smart contracts, and decentralized applications into the block chain (Bitcoin's large public ledger which grows in size daily).

Innovative thinkers are seeking to use some of the concepts introduced by Bitcoin in a truly open voting system, where voters can confirm that their votes have been properly counted and can, at any time, view a complete vote count, thus ensuring transparency. Bitcoin has therefore clearly sparked a new technological revolution that capitalizes on the Internet, another innovation that changed the world.

I am quite open to suggestions and corrections with respect to this book and its contents. Also, if you have private email exchanges with Satoshi that you feel can be made public, I will be glad to consider them for inclusion. Please feel free to contact me at BookOfSatoshi@gmail.com.

2

How and Why Bitcoin works

BITCOIN HAS BEEN DESCRIBED as libertarian in nature, but not all libertarians and those in favor of a gold-backed currency appreciate it however, and some, in point of fact, actively despise it. In our experience, some fundamental concepts related to Bitcoin are not well understood by these. To fully understand Bitcoin, knowing how and, just as importantly, philosophically why it works is essential. How can a distributed system composed of several different groups and managed by several individuals at the same time maintain its integrity and avoid the condition termed "tragedy of the commons" by Garrett Hardin? In this economic condition, individuals, acting independently and rationally according to self-interest, behave contrary to the whole group's long-term best interests by depleting common resources. A typical example is where a group of farmers share a

common pasture for grazing their cattle. Overuse and depletion of the common resource, the pasture, can occur since it is in no one farmer's individual self-interest to conserve it by limiting his cattle's consumption of the pasture.

Let's begin with a discussion of how Bitcoin works. To appreciate and understand most of this book, some basic understanding of Bitcoin's key concepts is necessary. This chapter will provide that and will conclude with a perspective on why Bitcoin, as a payment system, has been proven so far to be a viable solution. To complete our discussion, we will elaborate Bitcoin's economic implications.

At its core, Bitcoin incorporates the following concepts:

- A public ledger (called Bitcoin's *block chain*). Consider this as essentially a giant book that is publicly available and contains the bookkeeping records of all transactions ever made in the Bitcoin system, with new pages constantly being added.
- A cryptographic algorithm called asymmetric encryption used for authorization of the transactions.
- A distributed network of computer *nodes* (also commonly known as *miners*) that verify and validate Bitcoin transactions and update the public ledger.

Let's explore these concepts in greater detail.

Bitcoin's block chain: public bookkeeping

All members of the Bitcoin network share its public ledger, the *block chain*. Imagine a giant accounting book with each page listing a series of transactions. A new page containing the latest Bitcoin transactions sent by payers across the world is added approximately every 10 minutes. This giant book is constantly available on the Internet to anyone

who runs the Bitcoin software. Note that software programs called Bitcoin *wallets* can run on smartphones or personal computers and allow a user to make payments over the Bitcoin network.

In the context of Bitcoin, the pages forming the ledger are called *blocks* because they represent "blocks" of data. The block chain, composed of many individual blocks, grows constantly in length and contains all transactions performed in Bitcoin since its launch in January 2009.

A Bitcoin transaction request contains the following:

1. The Bitcoin address of the payer, which contains the source of funds for the payment,
2. The recipient's (payee's) Bitcoin address, and
3. The amount of bitcoins being transferred.

Since the block chain contains the history of all outgoing and incoming payments associated with the payer's Bitcoin address, miners, who also manage the Bitcoin network, can validate that the payer has sufficient funds to cover the payment. At any time, anyone can view the amount of bitcoins linked to (or, in an abstract way, held in) any specific Bitcoin address. See for yourself. Go to *blockchain.info* and enter the following address.

1GaMmGRxKCNuyymancjmAcu3mvUnVjTVmh

Under "Search", the number of bitcoins associated with this address will be returned.

Although the owner's identity cannot be known from his Bitcoin address without his having provided this information, any transfers in and out of his account, as well as his current balance, are publicly available for viewing.

Asymmetric encryption: who gets to spend those bitcoins

Encryption keys are associated with a transaction such as the one described above. Bitcoin employs a system of asymmetric encryption (also known as public-key cryptography), so called because the encryption algorithm requires a pair of keys, each consisting of a long series of digits. One is public and controls the decryption operation, while the other, the private key, governs the encryption operation, or vice versa.

It is easy for the algorithm to create a private key and to derive its corresponding public key. However, determining a private key from the corresponding public key is computationally unfeasible, thus allowing the public key to, as its name implies, be made public. With the public key, the payee can retrieve the transaction information, allowing the

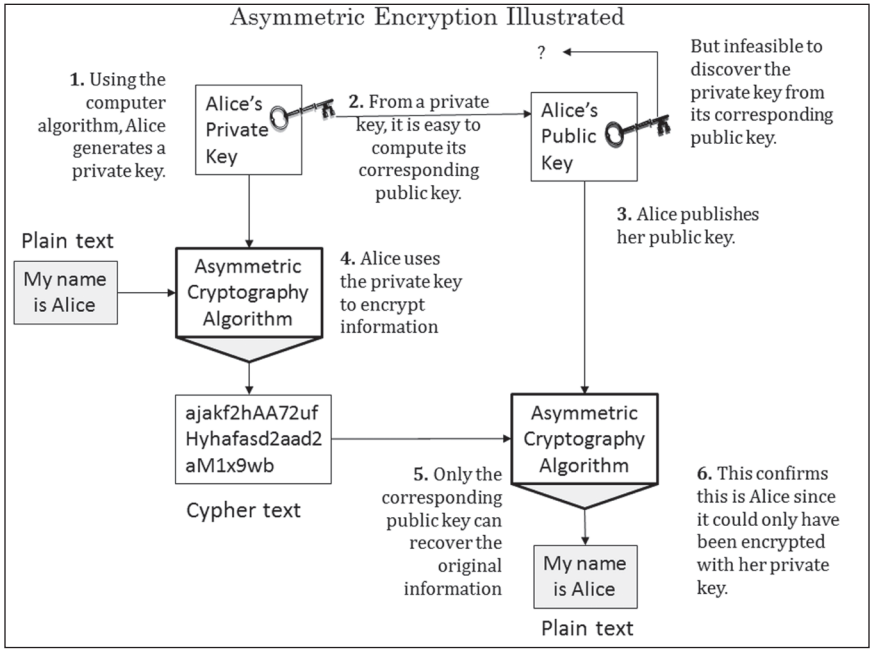


FIGURE 2: ASYMMETRIC ENCRYPTION ILLUSTRATED

transfer of bitcoins to proceed. The following Figure 2 conceptually illustrates Bitcoin's double key system, which provides part of the basis for Bitcoin's operation.

The Bitcoin software's algorithm allows only the owner of the private key to "spend" bitcoins associated with that Bitcoin address. The recipient, or payee, shares his Bitcoin address with the payer. Since only the recipient knows the private key linked to his address, only he will be able to access, spend, or transfer those bitcoins at a later time.

Within Bitcoin, a sender digitally signs a Bitcoin transaction with his private key. Bitcoin transactions actually contain the public key (assume this is the Bitcoin address for now). Using this public key, the system verifies that the digital signature is valid and thereby confirms that the sender is indeed the private key's owner. This system allows the owner to "spend" the bitcoins associated with his Bitcoin address in the public ledger, and the public ledger (i.e., the block chain) will be updated with a new page (i.e., block) containing this transaction. The addition of this new transaction to the block chain effectively tells the Bitcoin network to credit those bitcoins to the recipient's address and debit them from the sender's Bitcoin address. Private keys are made of a long series of digits stored and managed by password-protected Bitcoin *wallets* (i.e., software on the user's computer, mobile device, or other web application).

A network of miners acting as minters, bookkeepers, and regulators of the system

So far, we have talked about what transactions look like and how they are validated. If Bitcoin were a centrally operated system, the story would end here: A single entity would be responsible for this task. However, Bitcoin is a decentralized system, and, as such, this task is shared among a collection of voluntarily participating nodes

(miners) distributed across the world. Understanding how a system that includes bookkeeping and payment transfer authorization could be operated by different entities in such a way as to support his or her own self-interest is essential. This characteristic of the system is one of the key understandings to which I alluded earlier as one that is often missed by critics of Bitcoin.

Miners, the nodes responsible for operating the Bitcoin network, verify that transactions are valid and update the block chain with new blocks consisting of the latest transactions on a regular basis. The Bitcoin software run by miners on their individual computers incorporates the Bitcoin protocol with its set of rules and agreements.

Overall, the Bitcoin network requires that the block chain (public book ledger) be continually updated with the addition of new blocks (pages in the ledger book). Approximately every 10 minutes, a new block is added with the list of the latest transactions. Although all miners are working on the next block, only one will be selected to have his specific version of the block added to the block chain. Indeed, each miner is operating in his self-interest when he creates his own version of this next block and so personally collects the transaction fees associated with that block of transactions. Although the core parameters of Bitcoin transactions are unaltered (payer, payee, amount), most of them include transaction fees, disbursed by the payer and to be credited to the account of the miner whose block is selected for inclusion in the block chain. This miner will therefore update each of these transactions and will credit the fees associated with those transactions to his very own Bitcoin address.

In addition to transaction fees, miners whose blocks are added to the block chain also earn additional credits with newly minted bitcoins. They create an extra transaction that adds these to their own bitcoin accounts. This is called a block reward. Currently, Bitcoin's protocol allows miners to allocate themselves 25 new bitcoins per block created. This is in addition to the sum of transaction fees. Initially, at

Bitcoin's launch, 50 bitcoins (BTC) were allocated as the block reward per block, which is halved approximately every four years.

With the new bitcoins credited to his address, the miner whose version of the block is selected for inclusion in the block chain clearly benefits from finding a solution before his fellow miners do. How this selection process works will be explained shortly. For now, however, view it as solving a mathematical problem by executing a very expensive computing task. The solution is difficult to find but, once found, its correctness is easy to verify. The first miner to find the solution to his block is allowed to publish this version to the entire network of miners.

These miners receive the block and its solution and then work to authenticate and validate it, that is, certify that the solution found by the first miner to the block is correct. The Bitcoin protocol sets the difficulty of the problem in such a way that an average of around 10 minutes are required for the solution to be found.

If the miner solving the block were to credit himself with more than the 25 new bitcoins currently allowed, the other miners would reject that miner's block and would continue working on finding the solution for their own versions of it. Each block is slightly different and therefore each has a different solution.

In what might seem counterintuitive, when a miner solves the computing task, all other miners accept defeat, agree to include this miner's block as the next block in the block chain provided it is able to be validated, and begin work on the next block. This work involves each miner's adding all the most recent transactions that have come in since the creation of the previous block to a new block, which will in its turn be solved and added to the never-ending block chain.

The manner in which Bitcoin operates explains why the miner who was first to arrive at a solution will credit himself with only the amount of block rewards allowed by the Bitcoin protocol. Doing so ensures acceptance of his block by the other miners and receipt of its

associated rewards (i.e., transaction fees). Equivalently, the other miners achieve no gains by rejecting the block even though it is valid. The Bitcoin payment system will hold its value only when it is functioning properly. If miners were to reject all blocks but their very own, no consensus would ever be reached, the value of the overall system would be destroyed, and none of the miners would be able to benefit. In such a case, whatever amounts of bitcoins the miners hold would then become worthless. Therefore, all miners benefit if all respect the Bitcoin protocol established within the shared Bitcoin software. Thus, Bitcoin embodies the inverse of the tragedy of the commons described earlier.

Now let's delve into the details of what we earlier described as the expensive computing task required to solve the mathematical problem of a block. For a miner to have his block selected, he must have solved a problem associated with the block. This selection process is called "proof-of-work" as it implies the miner had to work for it. To fully understand the mechanism involved, we need to first understand a cryptographic concept known as a *hash function*. Then, we can explain how it is used in the context of a miner's proof of work.

Cryptographic hash function— a digital "fingerprint"

Cryptographic hash is a complex algorithm that performs a very basic task–transforming text of arbitrary length (an entire book, a document, a sentence, or even a single word) into a fixed-length string of numbers that appears random. The following Figure 3 provides some examples. The output of a hash function, or simply hash, is usually called the message digest and can be considered the document's "fingerprint".

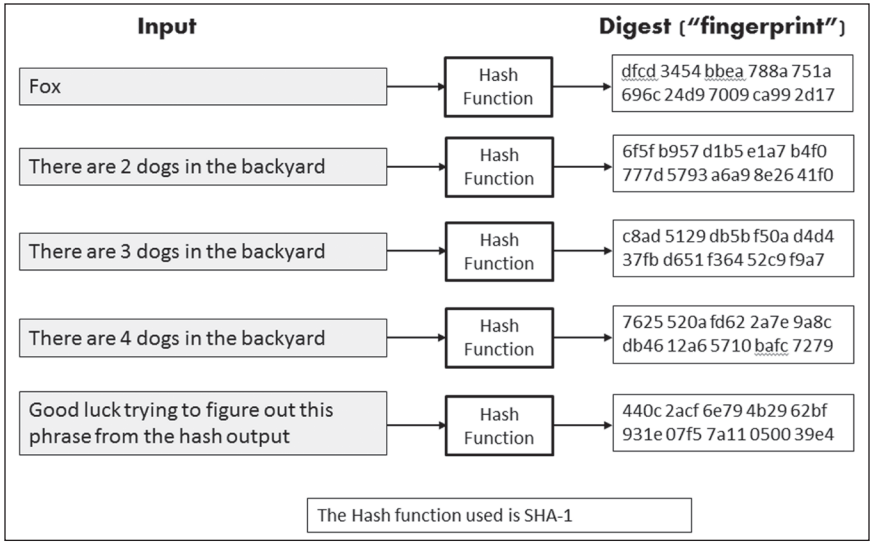


FIGURE 3: THE HASH ALGORITHM IN ACTION

In the figure above, note that the input "There are 2 dogs in the backyard" leads to a completely different digest than "There are 3 dogs in the backyard". Simply changing one character leads to an output with all digits completely different. The digest outputs in this figure are expressed as hexadecimal numbers. Unlike the decimal system we commonly use, the hexadecimal system has a base of 16. It employs sixteen symbols to represent the sixteen numbers in the system. Symbols 0 through 9 represent the numbers 0 through 9, and letters A through F represent the numbers 10 through 15. Thus, hexadecimal F represents the number 15. The hexadecimal number 5A36 is therefore equal to $(5 \times 16^3) + (10 \times 16^2) + (3 \times 16^1) + (6 \times 16^0)$, which equals, in the decimal numbering system, to 23,094. Experiment with switching from Hex to Dec on your own computer's calculator to see how it works.

A Bitcoin user has no control over what the output (the digest in Figure 3) will look like. Also, given a specific digest output, finding an input that would generate it is nearly impossible. Thus, generating a digest is easy, but deriving the original text from the digest is

impossible. Employing the analogy of the human fingerprint, given a single fingerprint, we would find it impossible to identify the person who left it unless that person had been fingerprinted beforehand.

Earlier we mentioned that all miners can easily verify that a solution is correct once it has been found but that finding it is the difficult part. That's why cryptographic hash is ideal for Bitcoin's purpose. Miners, in their attempts to solve a block, must reproduce a specific pattern displayed by the contents of the digest. Since reproducing a specific output within the digest is impossible, they must increment a digit in the text and recalculate the hash again and again until they stumble upon the specific pattern in the digest that is required by the Bitcoin protocol. This process is analogous to varying the number of dogs ("2 dogs", "3 dogs", "4 dogs") in the example in Figure 3 to create different digests. For instance, say that the current Bitcoin protocol specified that the contents of the digest display a pattern beginning with "00". By varying the number of dogs in the example, the corresponding hexadecimal number in the digest will eventually satisfy this requirement, indicating a solution to the block.

Miners looking for the solution must usually calculate the hash millions of times to find the right pattern, but only a single hash calculation by other miners is necessary to validate it once it is found.

Bitcoin's hash algorithm, which creates the contents of the digest from the input text, makes the system described above possible. Thus, an ideal cryptographic hash function has four main properties[1]:

- *Computing the hash value corresponding to any given message is simple.*
- *Generating a message that has a given hash is impossible.*
- *Modifying a message without changing the hash is impossible.*
- *Finding two different messages having the same hash is impossible.*

[1] http://en.wikipedia.org/wiki/Cryptographic_hash_function

The following example, taken from Wikipedia, illustrates the hash function in use.

> *Alice poses a tough math problem to Bob and claims she has solved it. Bob would like to try it himself, but would also like to ensure that Alice is not bluffing. Therefore, Alice writes down her solution, computes its hash and tells Bob the hash value (whilst keeping the solution secret). Then, when Bob comes up with the solution himself a few days later, Alice can prove that she had the solution earlier by revealing it and having Bob hash it and check that it matches the hash value given to him before. (This is an example of a simple commitment scheme; in actual practice, Alice and Bob will be computer programs, and the secret would be something less easily spoofed than a claimed puzzle solution).*

Hash functions form part of the process enabling users to digitally sign a document or text in Bitcoin. In the context of Bitcoin's proof-of-work, which will be discussed below, the two most useful characteristics of the hash functions are the following:

- The impossibility of generating a message from a given hash
- Generating an entirely new hash by changing only one character in the message

Several types of hash algorithm have been created, and Bitcoin uses two of them: SHA-256 for the proof-of-work and RIPEMD-160 for the Bitcoin address. The hash function is at the heart of the proof-of-work, which we'll discuss next.

Miner's Proof of work

At any given time, each miner is actively engaged in creating the next block to be added to the block chain by resolving a difficult problem, which is called a *proof-of-work*. The first miner to solve the proof-of-work is rewarded with freshly minted bitcoins (25 bitcoins as of this writing) and with the cumulative transaction fees associated with the transactions included in the block being created. Transaction fees, typically a nominal amount, are added by payers when they send their transactions. By around the year 2140, all bitcoins will be mined, and miners will be rewarded solely with transaction fees.

The proof-of-work can thus be thought of as a race between bitcoin miners to discover the SHA-256 hash of the block they are trying to create that will have a certain characteristic. As we saw earlier, the hash output is simply a very large number expressed in hexadecimal. The miner's goal, the problem that must be solved, is to generate a hash output that is below a certain value. The first miner to compute a value having this characteristic wins, and his version of the block will, after validation by the other miners, be added to the block chain discussed earlier in this chapter.

For simplicity, imagine that the hash output was actually a number between 0 and 1,000,000 and that the first miner to get a hash output of less than 10,000 wins. The 10,000 acts as a threshold, and each block within Bitcoin contains a number whose sole purpose is to obtain the threshold.

The number within the Bitcoin block that is tested against the threshold value is known as the "nonce". Each miner increments its nonce by a certain amount until the hash output for its block is below the threshold. As we said earlier, each miner's block has different information and therefore a different hash output for the same "nonce". This process is illustrated in Figure 4.

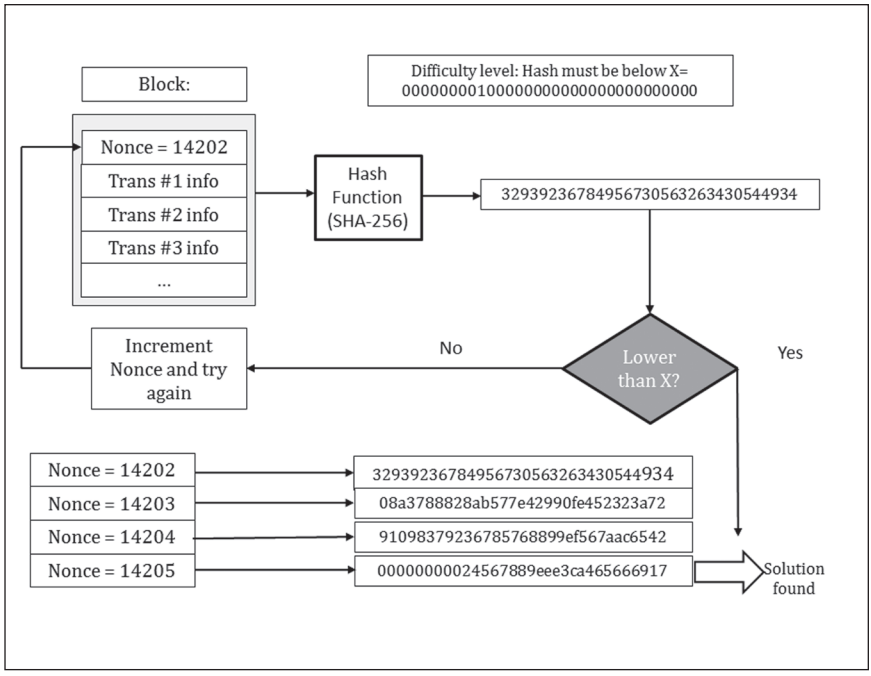


FIGURE 4: PROOF-OF-WORK ILLUSTRATED

The Bitcoin protocol, operated by the software running on each miner's computer, adjusts the difficulty level of the problem so as to take around 10 minutes before the first miner solves it. The purpose is to have the block chain updated on a regular basis with a new block containing the latest transactions sent during the prior 10 minutes. This value is somewhat arbitrary and, as will be seen in later chapters, Satoshi devoted some of his discussions to this topic.

The previous discussion compared the nonce to a threshold. Because the hash's numbers, termed the proof-of-work, are in a hexadecimal, or base 16, numbering system, this translates to the first X number of bytes being the digit 0, where X is adjusted periodically to keep the difficulty level of the proof-of-work fairly constant.

For example, assume that block #282,435 of the block chain has the following SHA-256 output:

0000000000000000c6647dad26b01b28f534223450d75d3b6b2882855039b673

Recall that in the base 16 number system, there are symbols representing the sixteen numbers 0 through 15; the symbols representing 0 through 9 in this system are 0 through 9 as in the decimal, or base 10, system, and numbers 10 through 15 of the hex system are represented by A through F. The hexadecimal number above is comprised of 64 digits. Since the terms to the left in a hexadecimal number represent higher powers of 16 hence larger numbers, to make the hash output smaller, the leading digits within the hash output must be 0. This is why stating that the hash output requiring to be below a certain threshold translate to have a certain number of leading digits be 0. Viewed in either way, proof-of-work is finding a nonce that will generate a hash output below the threshold established by the Bitcoin protocol at the time.

In the example in Figure 4—Proof-of-work illustrated, only with the first sixteen digits of the output equaling 0 could the hash output fall below the threshold set by Bitcoin's protocol. Therefore, the miner who obtained this number first and so "won" that block had to keep changing the "nonce" number until a hexadecimal number having at least the desired number of leading 0s was generated. As in a lottery, the miners buying the most "tickets" (i.e., generating the most numbers of SHA-256 output) have a better chance of finding a number having the correct number of 0s. This requirement of the Bitcoin system has led to a race to create hardware capable of generating more hash per second. The lucky miner who first discovered the hash for block #282,435 of the block chain incremented the nonce to 505,482,605 stated in decimal, meaning this miner had to generate over 500 million "hash" before finding one with the correct number of leading zeroes.

As stated previously, the Bitcoin protocol's goal is to have a block of transactions created approximately every 10 minutes. For a given level of difficulty, if more miners join—or more precisely, as more hash are

calculated per second—the chances of discovering the required digest (hash output) in less than 10 minutes increases. After a certain number of blocks, the Bitcoin protocol evaluates how fast blocks are being generated; if sooner than 10 minutes on average, the level of difficulty is increased (i.e., the number of leading 0s increases, decreasing the probability of any single miner's obtaining a digest having that characteristic); if longer, the difficulty is decreased (i.e., the number of leading 0s decreases, increasing the probability of obtaining it).

Once a miner discovers a nonce providing the correct hash output, the block is broadcasted, and other miners verify it, accept it, and begin work on the next block. Thus, Bitcoin operates like an ongoing lottery game restarting every 10 minutes. Who will be the lucky miner to find a nonce with the correct characteristics?

Figure 5 illustrate the concept behind the proof-of-work. Note that there is more information in the blocks than shown; it has been reduced for simplicity.

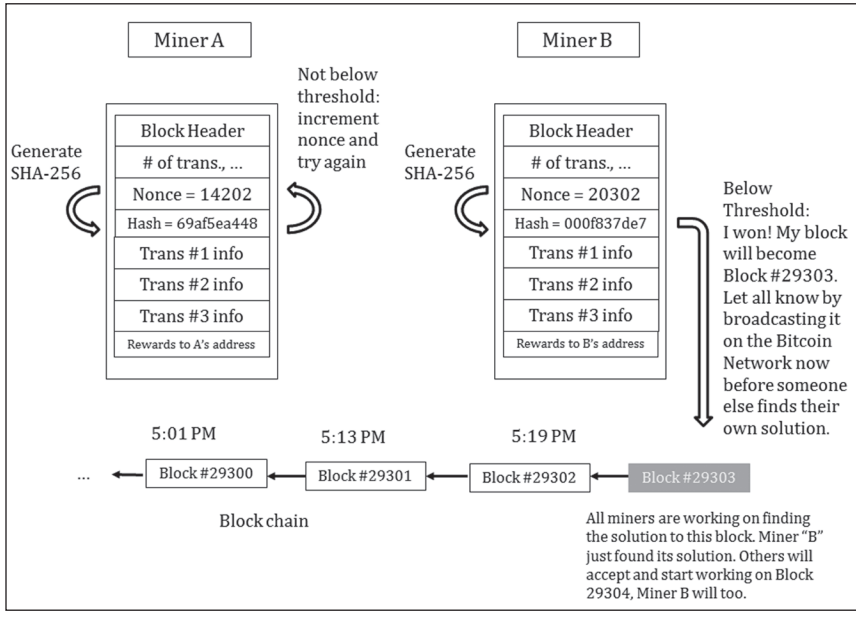


FIGURE 5: WINNER OF PROOF-OF-WORK

Miners' consensus & orphan blocks

As stated earlier, Bitcoin relies heavily on consensus in order to function. This concept, which will be discussed further in Chapter 9, comes into play when two miners solve their blocks at about the same time. When this occurs, the two miners both broadcast their blocks including solutions across the Bitcoin system. All other miners receive and retain both but their work on their next block will be based upon which of the two current blocks they receive first. Say 50% of the miners receive the block from Miner A first and the others receive Miner B's block first. This situation is illustrated for block #29302 in Figure 6 below.

This situation is analogous to a race going into overtime. Which of the two blocks becomes part of the true block chain will depend upon how quickly the next block is solved and by whom, a miner who received A's block or one who received B's block. At this point, two versions of the block chain exist, with half the miners having miner A's version of block #29302 and the other having miner B's version. Which of these two versions will survive depends on which version

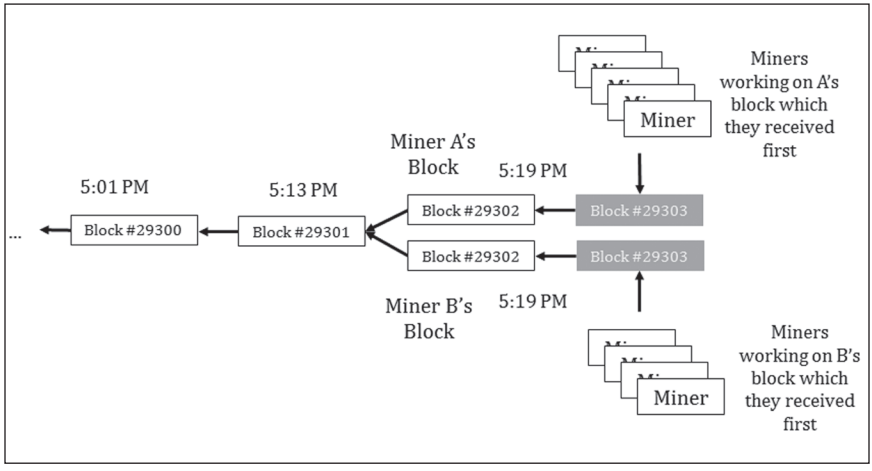


FIGURE 6: A BLOCK SPLIT

the miner solving the next block, #29303 in Figure 6, has on his computer. When block #29303 is solved, this version of the block chain becomes the longest of the two and hence the official one. All miners then drop the other version of the block chain, which becomes what is known as an *orphan block*. This process is illustrated in Figure 7.

Why does Bitcoin work?

So far we've covered how Bitcoin works, but not why. To understand

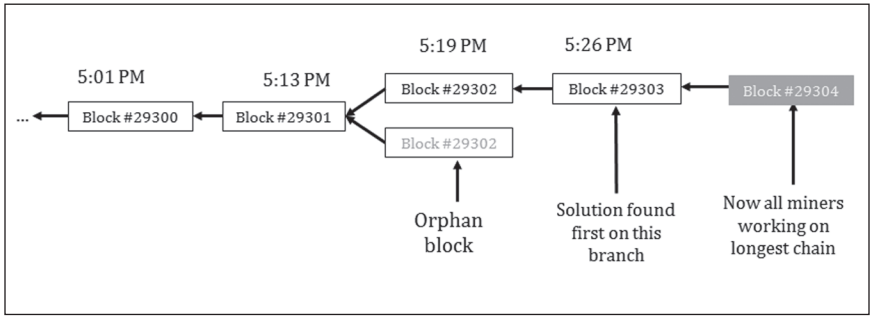


FIGURE 7: THE LONGEST CHAIN WINS

this, knowledge of a few additional concepts, open source software for instance, is necessary. These concepts are as follows and are explained below:

- Bitcoin is *open source software*.
- Bitcoin software establishes *the operating directives* the miners and wallet clients must follow.
- Bitcoin software also defines and operates *a communication protocol*.
- *Distributed file sharing* of the block chain allows for open bookkeeping.

Open source software is computer software whose source code is available for anyone to see. Moreover, it operates under a special license that allows anyone to modify and to use it. With the source code, a programmer can recreate the program (the binary file that runs on computers) and modify it at will. Thus have sprung up many imitators of Bitcoin, other virtual currencies differing from it only cosmetically and, for the most part, incorporating no significant innovations, with the exception of a very few like Namecoin. The majority of these alternative virtual currencies are based on changing the rate at which blocks are created, the total number of coins in circulation, and the cryptographic hash algorithm used.

A software's code being open source allows an expert to analyze it and to validate its integrity, that is, confirm that it does what it purports to do. A prominent example of open source software is Linux, which has displaced Microsoft Windows in market share in the server industry. Because it is open source, problems are found and fixed much more rapidly than if it were proprietary since multiple programmers are continually examining and improving the code. Linux has so far demonstrated that the greater good and self-interest can work in concert, at least with respect to managing open source software. This openness ensures a high level of integrity not achievable in proprietary software, where only the reputation of the company responsible for the software guarantees that it does what it is supposed to do.

Bitcoin also operates over the Internet using a defined protocol of operations that miners and wallet clients must follow. Wallet clients—software programs that are apps on smartphones or programs on personal computers—are what is used when someone is sending a payment transaction, which miners then validate prior to their being incorporated in the block chain. A single miner deviating from the protocol would have his operation rejected by the rest of the miners and would not be allowed to contribute to the operation of the network.

One typical argument raised against Bitcoin concerns the limit on the maximum number of bitcoins that will ever be created, which Satoshi Nakamoto set at 21 million. Once reached, what could prevent someone from increasing this limit? Nothing really, but he would need the cooperation of the majority of miners for this change to be accepted. Even were the majority of miners to agree to lift this restriction, if all did not agree, then a split in the block chain would result. Those in favor of lifting the restriction would use one version of the block chain while those not in favor would use a different version. In effect, we would have two virtual currencies rather than one, the "original Bitcoin" and a "Quantitative Easing Bitcoin". Over the long term, one would hold its value longer and better and would therefore become the preferred version while the other would drop in value. What would be your guess as to which one would hold its value longer and retain the interest of users of Bitcoin? Personally, I have a very good idea which one.

The Bitcoin development community is very conservative with regard to changes, and, at least so far, the preferred means of instituting major change has been the creation of new virtual currencies, some of which have no limits as to number of coins.

A final characteristic underpinning Bitcoin is that, not only is the software open source, but so is its bookkeeping. Some have termed the block chain "triple-entry bookkeeping" as it revolutionizes accounting. Anyone can inspect the block chain and verify that the accounting does follow the current established requirements and specifications of the Bitcoin protocol. The distributed file sharing of the block chain means that anyone running the Bitcoin software is connected to the Bitcoin network and has access to the block chain.

To gain a greater understanding of the brilliance of the conceptual basis of Bitcoin, I highly recommend reading Satoshi Nakamoto's white paper. The information I have provided here should make the paper more accessible. A reproduction of this paper is included at the end of this book.

http://bitcoin.org/bitcoin.pdf

We hope this chapter has helped you understand the core concepts. You should now be capable of reading the Bitcoin paper and the remainder of this book with considerably more ease.

Implications of Bitcoin

Bitcoin's impact as a monetary system is tremendous. One advantage is the ability it gives people to "wire" currency across the planet as simply as sending an email. This is particularly advantageous to immigrant workers who wish to send money to their relatives in their countries of origin. In contrast, companies that wire money across borders charge high fees to do so. There are fees associated with converting from national currencies to BTC and back again, but these conversion fees are small in comparison to wiring costs.

Another benefit touched on earlier regards online shopping and online donations. I'm confident that the current system of paying with credit cards will be completely changed in the future. Credit card payments require giving extensive information about the payer, including billing address and the 3-digit code on the back of credit cards. In essence, this is the Bitcoin equivalent of giving your private encryption keys to the merchant. The high number of frauds resulting from this security weakness has manifested itself in the form of high fees and chargeback with which merchants have to cope. Credit card companies spend a huge amount of cash every year in dealing with fraudulent charges. These costs are transferred to merchants, who, in turn, transfer them to consumers via higher prices for goods and services.

Another major impact of Bitcoin is on the monetary front, specifically in the system's ability to be money and not just a currency. A currency has the following properties:

- Is a medium of exchange (used as an intermediary in trade)
- Is a unit of account (can be counted, is quantifiable)
- Is durable (long duration)
- Is divisible (so to have smaller units)
- Is portable (so as to be easily transportable)
- Is fungible (mutually interchangeable, 1 unit of a specific value can replace another identical unit)

Money has all the properties listed above and, in addition, one other:

- The ability to preserve its value over the long term.

Unlike money, a currency is subject to inflation. In the early 1900s, inflation was defined simply as the action of inflating something, as in the case of a currency, by printing more of it. Today's dictionary defines it as a general increase in prices. However, rising prices are a symptom of a devaluating currency, which occurs when more of it is present than there was before. It is interesting but not surprising that this transition in definition corresponds to a time over which paper currencies became further and further detached from gold and silver, a development which leads to higher prices. Our ancestors saw, for instance, food prices remain virtually unchanged throughout their lifetimes. However, today's population has been conditioned to view rising prices as an immutable fact of life, like gravity. It is as if, in a place where it rains all the time, nobody has made the connection between clouds and rain. But who could blame them since they have never seen a blue sky? In the same manner, most people today do not perceive rising food prices as caused by currency inflation, with sometimes a lag of several years for the rising prices to manifest themselves. This was the case with the currency inflation of the 1960s only manifesting itself in the following decade, the 1970s.

To maintain its purchasing power over the long term (i.e., to not be subject to inflation), the money supply must be limited. Gold and silver have been the money of choice for thousands of years. Their supply on this planet is limited and requires anyone who intends to acquire more of it to trade energy and time for them through mining. You could say that the effort expended in mining a precious metal is analogous to proof-of-work in the Bitcoin system. Contrast this real work with simply printing more dollar bills. Paper currencies were initially adopted to act only as a convenient substitute (derivative) for precious metals, thus facilitating transactions. Paper currencies, being easily reproducible, have always been subject to inflation, as goldsmiths – and later bankers – used fractional reserve banking to lend more (i.e., print more paper currency) than they actually had gold in storage. This has led to the frequent "bank run" crises littering the history books.

Before the advent of computers and networking, transactions were limited to precious metals and paper currencies. Since then, electronic communications have introduced a new way of performing transactions of which gold and silver could never be directly a part. Until now, only centrally controlled and electronically transmittable currencies existed, allowing the controllers free rein in deciding the size of the underlying currency's supply. President Nixon demonstrated this clearly when he removed the dollar's convertibility into gold on foreign exchange markets. The Vietnam War and Lyndon Johnson's "great society" were funded by diluting the US dollar via the electronic printing press. It took time to manifest itself via the rising prices of commodities, but once it did, the price of gold in dollars has effectively been higher than the fixed $35 per ounce of gold that prevailed before the dollar was unlinked from the gold standard. It then became a free-floating, constantly inflating currency, like any other national currency in existence today.

As we discuss in Chapter 7, paper currencies (fiat) allow governments to fund deficit spending by stealing from the value of the

currency in circulation. The poor and, to some extent, the middle class are the most affected by currency inflation while the rich use debt and various financial derivatives to acquire companies and income-producing commercial real estate. They know the debt will be devalued along with the currency, providing an artificially obtained additional gain. The first way to address the "war on poverty" is to get rid of currency inflation and return to a form of money whose value holds over the long term. But do not expect government to propose or even entertain a proposal involving this course of action.

Currently, many magazine and newspaper articles on Bitcoin present its "deflationary" nature as its main negative. By deflation, they mean that prices measured in BTC will decline. In reality, this is Bitcoin's primary benefit. They report that people will be "hoarding" bitcoins rather than spending them in the economy. First of all, imagine that tomorrow bitcoins were to become the currency of choice for your country. Being human, you would still have to eat and to provide for a shelter; hence you would have to make these two expenses. What the comments in these articles demonstrate is a misconception about what money is. By saving rather than spending—"hoarding" is merely a pejorative term for saving—people are delaying consumption to a later time. We have seen this type of behavior exhibited recently by some so-called "bitcoin millionaires", who, at some point, become comfortable enough to spend some of their bitcoins on luxury items. In an economic system based on money—currency that holds its value over the long term—savers are not competing for resources with manufacturers, builders, factories, and those extracting commodities (i.e., marketable items) by deferring spending. By resources, we mean any form of energy, commodities, time, and labor, particularly specialized labor. Imagine the case of a person who decides to save by staying home rather than hooking up his trailer and traveling cross-country for vacation. By not traveling, he allows the gasoline he would have expended in traveling to be used by, for instance, a manufacturer to

transport material for building a new plant. Printing dollars do not create more barrels of oil, more gigawatts of electricity, or more hours in a day. I've illustrated this concept with rather simple examples, but I hope you can see that a currency like Bitcoin, with the ability to hold its value derived from its limited supply, has major ramifications.

In this chapter, we've covered the technology behind Bitcoin, the software concept underlying it, and we've touched on an alternative view of economics to which Satoshi Nakamoto himself likely adhered. Now that you have a good understanding of what Bitcoin is all about and how it works, turn the page and meet Bitcoin's creator, Satoshi Nakamoto!

3

The First Post on Crypto Mailing List

THIS IS SATOSHI NAKAMOTO'S announcement of Bitcoin. It was posted on the Cryptography Mailing list, a forum for those interested in anything related to cryptography.

Bitcoin P2P e-cash paper

Satoshi Nakamoto Sat, 01 Nov 2008 16:16:33 -0700

I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

The paper is available at:

http://www.bitcoin.org/bitcoin.pdf

The main properties:

Double-spending is prevented with a peer-to-peer network.

No mint or other trusted parties.

Participants can be anonymous.

New coins are made from Hashcash style proof-of-work.

The proof-of-work for new coin generation also powers the network to prevent double-spending.

Bitcoin: A Peer-to-Peer Electronic Cash System

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without the burdens of going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as honest nodes control the most CPU power on the network, they can generate the longest chain and outpace any attackers. The network itself requires minimal structure. Messages are broadcasted on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

Full paper at: http://www.bitcoin.org/bitcoin.pdf

Satoshi Nakamoto

The Cryptography Mailing List

4

Scalability Concerns

HERE, SATOSHI REPLIES to a comment concerning scalability. To make a payment, a client's wallet needs to have the full block chain, and with a growing block chain, it would put a memory burden on those small client wallet. This issue was addressed by Satoshi in a later release. Today, a smartphone app can easily handle transactions by connecting to a server it trusts that has the full block chain.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Sun, 02 Nov 2008 17:56:27 -0800

> James A Donald wrote:
>
> > Satoshi Nakamoto wrote:

> I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.
>
> The paper is available at:
> http://www.bitcoin.org/bitcoin.pdf

> We very, very much need such a system, but the way I understand your proposal, it does not seem to scale to the required size.
>
> For transferable proof of work tokens to have value, they must have monetary value. To have monetary value, they must be transferred within a very large network - for example a file trading network akin to bittorrent.
>
> To detect and reject a double spending event in a timely manner, one must have most past transactions of the coins in the transaction, which, naively implemented, requires each peer to have most past transactions, or most past transactions that occurred recently. If hundreds of millions of people are doing transactions, that is a lot of bandwidth - each must know all, or a substantial part thereof.

Long before the network gets anywhere near as large as that, it would be safe for users to use Simplified Payment Verification (section 8) to check for double spending, which only requires having the chain of block headers, or about 12KB per day. Only people trying to create new coins would need to run network nodes. At first, most users would run network nodes, but as the network grows beyond a certain point, it would be left more and more to specialists with server farms of specialized hardware. A server farm would only need to have one node on the network and the rest of the LAN connects with that one node.

The bandwidth might not be as prohibitive as you think. A typical transaction would be about 400 bytes (ECC is nicely compact). Each transaction has to be broadcast twice, so lets say 1KB per transaction. Visa processed 37 billion transactions in FY2008,

or an average of 100 million transactions per day. That many transactions would take 100GB of bandwidth, or the size of 12 DVD or 2 HD quality movies, or about $18 worth of bandwidth at current prices.

If the network were to get that big, it would take several years, and by then, sending 2 HD movies over the Internet would probably not seem like a big deal.

Satoshi Nakamoto

The Cryptography Mailing List

5

The 51% Attack

In this post, Satoshi addresses an argument concerning the so-called 51% attack. In this scenario, a miner or group of miners could gain a majority of hash generation power (i.e., the proof-of-work) in order to initiate and then reverse transactions and so double spend, to prevent some transactions from being confirmed or to prevent some or all other miners from mining valid blocks.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Mon, 03 Nov 2008 11:45:58 -0800

> John Levine wrote:
>
> ---
>
> > Satoshi Nakamoto wrote:
> > As long as honest nodes control the most CPU power on the

> network, they can generate the longest chain and outpace any attackers.

But they don't. Bad guys routinely control zombie farms of 100,000 machines or more. People I know who run a blacklist of spam sending zombies tell me they often see a million new zombies a day.

This is the same reason that hashcash can't work on today's Internet–the good guys have vastly less computational firepower than the bad guys.

Thanks for bringing up that point.

I didn't really make that statement as strong as I could have. The requirement is that the good guys collectively have more CPU power than any single attacker.

There would be many smaller zombie farms that are not big enough to overpower the network, and they could still make money by generating bitcoins. The smaller farms are then the "honest nodes". (I need a better term than "honest") The more smaller farms resort to generating bitcoins, the highe the bar gets to overpower the network, making larger farms also too small to overpower it so that they may as well generate bitcoins too. According to the "long tail" theory, the small, medium and merely large farms put together should add up to a lot more than the biggest zombie farm.

Even if a bad guy does overpower the network, it's not like he's instantly rich. All he can accomplish is to take back money he himself spent, like bouncing a check. To exploit it, he would have to buy something from a merchant, wait till it ships, then overpower the network and try to take his money back. I don't think he could make as much money trying to pull a carding scheme like that as he could by generating bitcoins. With a zombie farm that big, he could generate more bitcoins than everyone else combined.

The Bitcoin network might actually reduce spam by diverting zombie farms to generating bitcoins instead.

Satoshi Nakamoto

The Cryptography Mailing List

6

About Centrally Controlled Networks Versus Peer-to-Peer Networks

SATOSHI MAKES A REFERENCE to the ability of governments to shut down any centralized system such as the music file-sharing website, Napster, or the digital gold currency E-gold. Pure peer-to-peer network systems have been demonstrated to be more resilient.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Fri, 07 Nov 2008 09:30:36 -0800

> [Lengthy exposition of vulnerability of a systm to use-of-force monopolies ellided.]
>
> You will not find a solution to political problems in cryptography.

Yes, but we can win a major battle in the arms race and gain a new territory of freedom for several years.

Governments are good at cutting off the heads of a centrally controlled networks like Napster, but pure P2P networks like Gnutella and Tor seem to be holding their own.

Satoshi

The Cryptography Mailing List

7

Satoshi on the Initial Inflation Rate of 35%

INITIALLY, with 50 bitcoins created every 10 minutes for the first few years, 2.6 million bitcoins were being created yearly. After Bitcoin started with a balance of 0 bitcoins in January, 2009, the rate of inflation of the bitcoin currency was initially staggering. However, the growth of demand for the currency given its very limited initial supply accounted for its high rate of inflation. In contrast, established national currencies such as Venezuela's Bolivar, Argentina's peso, or Zimbabwe's dollar began with sufficient and relatively stable supplies. However, the rate of printing of these currencies was then increased as a method for the country's government to fund its deficit spending.

There are three ways in which a government can fund deficit spending: currency inflation (printing new currency), borrowing from the

public, and taxation. Governments tend to favor currency by fiat (i.e., creating new currency), which allows it to blame the inevitable price increases on speculators rather than on its true culprit, currency inflation. This was the excuse used by Venezuela's government in 2013 and again in 2014. Were governments forced to use gold, silver, or bitcoins to fund their deficit spending, they would have to fund it with tax increases, a recourse not popular with the public, or with borrowing in the credit markets. This latter action leads to higher interest rates as demand for money to borrow increases, and, should governments not address their deficit spending with cuts in spending, they are forced to raise rates of taxation.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Sat, 08 Nov 2008 13:38:26 -0800

> Ray Dillinger:
>
> the "currency" is inflationary at about 35% as that's how much faster computers get annually . . . the inflation rate of 35% is almost guaranteed by the technology

Increasing hardware speed is handled: "To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases."

As computers get faster and the total computing power applied to creating bitcoins increases, the difficulty increases proportionally to keep the total new production constant. Thus, it is known in advance how many new bitcoins will be created every year in the future.

The fact that new coins are produced means the money supply

increases by a planned amount, but this does not necessarily result in inflation. If the supply of money increases at the same rate that the number of people using it increases, prices remain stable. If it does not increase as fast as demand, there will be deflation and early holders of money will see its value increase.

Coins have to get initially distributed somehow, and a constant rate seems like the best formula.

Satoshi Nakamoto

The Cryptography Mailing List

8

About Transactions

SEVERAL QUESTIONS and answers were covered in this post. Hal Finney, the first recipient of a bitcoin transaction, posed the questions.

In the first part, Satoshi explains how miners retain transactions until they form them into a block.

In the second, he explains how double spending cannot occur on a specific block chain and how only one block chain will prevail given that two miners solve their blocks simultaneously. It also covers how transactions need to be held for an hour by receivers until they are formally confirmed in the block chain. Satoshi refers to six blocks (10 minutes per block times six blocks gives an hour) as an appropriate amount of time for a transaction to be confirmed and forever made a part of the block chain.

To the third question, he describes what an attacker would have to do to "rewrite history", i.e., reconstruct and change the block chain. To

add or remove transactions in prior past blocks would require rewriting them faster than all miners on the network still working on the existing block chain. Remember from the discussion of orphan blocks that the longest block chain is what the network uses. Satoshi says: *The CPU power proof-of-work vote must have the final say. The only way for everyone to stay on the same page is to believe that the longest chain is always the valid one, no matter what.*

The fourth question concerns transaction verification of a payment transfer by a recipient.

The fifth question concerns the role of nodes (i.e., miners) in the system. When one miner discovers the proof-of-work (the hash with the appropriate number of leading 0s), it will broadcast the block it just "mined", which contains several transactions. Each miner on the network that receives this block has to validate it by checking the validity of each transaction the block contains.

Finally, Satoshi reports that he wrote the code prior to writing the white paper announcing Bitcoin in order to prove to himself that all issues were resolved.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Sun, 09 Nov 2008 11:13:34 -0800

> Hal Finney wrote:
>
> it is mentioned that if a broadcast transaction does not reach all nodes, it is OK, as it will get into the block chain before long. How does this happen - what if the node that creates the "next" block (the first node to find the hashcash collision) did not hear about the transaction, and then a few more blocks get added also by nodes that did not hear about that transaction? Do all the nodes that did hear it keep that transaction around, hoping to incorporate it into a block

> once they get lucky enough to be the one which finds the next collision?

Right, nodes keep transactions in their working set until they get into a block. If a transaction reaches 90% of nodes, then each time a new block is found, it has a 90% chance of being in it.

> Or for example, what if a node is keeping two or more chains around as it waits to see which grows fastest, and a block comes in for chain A which would include a double-spend of a coin that is in chain B? Is that checked for or not? (This might happen if someone double-spent and two different sets of nodes heard about the two different transactions with the same coin.)

That does not need to be checked for. The transaction in whichever branch ends up getting ahead becomes the valid one, the other is invalid. If someone tries to double spend like that, one and only one spend will always become valid, the others invalid.

Receivers of transactions will normally need to hold transactions for perhaps an hour or more to allow time for this kind of possibility to be resolved.

They can still re-spend the coins immediately, but they should wait before taking an action such as shipping goods.

> I also don't understand exactly how double-spending, or cancelling transactions, is accomplished by a superior attacker who is able to muster more computing power than all the honest participants. I see that he can create new blocks and add them to create the longest chain, but how can he erase or add old transactions in the chain? As the attacker sends out his new blocks, aren't there consistency checks which honest nodes can perform, to make sure that nothing got erased? More explanation of this attack would be helpful, in order to judge the gains to an attacker from this, versus simply using his computing power to mint new coins honestly.

The attacker isn't adding blocks to the end. He has to go back and redo the block his transaction is in and all the blocks after it, as well as any new blocks the network keeps adding to the end while he's doing that. He's rewriting history. Once his branch is longer, it becomes the new valid one.

This touches on a key point. Even though everyone present may see the shenanigans going on, there's no way to take advantage of that fact.

It is strictly necessary that the longest chain is always considered the valid one. Nodes that were present may remember that one branch was there first and got replaced by another, but there would be no way for them to convince those who were not present of this. We can't have subfactions of nodes that cling to one branch that they think was first, others that saw another branch first, and others that joined later and never saw what happened. The CPU power proof-of-work vote must have the final say. The only way for everyone to stay on the same page is to believe that the longest chain is always the valid one, no matter what.

> As far as the spending transactions, what checks does the recipient of a coin have to perform? Does she need to go back through the coin's entire history of transfers, and make sure that every transaction on the list is indeed linked into the "timestamp" block chain? Or can she just do the latest one?

The recipient just needs to verify it back to a depth that is sufficiently far back in the block chain, which will often only require a depth of 2 transactions. All transactions before that can be discarded.

> Do the timestamp nodes check transactions, making sure that the previous transaction on a coin is in the chain, thereby enforcing the rule that all transactions in the chain represent valid coins?

Right, exactly. When a node receives a block, it checks the

signatures of every transaction in it against previous transactions in blocks. Blocks can only contain transactions that depend on valid transactions in previous blocks or the same block. Transaction C could depend on transaction B in the same block and B depends on transaction A in an earlier block.

> Sorry about all the questions, but as I said this does seem to be a very promising and original idea, and I am looking forward to seeing how the concept is further developed. It would be helpful to see a more process oriented description of the idea, with concrete details of the data structures for the various objects (coins, blocks, transactions), the data which is included in messages, and algorithmic descriptions of the procedures for handling the various events which would occur in this system. You mentioned that you are working on an implementation, but I think a more formal, text description of the system would be a helpful next step.

I appreciate your questions. I actually did this kind of backwards. I had to write all the code before I could convince myself that I could solve every problem, then I wrote the paper. I think I will be able to release the code sooner than I could write a detailed spec. You're already right about most of your assumptions where you filled in the blanks.

Satoshi Nakamoto

The Cryptography Mailing List

9

On the Orphan Blocks

An "ORPHAN BLOCK" occurs when two miners satisfy the proof-of-work at approximately the same time. The two blocks created by the two miners are different since they may not contain all of the same Bitcoin transactions, in which case the transactions wherein the two "winning" miners transfer the block's transaction fees to their accounts are also different. But only one of those two blocks will ultimately be added to the block chain, while the other will become an "orphan block". Any transactions present in the orphan block but not included in the accepted block will be included in the next block for which miners are competing. For more details, see the explanation of orphan blocks in Chapter 2.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Sun, 09 Nov 2008 11:17:24 -0800

> James A. Donald wrote:
>
> OK, suppose one node incorporates a bunch of transactions in its proof of work, all of them honest legitimate single spends and another node incorporates a different bunch of transactions in its proof of work, all of them equally honest legitimate single spends, and both proofs are generated at about the same time.

What happens then?

They both broadcast their blocks. All nodes receive them and keep both, but only work on the one they received first. We'll suppose exactly half received one first, half the other.

In a short time, all the transactions will finish propagating so that everyone has the full set. The nodes working on each side will be trying to add the transactions that are missing from their side. When the next proof-of-work is found, whichever previous block that node was working on, that branch becomes longer and the tie is broken. Whichever side it is, the new block will contain the other half of the transactions, so in either case, the branch will contain all transactions. Even in the unlikely event that a split happened twice in a row, both sides of the second split would contain the full set of transactions anyway.

It's not a problem if transactions have to wait one or a few extra cycles to get into a block.

Satoshi Nakamoto

The Cryptography Mailing List

10

About Synchronization of Transactions

In this post, Satoshi explains what happens when a miner receives two conflicting transactions. The first transaction received is the one that the miner incorporates in the next proof-of-work. If more information is needed, see the explanation in Chapter 2.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Sun, 09 Nov 2008 11:14:17 -0800

James A. Donald wrote:

> The core concept is that lots of entities keep complete and consistent information as to who owns which bitcoins.
>
> But maintaining consistency is tricky. It is not clear to me what happens when someone reports one transaction to one maintainer, and someone else transports another transaction to another maintainer. The transaction cannot be known to be valid until it has been incorporated into a globally shared view of all past transactions, and no one can know that a globally shared view of all past transactions is globally shared until after some time has passed, and after many new transactions have arrived.
>
> Did you explain how to do this, and it just passed over my head, or were you confident it could be done, and a bit vague as to the details?

The proof-of-work chain is the solution to the synchronisation problem, and to knowing what the globally shared view is without having to trust anyone.

A transaction will quickly propagate throughout the network, so if two versions of the same transaction were reported at close to the same time, the one with the head start would have a big advantage in reaching many more nodes first. Nodes will only accept the first one they see, refusing the second one to arrive, so the earlier transaction would have many more nodes working on incorporating it into the next proof-of-work. In effect, each node votes for its viewpoint of which transaction it saw first by including it in its proof-of-work effort.

If the transactions did come at exactly the same time and there was an even split, it's a toss up based on which gets into a proof-of-work first, and that decides which is valid.

When a node finds a proof-of-work, the new block is propagated throughout the network and everyone adds it to the chain and starts working on the next block after it. Any nodes that had the

other transaction will stop trying to include it in a block, since it's now invalid according to the accepted chain.

The proof-of-work chain is itself self-evident proof that it came from the globally shared view. Only the majority of the network together has enough CPU power to generate such a difficult chain of proof-of-work. Any user, upon receiving the proof-of-work chain, can see what the majority of the network has approved. Once a transaction is hashed into a link that's a few links back in the chain, it is firmly etched into the global history.

Satoshi Nakamoto

The Cryptography Mailing List

11

Satoshi Discusses Transaction Fees

THIS POST DISCUSSES USE of transaction fees as opposed to *seigniorage* as a mean of paying miners for their work of maintaining the Bitcoin network. *Seigniorage* is an economic term used to describe the creation of additional units of a currency. When all bitcoins have been mined and the maximum of 21 million BTC has been created, incentives for miners to work to maintain Bitcoin will come only from transaction fees collected in the course of maintaining Bitcoin. However, prior to this, the yearly rate of bitcoin inflation will be so low in the end that it will effectively be the same as after all bitcoins have been mined.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Mon, 10 Nov 2008 11:09:26 -0800

> James A. Donald wrote:
>
> > Furthermore, it cannot be made to work, as in the proposed system the work of tracking who owns what coins is paid for by seigniorage, which requires inflation.

If you're having trouble with the inflation issue, it's easy to tweak it for transaction fees instead. It's as simple as this: let the output value from any transaction be 1 cent less than the input value. Either the client software automatically writes transactions for 1 cent more than the intended payment value, or it could come out of the payee's side. The incentive value when a node finds a proof-of-work for a block could be the total of the fees in the block.

Satoshi Nakamoto

The Cryptography Mailing List

12

On Confirmation and Block Time

In the first answer below, Satoshi addresses *double spend* and *confirmation*.

In the second answer, he covers how the difficulty on the proof-of-work is adjusted based on the effective time between each block so that the network attempts to maintain 10 minutes per block. Chapter 2's discussion on proof-of-work compared it to a lottery. A maximum number, in hexadecimal or base 16, is selected, and the miners' proof-of-work consists of generating a number that is less than this number. The number is generated through the Bitcoin system and is random. The first miner to obtain a hash output less than the maximum "wins" the right to process that block and be awarded its transaction fees and the 25 BTC awarded per block. The value chosen for the maximum determines the level of difficulty of the proof-of-work; the larger the

value, the more likely a hash output generated by the miner's system is to fall below the maximum, and the smaller the number, the less likely the miner's number is to fall below the maximum.

The last question addressed is in regard to the speed of the transaction not being a feature. He makes the point that bouncing checks and credit card chargebacks can take several days or even weeks to process, in contrast to the 60 minutes or so for Bitcoin to validate with a high level of confidence a fully irreversible bitcoin transaction.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Tue, 11 Nov 2008 06:30:22 -0800

> James A. Donald wrote:
>
> So what happened to the coin that lost the race?

. . . it is a bit harsh if the guy who came second is likely to lose his coin.

When there are multiple double-spent versions of the same transaction, one and only one will become valid.

The receiver of a payment must wait an hour or so before believing that it's valid. The network will resolve any possible double-spend races by then.

The guy who received the double-spend that became invalid never thought he had it in the first place. His software would have shown the transaction go from "unconfirmed" to "invalid". If necessary, the UI can be made to hide transactions until they're sufficiently deep in the block chain.

> Further, your description of events implies restrictions on timing and coin generation - that the entire network generates coins slowly compared to the time required for news of a new

> coin to flood the network

Sorry if I didn't make that clear. The target time between blocks will probably be 10 minutes.

Every block includes its creation time. If the time is off by more than 36 hours, other nodes won't work on it. If the timespan over the last 6*24*30 blocks is less than 15 days, blocks are being generated too fast and the proof-of-work difficulty doubles. Everyone does the same calculation with the same chain data, so they all get the same result at the same link in the chain.

> We want spenders to have certainty that their transaction is valid at the time it takes a spend to flood the network, not at the time it takes for branch races to be resolved.

Instantant non-repudiability is not a feature, but it's still much faster than existing systems. Paper cheques can bounce up to a week or two later. Credit card transactions can be contested up to 60 to 180 days later. Bitcoin transactions can be sufficiently irreversible in an hour or two.

> If one node is ignoring all spends that it does not care about, it suffers no adverse consequences.

With the transaction fee based incentive system I recently posted, nodes would have an incentive to include all the paying transactions they receive.

Satoshi Nakamoto

The Cryptography Mailing List

13

The Byzantine General's Problem

In what is possibly the most interesting post made by Satoshi, he explains how the block chain solves a problem in computer science known as the "Byzantine fault tolerance", a more generalized version of the "Two Generals' Problem". In this problem, two (or more) persons need to share information in an unreliable communication environment, where messages sent can be lost or tampered with. The statement of the problem first appeared in the 1970s in network computing literature, and at that time the problem was considered unsolvable. In this post, Satoshi claims that Bitcoin solves it.

To illustrate the problem, imagine that two generals are required to attack a city at the same time. If either one attacks and the other does not, the forces of the attacking general will be annihilated by the city's defenses. Communication between the generals is unreliable;

the courier sending the message regarding when to attack must go through the city and so could be intercepted. The first general can, by 9 am, dispatch the messenger with the message communicating that the attack will commence on that same day. However, once dispatched, the first general will have no idea as to whether or not the messenger got through. This uncertainty may lead the first general to hesitate to attack since he might be attacking alone if the second general never received his message.

Knowing all this, the second general may send a confirmation back to the first to indicate that he received the message to attack. But that message, too, could be intercepted, leading the second general to hesitate as well. The first general could be sending a confirmation of the confirmation, but that too could have been intercepted. Hence, again, the first general could hesitate unless he gets back a confirmation of this confirmation of the first confirmation. This process could be carried out ad infinitum with no way for either general to know whether messages were dispatched or whether they were but were intercepted by the enemy.

To learn more, read the section "Illustrating the problem" in the following Wikipedia article:

http://en.wikipedia.org/wiki/Two_Generals%27_Problem

See also this article on the Byzantine fault tolerance:
http://en.wikipedia.org/wiki/Byzantine_fault_tolerance

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Thu, 13 Nov 2008 19:34:25 -0800

James A. Donald wrote:

It is not sufficient that everyone knows X. We also need

> everyone to know that everyone knows X, and that everyone knows that everyone knows that everyone knows X - which, as in the Byzantine Generals problem, is the classic hard problem of distributed data processing.

The proof-of-work chain is a solution to the Byzantine Generals' Problem. I'll try to rephrase it in that context.

A number of Byzantine Generals each have a computer and want to attack the King's wi-fi by brute forcing the password, which they've learned is a certain number of characters in length. Once they stimulate the network to generate a packet, they must crack the password within a limited time to break in and erase the logs, otherwise they will be discovered and get in trouble. They only have enough CPU power to crack it fast enough if a majority of them attack at the same time.

They don't particularly care when the attack will be, just that they all agree. It has been decided that anyone who feels like it will announce a time, and whatever time is heard first will be the official attack time. The problem is that the network is not instantaneous, and if two generals announce different attack times at close to the same time, some may hear one first and others hear the other first.

They use a proof-of-work chain to solve the problem. Once each general receives whatever attack time he hears first, he sets his computer to solve an extremely difficult proof-of-work problem that includes the attack time in its hash. The proof-of-work is so difficult, it's expected to take 10 minutes of them all working at once before one of them finds a solution. Once one of the generals finds a proof-of-work, he broadcasts it to the network, and everyone changes their current proof-of-work computation to include that proof-of-work in the hash they're working on. If anyone was working on a different attack time, they switch to this one, because its proof-of-work chain is now longer.

After two hours, one attack time should be hashed by a chain of 12 proofs-of-work. Every general, just by verifying the difficulty of the proof-of-work chain, can estimate how much parallel CPU power per hour was expended on it and see that it must have required the majority of the computers to produce that much proof-of-work in the allotted time. They had to all have seen it because the proof-of-work is proof that they worked on it. If the CPU power exhibited by the proof-of-work chain is sufficient to crack the password, they can safely attack at the agreed time.

The proof-of-work chain is how all the synchronisation, distributed database and global view problems you've asked about are solved.

The Cryptography Mailing List

14

On Block Time, an Automated Test, and the Libertarian Viewpoint

IN THIS POST, Satoshi explains why a single pending transaction pool is required and how these transactions are kept given that parallel branches of blocks exist. He references a few functions within the code. Recall the discussion on proof-of-work in Chapter 2. Not all miners might have assembled the same transactions, some of which might have come too late to be included in the block on which they are working. As new transactions arrive while they are working on the hash for their existing block, they will store these transactions in a transaction pool.

Then, he touches again on transaction propagation and the 10 minutes allocated per creation of a block, discussing the issue as to whether that might be too short a period of time.

Lastly, he makes a reference to how Bitcoin could be attractive to libertarians, people who advocate individual liberties.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Fri, 14 Nov 2008 14:29:22 -0800

> Hal Finney wrote:
>
> I think it is necessary that nodes keep a separate pending-transaction list associated with each candidate chain.
>
> . . . One might also ask . . . how many candidate chains must a given node keep track of at one time, on average?

Fortunately, it's only necessary to keep a pending-transaction pool for the current best branch. When a new block arrives for the best branch, ConnectBlock removes the block's transactions from the pending-tx pool. If a different branch becomes longer, it calls DisconnectBlock on the main branch down to the fork, returning the block transactions to the pending-tx pool, and calls ConnectBlock on the new branch, sopping back up any transactions that were in both branches. It's expected that reorgs like this would be rare and shallow.

With this optimisation, candidate branches are not really any burden. They just sit on the disk and don't require attention unless they ever become the main chain.

> Or as James raised earlier, if the network broadcast is reliable but depends on a potentially slow flooding algorithm, how does that impact performance?

Broadcasts will probably be almost completely reliable. TCP transmissions are rarely ever dropped these days, and the broadcast protocol has a retry mechanism to get the data from other nodes after a while. If broadcasts turn out to be slower in practice than expected, the target time between blocks may have to be increased to avoid wasting resources. We want blocks to usually propagate in much less time than it takes to generate them, otherwise nodes would spend too much time working on obsolete blocks.

I'm planning to run an automated test with computers randomly sending payments to each other and randomly dropping packets.

> 3. The bitcoin system turns out to be socially useful and valuable, so that node operators feel that they are making a beneficial contribution to the world by their efforts (similar to the various "@Home" compute projects where people volunteer their compute resources for good causes).

In this case it seems to me that simple altruism can suffice to keep the network running properly.

It's very attractive to the libertarian viewpoint if we can explain it properly. I'm better with code than with words though.

Satoshi Nakamoto

The Cryptography Mailing List

15

More on Double Spend, Proof-of-Work, and Transaction Fees

In this exchange, Satoshi provides several clarifications and discusses compensation of miners (i.e., nodes) via transaction fees once the entire supply of bitcoins has been created.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Mon, 17 Nov 2008 09:04:47 -0800

I'll try and hurry up and release the sourcecode as soon as possible to serve as a reference to help clear up all these implementation questions.

> Ray Dillinger (Bear) wrote:
>
> When a coin is spent, the buyer and seller digitally sign a (blinded) transaction record.

Only the buyer signs, and there's no blinding.

> If someone double spends, then the transaction record can be unblinded revealing the identity of the cheater.

Identities are not used, and there's no reliance on recourse. It's all prevention.

> This is done via a fairly standard cut-and-choose algorithm where the buyer responds to several challenges with secret shares

No challenges or secret shares. A basic transaction is just what you see in the figure in section 2. A signature (of the buyer) satisfying the public key of the previous transaction, and a new public key (of the seller) that must be satisfied to spend it the next time.

> They may also receive chains as long as the one they're trying to extend while they work, in which the last few "links" are links that are *not* in common with the chain on which they're working.
>
> These they ignore.

Right, if it's equal in length, ties are broken by keeping the earliest one received.

> If it contains a double spend, then they create a "transaction" which is a proof of double spending, add it to their pool A,

> broadcast it, and continue work.

There's no need for reporting of "proof of double spending" like that. If the same chain contains both spends, then the block is invalid and rejected.

Same if a block didn't have enough proof-of-work. That block is invalid and rejected. There's no need to circulate a report about it. Every node could see that and reject it before relaying it.

If there are two competing chains, each containing a different version of the same transaction, with one trying to give money to one person and the other trying to give the same money to someone else, resolving which of the spends is valid is what the whole proof-of-work chain is about.

We're not "on the lookout" for double spends to sound the alarm and catch the cheater. We merely adjudicate which one of the spends is valid. Receivers of transactions must wait a few blocks to make sure that resolution has had time to complete. Would be cheaters can try and simultaneously double-spend all they want, and all they accomplish is that within a few blocks, one of the spends becomes valid and the others become invalid. Any later double-spends are immediately rejected once there's already a spend in the main chain.

Even if an earlier spend wasn't in the chain yet, if it was already in all the nodes' pools, then the second spend would be turned away by all those nodes that already have the first spend.

> If the new chain is accepted, then they give up on adding their current link, dump all the transactions from pool L back into pool A (along with transactions they've received or created since starting work), eliminate from pool A those transaction records which are already part of a link in the new chain, and start work again trying to extend the new chain.

Right. They also refresh whenever a new transaction comes in, so L

pretty much contains everything in A all the time.

> CPU-intensive digital signature algorithm to sign the chain including the new block L.

It's a Hashcash style SHA-256 proof-of-work (partial pre-image of zero), not a signature.

> Is there a mechanism to make sure that the "chain" does not consist solely of links added by just the 3 or 4 fastest nodes? 'Cause a broadcast transaction record could easily miss those 3 or 4 nodes and if it does, and those nodes continue to dominate the chain, the transaction might never get added.

If you're thinking of it as a CPU-intensive digital signing, then you may be thinking of a race to finish a long operation first and the fastest always winning.

The proof-of-work is a Hashcash style SHA-256 collision finding. It's a memoryless process where you do millions of hashes a second, with a small chance of finding one each time. The 3 or 4 fastest nodes' dominance would only be proportional to their share of the total CPU power. Anyone's chance of finding a solution at any time is proportional to their CPU power.

There will be transaction fees, so nodes will have an incentive to receive and include all the transactions they can. Nodes will eventually be compensated by transaction fees alone when the total coins created hits the pre-determined ceiling.

> Also, the work requirement for adding a link to the chain should vary (again exponentially) with the number of links added to that chain in the previous week, causing the rate of coin generation (and therefore inflation) to be strictly controlled.

Right.

> You need coin aggregation for this to scale. There needs to

> be a "provable" transaction where someone retires ten single coins and creates a new coin with denomination ten, etc.

Every transaction is one of these. Section 9, Combining and Splitting Value.

Satoshi Nakamoto

The Cryptography Mailing List

16

On Elliptic Curve Cryptography, Denial of Service Attacks, and Confirmation

SATOSHI COVERS transaction signatures, adds a bit more on denial of service attacks, and, finally, revisits transaction speed. A merchant could wait for 2 minutes after the consumer has made the transaction with his smartphone. Then the merchant (or the Bitcoin payment service company the merchant has chosen) would watch for double spend transactions on the Bitcoin network. Imagine that a consumer makes a transaction which we will call "X" in which he or she pays 1.5 BTC from a Bitcoin address ABC that holds 2 BTC. The consumer's balance then falls to 0.5 BTC once the payment has

been fully confirmed. Discussed here are the actions the merchant has to perform in order to monitor the network to see if any other transactions involving bitcoin address ABC appear and, if so, if the amount involved exceeds 0.5 BTC. If transactions meeting this criterion are detected within say, 2 minutes, the payment is considered not valid. Waiting for 2 minutes gives plenty of lead for transaction "X" to clear prior to any competing transactions coming later from Bitcoin address ABC. This indicates to the merchant that transaction "X" is very likely to be included in the current block of the majority of Bitcoin miners on which they are working and hence assures its eventual inclusion in the block chain.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Mon, 17 Nov 2008 09:06:02 -0800

> Ray Dillinger wrote:
>
> > One way to do this would be to have the person recieving the coin generate an asymmetric key pair, and then have half of it published with the transaction. In order to spend the coin later, s/he must demonstrate posession of the other half of the asymmetric key pair, probably by using it to sign the key provided by the new seller.

Right, it's ECC digital signatures. A new key pair is used for every transaction.

It's not pseudonymous in the sense of nyms identifying people, but it is at least a little pseudonymous in that the next action on a coin can be identified as being from the owner of that coin.

> Mmmm. I don't know if I'm comfortable with that. You're saying there's no effort to identify and exclude nodes that

> don't cooperate? I suspect this will lead to trouble and possible DOS attacks.

There is no reliance on identifying anyone. As you've said, it's futile and can be trivially defeated with sock puppets.

The credential that establishes someone as real is the ability to supply CPU power.

> Until . . . until what? How does anybody know when a transaction has become irrevocable? Is "a few" blocks three? Thirty? A hundred? Does it depend on the number of nodes? Is it logarithmic or linear in number of nodes?

Section 11 calculates the worst case under attack. Typically, 5 or 10 blocks is enough for that. If you're selling something that doesn't merit a network-scale attack to steal it, in practice you could cut it closer.

> But in the absence of identity, there's no downside to them if spends become invalid, if they've already received the goods they double-spent for (access to website, download, whatever). The merchants are left holding the bag with "invalid" coins, unless they wait that magical "few blocks" (and how can they know how many?) before treating the spender as having paid.
>
> The consumers won't do this if they spend their coin and it takes an hour to clear before they can do what they spent their coin on. The merchants won't do it if there's no way to charge back a customer when they find the that their coin is invalid because the customer has doublespent.

This is a version 2 problem that I believe can be solved fairly satisfactorily for most applications.

The race is to spread your transaction on the network first. Think 6 degrees of freedom -- it spreads exponentially. It would only take something like 2 minutes for a transaction to spread widely

enough that a competitor starting late would have little chance of grabbing very many nodes before the first one is overtaking the whole network.

During those 2 minutes, the merchant's nodes can be watching for a double-spent transaction. The double-spender would not be able to blast his alternate transaction out to the world without the merchant getting it, so he has to wait before starting.

If the real transaction reaches 90% and the double-spent tx reaches 10%, the double-spender only gets a 10% chance of not paying, and 90% chance his money gets spent. For almost any type of goods, that's not going to be worth it for the scammer.

Information based goods like access to website or downloads are non-fencible. Nobody is going to be able to make a living off stealing access to websites or downloads. They can go to the file sharing networks to steal that. Most instant-access products aren't going to have a huge incentive to steal.

If a merchant actually has a problem with theft, they can make the customer wait 2 minutes, or wait for something in e-mail, which many already do. If they really want to optimize, and it's a large download, they could cancel the download in the middle if the transaction comes back double-spent. If it's website access, typically it wouldn't be a big deal to let the customer have access for 5 minutes and then cut off access if it's rejected. Many such sites have a free trial anyway.

Satoshi Nakamoto

The Cryptography Mailing List

17

More on the Transaction Pool, Networking Broadcast, and Coding Details

In the first section below, Satoshi expands on the transaction pool. He then describes his experiment on the networking broadcast mechanism where nodes request items from their neighbors. Lastly, Satoshi mentions that he has been working on the code for the last 18 months.

Re: Bitcoin P2P e-cash paper

Satoshi Nakamoto Mon, 17 Nov 2008 13:33:04 -0800

> James A. Donald wrote:
>
> > Satoshi wrote:
> > Fortunately, it's only necessary to keep a pending-transaction pool for the current best branch.
>
> This requires that we know, that is to say an honest well behaved peer whose communications and data storage is working well knows, what the current best branch is -

I mean a node only needs the pending-tx pool for the best branch it has. The branch that it currently thinks is the best branch. That's the branch it'll be trying to make a block out of, which is all it needs the pool for.

> Broadcasts will probably be almost completely reliable.
>
> Rather than assuming that each message arrives at least once, we have to make a mechanism such that the information arrives even though conveyed by messages that frequently fail to arrive.

I think I've got the peer networking broadcast mechanism covered.

Each node sends its neighbours an inventory list of hashes of the new blocks and transactions it has. The neighbours request the items they don't have yet. If the item never comes through after a timeout, they request it from another neighbour that had it. Since all or most of the neighbours should eventually have each item, even if the coms get fumbled up with one, they can get it from any of the others, trying one at a time.

The inventory-request-data scheme introduces a little latency, but it ultimately helps speed more by keeping extra data blocks off the transmit queues and conserving bandwidth.

> You have an outline and proposal for such a design, which is a big step forward, but the devil is in the little details.

I believe I've worked through all those little details over the last year and a half while coding it, and there were a lot of them. The functional details are not covered in the paper, but the sourcecode is coming soon. I sent you the main files. (available by request at the moment, full release soon)

Satoshi Nakamoto

The Cryptography Mailing List

18

First Release of Bitcoin

In this post, Satoshi announces the very first release of the Bitcoin software on *sourceforge.net*. *Sourceforge.net* is like github or other online services allowing people to share documents and source code. This release is no longer available there but has been recopied at the following locations:

http://www.zorinaq.com/pub/bitcoin-0.1.0.rar
http://www.zorinaq.com/pub/bitcoin-0.1.0.tgz
http://we.lovebitco.in/bitcoin-0.1.0.rar
http://www.bitcointrading.com/files/bitcoin-0.1.0.rar

This software is open source, meaning that the code is available, copyright free for use, reproduction, and modification.

Bitcoin v0.1 released

Satoshi Nakamoto Fri, 09 Jan 2009 17:05:49 -0800

Announcing the first release of Bitcoin, a new electronic cash system that uses a peer-to-peer network to prevent double-spending. It's completely decentralized with no server or central authority.

See bitcoin.org for screenshots.

Download link:

http://downloads.sourceforge.net/bitcoin/bitcoin-0.1.0.rar

Windows only for now. Open source C++ code is included.

- Unpack the files into a directory
- Run BITCOIN.EXE
- It automatically connects to other nodes

If you can keep a node running that accepts incoming connections, you'll really be helping the network a lot. Port 8333 on your firewall needs to be open to receive incoming connections.

The software is still alpha and experimental. There's no guarantee the system's state won't have to be restarted at some point if it becomes necessary, although I've done everything I can to build in extensibility and versioning.

You can get coins by getting someone to send you some, or turn on

Options->Generate Coins to run a node and generate blocks. I made the proof-of-work difficulty ridiculously easy to start with, so for a little while in the beginning a typical PC will be able to generate coins in just a few hours. It'll get a lot harder when competition makes the automatic adjustment drive up the difficulty. Generated coins must wait 120 blocks to mature before they can be spent.

There are two ways to send money. If the recipient is online, you can enter their IP address and it will connect, get a new public key and send the transaction with comments. If the recipient is not online, it is possible to send to their Bitcoin address, which is a hash of their public key that they give you. They'll receive the transaction the next time they connect and get the block it's in. This method has the disadvantage that no comment information is sent, and a bit of privacy may be lost if the address is used multiple times, but it is a useful alternative if both users can't be online at the same time or the recipient can't receive incoming connections.

Total circulation will be 21,000,000 coins. It'll be distributed to network nodes when they make blocks, with the amount cut in half every 4 years.

first 4 years: 10,500,000 coins
next 4 years: 5,250,000 coins
next 4 years: 2,625,000 coins
next 4 years: 1,312,500 coins
etc. . . .

When that runs out, the system can support transaction fees if needed. It's based on open market competition, and there will probably always be nodes willing to process transactions for free.

Satoshi Nakamoto

19

On the Purpose For Which Bitcoin Could Be Used First

FROM THIS READING, we can extrapolate that Satoshi Nakamoto did not expect Bitcoin to achieve such huge success so rapidly. He states that the first use might be for micropayments or a porn site. Interestingly, those were not the first actual uses. Satoshi also makes the suggestion that famous people could use it so that fans could send them personal messages.

He also observes, "It might make sense just to get some in case it catches on." I'm assuming he followed his own advice. A bitcoin was worth over $600 in early 2014 compared to pennies in its first year of existence.

Re: Bitcoin v0.1 released

Satoshi Nakamoto Sat, 17 Jan 2009 06:58:44 -0800

> Dustin D. Trammell wrote:
>
> > Satoshi Nakamoto wrote:
> > You know, I think there were a lot more people interested in the 90's, but after more than a decade of failed Trusted Third Party based systems (Digicash, etc), they see it as a lost cause. I hope they can make the distinction that this is the first time I know of that we're trying a non-trust-based system.
>
> Yea, that was the primary feature that caught my eye. The real trick will be to get people to actually value the BitCoins so that they become currency.

I would be surprised if 10 years from now we're not using electronic currency in some way, now that we know a way to do it that won't inevitably get dumbed down when the trusted third party gets cold feet.

It could get started in a narrow niche like reward points, donation tokens, currency for a game or micropayments for adult sites. Initially it can be used in proof-of-work applications for services that could almost be free but not quite.

It can already be used for pay-to-send e-mail. The send dialog is resizeable and you can enter as long of a message as you like. It's sent directly when it connects. The recipient doubleclicks on the transaction to see the full message. If someone famous is getting more e-mail than they can read, but would still like to have a way for fans to contact them, they could set up Bitcoin and give out the IP address on their website. "Send X bitcoins to my priority hotline at this IP and I'll read the message personally."

Subscription sites that need some extra proof-of-work for their free

trial so it doesn't cannibalize subscriptions could charge bitcoins for the trial.

It might make sense just to get some in case it catches on. If enough people think the same way, that becomes a self fulfilling prophecy. Once it gets bootstrapped, there are so many applications if you could effortlessly pay a few cents to a website as easily as dropping coins in a vending machine.

Satoshi Nakamoto

http://www.bitcoin.org

This subject was revisited on the BitcoinTalk forum later:

Re: Porn

Posted by satoshi, September 23, 2010, 05:56:55 PM

Bitcoin would be convenient for people who don't have a credit card or don't want to use the cards they have, either don't want the spouse to see it on the bill or don't trust giving their number to "porn guys", or afraid of recurring billing.

20

"Proof-of-Work" Tokens and Spammers

HERE IS AN INTERESTING CONVERSATION between Hal Finney, a well-known developer in the cryptography industry, and Satoshi Nakamoto that focuses on how Bitcoin's proof-of-work could be used to limit spammers or to reward spam recipients. Hal Finney is credited with creating the first "reusable proof-of-work system", a variant of Bitcoin's proof-of-work that is not necessary to be understood for this topic to be comprehensible. Also Hal Finney is the recipient of the first Bitcoin transaction, whose sender was Satoshi himself.

Re: Bitcoin v0.1 released

Satoshi Nakamoto Sun, 25 Jan 2009 08:34:34 -0800

> Hal Finney wrote:
>
> * Spammer botnets could burn through pay-per-send email filters trivially
>
> If POW tokens do become useful, and especially if they become money, machines will no longer sit idle. Users will expect their computers to be earning them money (assuming the reward is greater than the cost to operate). A computer whose earnings are being stolen by a botnet will be more noticeable to its owner than is the case today, hence we might expect that in that world, users will work harder to maintain their computers and clean them of botnet infestations.

Another factor that would mitigate spam if POW tokens have value: there would be a profit motive for people to set up massive quantities of fake e-mail accounts to harvest POW tokens from spam. They'd essentially be reverse-spamming the spammers with automated mailboxes that collect their POW and don't read the message. The ratio of fake mailboxes to real people could become too high for spam to be cost effective.

The process has the potential to establish the POW token's value in the first place, since spammers that don't have a botnet could buy tokens from harvesters. While the buying back would temporarily let more spam through, it would only hasten the self-defeating cycle leading to too many harvesters exploiting the spammers.

Interestingly, one of the e-gold systems already has a form of spam called "dusting". Spammers send a tiny amount of gold dust in order to put a spam message in the transaction's comment field. If the system let users configure the minimum payment they're willing to receive, or at least the minimum that can have a message with it, users could set how much they're willing to get paid to receive spam.

Satoshi Nakamoto

The Cryptography Mailing List

21

Bitcoin Announced on P2P Foundation

SATOSHI ANNOUNCES Bitcoin v0.1 at *p2pfoundation.ning.com*. This is another forum that involves peer-to-peer technology. Rather than copying the exact same text of his original announcement posted on the Cryptography mailing list, Satoshi wrote a slightly different announcement for publication here.

Bitcoin open source implementation of P2P currency

Satoshi Nakamoto February 11, 2009 at 22:27

I've developed a new open source P2P e-cash system called Bitcoin. It's completely decentralized, with no central server or

trusted parties, because everything is based on crypto proof instead of trust. Give it a try, or take a look at the screenshots and design paper:

Download Bitcoin v0.1 at http://www.bitcoin.org

The root problem with conventional currency is all the trust that's required to make it work. The central bank must be trusted not to debase the currency, but the history of fiat currencies is full of breaches of that trust. Banks must be trusted to hold our money and transfer it electronically, but they lend it out in waves of credit bubbles with barely a fraction in reserve. We have to trust them with our privacy, trust them not to let identity thieves drain our accounts. Their massive overhead costs make micropayments impossible.

A generation ago, multi-user time-sharing computer systems had a similar problem. Before strong encryption, users had to rely on password protection to secure their files, placing trust in the system administrator to keep their information private. Privacy could always be overridden by the admin based on his judgment call weighing the principle of privacy against other concerns, or at the behest of his superiors. Then strong encryption became available to the masses, and trust was no longer required. Data could be secured in a way that was physically impossible for others to access, no matter for what reason, no matter how good the excuse, no matter what.

It's time we had the same thing for money. With e-currency based on cryptographic proof, without the need to trust a third party middleman, money can be secure and transactions effortless.

One of the fundamental building blocks for such a system is digital signatures. A digital coin contains the public key of its owner. To transfer it, the owner signs the coin together with the public key of the next owner. Anyone can check the signatures to verify the chain of ownership. It works well to secure ownership, but leaves one big problem unsolved: double-spending. Any owner could try

to re-spend an already spent coin by signing it again to another owner. The usual solution is for a trusted company with a central database to check for double-spending, but that just gets back to the trust model. In its central position, the company can override the users, and the fees needed to support the company make micropayments impractical.

Bitcoin's solution is to use a peer-to-peer network to check for double-spending. In a nutshell, the network works like a distributed timestamp server, stamping the first transaction to spend a coin. It takes advantage of the nature of information being easy to spread but hard to stifle. For details on how it works, see the design paper at http://www.bitcoin.org/bitcoin.pdf

The result is a distributed system with no single point of failure. Users hold the crypto keys to their own money and transact directly with each other, with the help of the P2P network to check for double-spending.

Satoshi Nakamoto
http://www.bitcoin.org

22

On Decentralization as Key to Success

SATOSHI TALKS HERE about the importance of a decentralized currency as a key to success. As stated earlier, the government's ability to control the supply of a currency provides an easy way to finance deficit spending. Any centrally controlled electronic currencies that have appeared so far have been dismantled by governments for various reasons. Typical reasons include facilitating money laundering or the purchase of drugs, even though US dollars are the main choice for these financial activities.

Re: Bitcoin open source implementation of P2P currency

Satoshi Nakamoto February 15, 2009 at 16:42

> Sepp Hasslberger wrote:
>
> Could there be synergies with bitcoin?
>
> http://opencoin.org/

Could be. They're talking about the old Chaumian central mint stuff, but maybe only because that was the only thing available. Maybe they would be interested in going in a new direction.

A lot of people automatically dismiss e-currency as a lost cause because of all the companies that failed since the 1990's. I hope it's obvious it was only the centrally controlled nature of those systems that doomed them. I think this is the first time we're trying a decentralized, non-trust-based system.

23

On the Subject of Money Supply

SATOSHI EXPLAINS his general concept on this forum and follows up about the issue of money supply versus the population. He then compares Bitcoin to precious metals and refers to a feedback loop on the price which could occur when number of users grows faster than the supply of bitcoins. Interestingly, this was indeed what did occur.

Imagine if the population were to discover, through real life experience, what it is to conduct their lives with a currency that does not lose its value, but in reality gains in value. As our economy grows and as our manufacturing capabilities increase, prices go down. The only reason that prices are not going down today—except in products where improvements are very rapid (e.g., computers)—is because of government-caused currency inflation.

Re: Bitcoin open source implementation of P2P currency

Satoshi Nakamoto February 18, 2009 at 20:50

It is a global distributed database, with additions to the database by consent of the majority, based on a set of rules they follow:

- Whenever someone finds proof-of-work to generate a block, they get some new coins
- The proof-of-work difficulty is adjusted every two weeks to target an average of 6 blocks per hour (for the whole network)
- The coins given per block is cut in half every 4 years

You could say coins are issued by the majority. They are issued in a limited, predetermined amount.

As an example, if there are 1000 nodes, and 6 get coins each hour, it would likely take a week before you get anything.

To Sepp's question, indeed there is nobody to act as central bank or federal reserve to adjust the money supply as the population of users grows. That would have required a trusted party to determine the value, because I don't know a way for software to know the real world value of things. If there was some clever way, or if we wanted to trust someone to actively manage the money supply to peg it to something, the rules could have been programmed for that.

In this sense, it's more typical of a precious metal. Instead of the supply changing to keep the value the same, the supply is predetermined and the value changes. As the number of users grows, the value per coin increases. It has the potential for a positive feedback loop; as users increase, the value goes up, which could attract more users to take advantage of the increasing value.

24

Release of Bitcoin v0.1.3

In this release, the software has fixed communication issues. Satoshi talks about maturation countdown for block proof-of-work discovery, i.e., the reward miners obtain from solving a block.

[bitcoin-list] Bitcoin v0.1.3

Satoshi Nakamoto 2009-01-12 22:48:23

It looks like we're through with the worst of the Internet connection issues. 0.1.3 fixed a problem where your node's communications could go dead after a while. The network is running much more smoothly now with this version.

If you've successfully generated a block, you've seen it has a maturation countdown before you can spend it. Once it matures, the Credit column will change from 0.00 to 50.00. For a block to be valid, it has to be broadcasted to the network and get into the block chain, which is why Generate does not run if you're not connected. If you generated a block without being connected, the network wouldn't know about it and would continue building the chain without it, leaving it behind, and the maturation countdown would change to "(not accepted)" when your node sees that it wasn't used. If you subtract 1 from the status column, that's how many blocks have been chained after yours.

Satoshi Nakamoto

25

On Timestamping Documents

HERE, Hal mentions that some people suggested using the block chain to timestamp documents by way of an extra hash. (See the earlier explanation of cryptographic hash in the section entitled *Cryptographic hash function—a digital "fingerprint"* in Chapter 2.)

[bitcoin-list] Bitcoin v0.1.5 released

Satoshi Nakamoto 2009-03-04 16:29:12

Hal Finney wrote:

That sounds good. I'd also like to be able to run multiple

> coin/block generators on multiple machines, all behind a single NAT address. I haven't tried this yet so I don't know if it works on the current software.

The current version will work fine. They'll each connect over the Internet, while incoming connections only come to the host that port 8333 is routed to.

As an optimisation, I'll make a switch "-connect=1.2.3.4" to make it only connect to a specific address. You could make your extra nodes connect to your primary, and only the primary connects over the Internet. It doesn't really matter for now, since the network would have to get huge before the bandwidth is anything more than trivial.

> BTW I don't remember if we talked about this, but the other day some people were mentioning secure timestamping. You want to be able to prove that a certain document existed at a certain time in the past. Seems to me that bitcoin's stack of blocks would be perfect for this.

Indeed, Bitcoin is a distributed secure timestamp server for transactions. A few lines of code could create a transaction with an extra hash in it of anything that needs to be timestamped. I should add a command to timestamp a file that way.

> Later I want to add interfaces to make it really easy to integrate into websites from any server side language.

> Right, and I'd like to see more of a library interface that could be called from programming or scripting languages, on the client side as well.

Exactly.

Satoshi Nakamoto

http://www.bitcoin.org

26

Bitcointalk Forum Welcome Message

SATOSHI ANNOUNCES the launch of a new forum dedicated to Bitcoin on *sourceforge.net*.

Welcome to the new Bitcoin forum!

Satoshi Nakamoto November 22, 2009, 06:04:28 PM

Welcome to the new Bitcoin forum!

The old forum can still be reached here:
http://bitcoin.sourceforge.net/boards/index.php

I'll repost some selected threads here and add updated answers to questions where I can.

FAQ
http://bitcoin.sourceforge.net/wiki/index.php?page=FAQ

Download
http://sourceforge.net/projects/bitcoin/files/

27

On Bitcoin Maturation

MATURATION is specific to bitcoins that have been newly created as rewards given to miners for their work on the block chain. Once a block has little or no chance of becoming an orphan block, the corresponding awarded bitcoins are mature enough to be safely credited to the miner.

Bitcoin Maturation?

Satoshi Nakamoto November 22, 2009, 06:31:44 PM

Bitcoin Maturation

Posted:Thu 01 of Oct, 2009 (14:12 UTC)

From the user's perspective the bitcoin maturation process can be broken down into 8 stages.

1. The initial network transaction that occurs when you first click Generate Coins.
2. The time between that initial network transaction and when the bitcoin entry is ready to appear in the All Transactions list.
3. The change of the bitcoin entry from outside the All Transaction field to inside it.
4. The time between when the bitcoin appears in the All Transfers list and when the Description is ready to change to Generated (50.00 matures in x more blocks).
5. The change of the Description to Generated (50.00 matures in x more blocks).
6. The time between when the Description says Generated (50.00 matures in x more blocks) to when it is ready to change to Generated.
7. The change of the Description to Generated.
8. The time after the Description has changed to Generated.

Which stages require network connectivity, significant local CPU usage and or significant remote CPU usage? Do any of these stages have names?

Re: Bitcoin Maturation?

Sirius-m October 22, 2009, 02:26 UTC

As far as I know, there's no network transaction when you click Generate Coins–your computer just starts calculating the next proof-of-work. The CPU usage is 100% when you're generating coins.

In this example, the network connection is used when you broadcast the information about the proof-of-work block

you've created (that which entitles you to the new coin). Generating coins successfully requires constant connectivity, so that you can start working on the next block when someone gets the current block before you

Bitcoin Maturation?

Satoshi Nakamoto November 22, 2009, 06:34:21 PM

It's important to have network connectivity while you're trying to generate a coin (block) and at the moment it is successfully generated.

1. During generation (when the status bar says "Generating" and you're using CPU to find a proof-of-work), you must constantly keep in contact with the network to receive the latest block. If your block does not link to the latest block, it may not be accepted.
2. When you successfully generate a block, it is immediately broadcast to the network. Other nodes must receive it and link to it for it to be accepted as the new latest block.

Think of it as a cooperative effort to make a chain. When you add a link, you must first find the current end of the chain. If you were to locate the last link, then go off for an hour and forge your link, come back and link it to the link that was the end an hour ago, others may have added several links since then and they're not going to want to use your link that now branches off the middle.

After a block is created, the maturation time of 120 blocks is to make absolutely sure the block is part of the main chain before it can be spent. Your node isn't doing anything with the block during that time, just waiting for other blocks to be added after yours. You don't have to be online during that time.

28

How Anonymous are Bitcoins?

UNLIKE A SUITCASE full of $100 bills, which can be moved without any trace, Bitcoin transactions are recorded in the public ledger. Although Bitcoin addresses are anonymous in nature, the transactions conducted in the names of these addresses are not.

How anonymous are bitcoins?

Satoshi Nakamoto November 25, 2009, 06:17:23 PM

Can nodes on the network tell from which and or to which bitcoin address coins are being sent? Do blocks contain a history of where bitcoins have been transfered to and from?

> Bitcoins are sent to and from bitcoin addresses, which are essentially random numbers with no identifying information.

When you send to an IP address, the transaction is still written to a bitcoin address. The IP address is only used to connect to the recipient's computer to request a fresh bitcoin address, give the transaction directly to the recipient and get a confirmation.

Blocks contain a history of the bitcoin addresses that a coin has been transferred to. If the identities of the people using the bitcoin addresses are not known and each address is used only once, then this information only reveals that some unknown person transferred some amount to someone else.

The possibility to be anonymous or pseudonymous relies on you not revealing any identifying information about yourself in connection with the bitcoin addresses you use. If you post your bitcoin address on the web, then you're associating that address and any transactions with it with the name you posted under. If you posted under a handle that you haven't associated with your real identity, then you're still pseudonymous.

For greater privacy, it's best to use bitcoin addresses only once. You can change addresses as often as you want using Options->Change Your Address. Transfers by IP address automatically use a new bitcoin address each time.

> Can nodes tell which bitcoin addresses belong to which IP addresses?

No.

> Is there a command line option to enable the sock proxy the first time that bitcoin starts?

In the next release (version 0.2), the command line to run it through a proxy from the first time is:
bitcoin -proxy=127.0.0.1:9050

The problem for TOR is that the IRC server which Bitcoin uses to initially discover other nodes bans the TOR exit nodes, as all IRC servers do. If you've already connected once before then you're already seeded, but for the first time, you'd need to provide the address of a node as such:
bitcoin -proxy=127.0.0.1:9050 -addnode=<someipaddress>

If someone running a node with a static IP address that can accept incoming connections could post their IP to use for -addnode, that would be great.

> What happens if you send bitcoins to an IP address that has multiple clients connected through network address translation (NAT)?

Whichever one you've set your NAT to forward port 8333 to will receive it. If your router can change the port number when it forwards, you could allow more than one client to receive. For instance, if port 8334 forwards to a computer's port 8333, then senders could send to "x.x.x.x:8334"

If your NAT can't translate port numbers, there currently isn't a command line option to change the incoming port that bitcoin binds to, but I'll look into it.

29

A Few Questions Answered by Satoshi

In THESE POSTS, Satoshi answers a wide variety of questions such as how anonymous Bitcoin is, the requirement for backups, and what happens in the case of lost coins. Another question asked was whether Bitcoin's being open source could pose a security problem as, for example a miner was changing the code. Satoshi replied that other miners would not accept it as it would be a deviation of the Bitcoin protocol.

Re: Questions about Bitcoin

Satoshi Nakamoto December 10, 2009 08:49:02 PM

SmokeTooMuch wrote:

Hi, yesterday i stumpled upon this great payment option.

I read my way trough many sites but now I have some questions that couldn't get answered.

1. Is Bitcoin really anonymous ? I mean totally and completely ? Is my ISP able to detect, that i have sent or received a Bitcoin payment ? Maybe he is even able to see that i am running Bitcoin right now ?
2. If i understood this correctly, my payment partners are not able to see who I am. Does this mean, he can not see my real IP adress ? Only the Bitcoin-adress ? Even if he monitors his network connections and stuff ?
3. If there is a way to tell that I am running Bitcoin for my ISP or a way to find out my IP for my payment partners, would it be more safe to tunnel the network traffic through a VPN (payed with Paysafecard for example). ? Could this be dangerous, because the VPN provider will be able to capture my payment ?
4. What files need to be backed up for not loosing my "money" ? Only the wallet.dat or the whole Bitcoin AppData directory ?
5. Isn`t it possible to multiply a wallet and use it on different machines ? This way you would double your money without doing anything for it.
   Are there security measures for this case ?
6. When someone loses his wallet, will there be a way to recreate the lost coins in the system ? Else the 21 million maxmimum will not be correct.
   (I mean not to recover the lost coins for one person, but if all the 21mio coins were created, and someone loses his wallet with 1mio coins, will the the others be able to create these 1mio coins now or are they totally lost for the bitcoin network ?)
7. I have read that there currently are about 130k blocks

out there. At my pc it only shows me about 24k. Is there something wrong or is this a normal behaviour ?

8. I`m afraid I didn`t understand everything about the bitcoin creation. How many coins are created by a machine in 24h in average ?
9. I know that port 8333 should be forwarded to the bitcoin-running machine. Now I ask myself if this goes for the TCP or the UDP.
   And is this port required for generating coins ? Or only for payment transactions ?
10. I`ve seen that the source code for bitcoin is open for everybody. Can this be an actual danger ? If the code is manipulated people can create more bitcoins than others, can`t they ? This would be a massive leak of security.
11. I`ve seen a formular to clalculate the coins that will be created in a certain amount of time. It had something to do with the maximum cpu speed and the availabe. Can`t find it anymore, so I`m asking you to explain me the coin creating. Do slow machines produce as much coins as high-end ones ?
12. Are there any other exchanging systems or potential payment partners except for new liberty standard ?
13. What happens when my system crashes ? Is the wallet saved automatically or only when bitcoin gets closed manually ? (Maybe even real-time saving when a coint is created or payment is made ?)
14. Is there a way to see how many bitcoins have beenm generated this far ? And how old is Bitcoin already ?

I know .... Many many questions but I am really interested in your service and want to know everything before i start using it more frequently.

(Sorry for my bad English . . .)

1–3: For that level of anonymity you need to connect through TOR, which will be possible with version 0.2, which is only a few weeks away. I'll post TOR instructions at that time.

4: Version 0.1.5: backup the whole %appdata%\Bitcoin directory. Version 0.2: you can backup just wallet.dat.

5: Nope. The whole design is all about preventing that from working.

6: Those coins can never be recovered, and the total circulation is less. Since the effective circulation is reduced, all the remaining coins are worth slightly more. It's the opposite of when a government prints money and the value of existing money goes down.

7: It's currently 29,296 blocks. The circulation is the number of blocks times 50, so the current circulation is 1,464,800 bc.

If you only have 24k blocks, it must not have finished the initial block download. Exit bitcoin and start it again. Version 0.2 is better/faster at the initial block download.

8: Typically a few hundred right now. It's easy now but it'll get harder as the network grows.

9: Good question, it's TCP. The website needs to be updated to say TCP port 8333.

The port forwarding is so other nodes can connect to you, so it helps you stay connected because you are able to be connected with more nodes. You also need it to receive payments by IP address.

10: No, the other nodes won't accept that.

Being open source means anyone can independently review the code. If it was closed source, nobody could verify the security. I think it's essential for a program of this nature to be open source.

11: Slower machines produce fewer coins. It's proportional to CPU speed.

12: There are more coming.

13: It uses a transactional database called Berkeley DB. It will not lose data in a system crash. Transactions are written to the database immediately when they're received.

14: For now, you can just multiply the total blocks by 50. The Bitcoin network has been running for almost a year now. The design and coding started in 2007.

Re: Questions about Bitcoin

SmokeTooMuch wrote:

Wow, thanks alot for these detailed answers.

But today another question came to my mind.

Lets say we know, that our neighbar uses Bitcoin, and we also know that he will receive a payment soon (maybe because he owns an internet shop and accepts bitcoin as payment option).

Also, we know that he uses WLAN and his network is unsecured or weak protected. Same goes for router configuration.

We now could log into his router configuration, change the ip adresses for the forwarded port 8333 to our system ip. Now every payment would be received by our bitcoin client.

Is this actually going to work ?

I know this is highly criminal and the scenario is .. well, lets call it "uncommon", but in theory it should work, right ? (Not that I have an interest in harming people, but I know that criminal people will try many ways to get some money.)

> BTW: same should work when you are on a LAN party with unprotected router config.
>
> Edit: Or are these scenarios totally impossible because no matter which ip adress uses the port, the payment will go to the bitcoin or ip adress that was defined from the payer ?

That's true, with the send-to-IP option, you are sending to whoever answers that IP. Sending to a bitcoin address doesn't have that problem.

The plan is to implement an IP + bitcoin address option that would have the benefits of both. It would still use a different address for each transaction, but the receiver would sign the one-time-use address with the given bitcoin address to prove it belongs to the intended receiver.

30

On "Natural Deflation"

THE TOPIC OF LOSING COINS has been covered a few times. They are referred to as "natural deflation." Here are two discussions relating to this issue. Note that national currencies today are born out of debt. When a loan is taken for a car or a house, the same number of dollars is created, and, once the loan is repaid, the currency disappears. A deflationary environment in our current system means that the value of assets (houses, cars, etc.) will decline, but, since loans have been taken out to purchase them, a cascade of bankruptcies will follow as people own more than they can purchase.

On the other hand, when a currency is intrinsically fixed in amount, loans are extremely rare. Before the creation of the Federal Reserve in the USA in 1913, the majority of purchases were done in cash, even for houses. The implication of a currency fixed in value, or even one

that gains in value, are important. People would not have to speculate in mutual funds for their retirement; instead one could simply save the money to make a purchase. This is typically called "hoarding" by the financial media, but so are retirement funds. Essentially, saving means you are delaying consumption of material, resources, and time so that others, including companies investing in new plants, can improve productivity now. Later you enjoy your retirement because of this delayed consumption. The concept of money is more abstract than most people think.

Re: A few suggestions

Satoshi Nakamoto December 13, 2009 04:51:25 PM

> The Madhatter wrote:
>
> > One quick question about "natural deflation" (as I call it). I have noticed that it is possible to spend to old addresses that no longer work. In essence the coins can not be claimed. Wouldn't there be a natural deflation effect because of this? I mean if the coins max out at 21,000,000 wouldn't the number of coins slowly work backwards due to payment errors?

There would be a command line switch at runtime to tell it to run without UI. All it needs to do is not create the main window. A simplistic way would be to disable "pframeMain->Show" and "ptaskbaricon->Show" in ui.cpp. The network threads don't care that the UI isn't there. The only other UI is a message box in CheckDiskSpace if it runs out of disk space.

Then a separate command line utility to communicate with it to do things. Not sure what it should be named.

"natural deflation"... I like that name for it. Yes, there will be natural deflation due to payment mistakes and lost data. Coin creation will eventually get slow enough that it is exceeded by natural deflation and we'll have net deflation.

The second conversation:

Re: Dying bitcoins

Satoshi Nakamoto June 21, 2010 05:48:26 PM

> Hello,
>
> if somebody's loosing his wallet (e.g. due to disk crash) he's not able to get back his coins, is he?
> So every time a person looses coins, they're lost forever? So the bitcoin network will slowly shrink over time? (Because there will always be people who loose wallets!)
>
> TIA
>
> virtualcoin

Lost coins only make everyone else's coins worth slightly more. Think of it as a donation to everyone.

> Quote from: laszlo on June 21, 2010, 01:54:29 PM
>
> I wonder though, is there a point where the difficulty of generating a new coinbase is so high that it would make more sense to try to recover keys for lost coins or steal other people's coins instead? The difficulty of that is really high so for now it makes a lot more sense to generate but I just wonder what the real figures are.. would that ever become more productive? Maybe Satoshi can address this..

Computers have to get about 2 ^ 200 times faster before that starts to be a problem. Someone with lots of compute power could make more money by generating than by trying to steal.

31

Bitcoin Version 0.2 is Here!

SATOSHI ANNOUNCES version 0.2 of Bitcoin.

Bitcoin version 0.2 is here!

Satoshi Nakamoto December 16, 2009 10:45:36 PM

Bitcoin version 0.2 is here!

Download links:
http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.2.0-win32-setup.exe/download
http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.2.0-win32.zip/download

http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.2.0-linux.tar.gz/download

New Features

- Martti Malmi
- Minimize to system tray option
- Autostart on boot option so you can keep it running in the background automatically
- New options dialog layout for future expansion
- Setup program for Windows
- Linux version (tested on Ubuntu)

Satoshi Nakamoto

- Multi-processor support for coin generation
- Proxy support for use with TOR
- Fixed some slowdowns in the initial block download

Major thanks to Martti Malmi (sirius-m) for all his coding work and for hosting the new site and this forum, and New Liberty Standard for his help with testing the Linux version.

32

Recommendation on Ways to Do a Payment For an Order

THERE ARE MULTIPLE TYPES of cryptographic algorithms used in asymmetric encryption. Here Satoshi's main point on the reasoning for using elliptic curve cryptography (EDCSA) instead of RSA is the size of the transaction (bytes). To make the size of each transaction as small as possible so that block size stays manageable, Satoshi decided to use EDCSA.

Re: A newb's test—anyone want to buy a picture for $1?

Satoshi Nakamoto Jnauary 29, 2010 12:22:13 PM

The recommended ways to do a payment for an order:

1. The merchant has a static IP, the customer sends to it with a comment.
2. The merchant creates a new bitcoin address, gives it to the customer, the customer sends to that address. This will be the standard way for website software to do it.

RSA vs ECDSA: it's not the size of the executable but the size of the data. I thought it would be impractical if the block chain, bitcoin addresses, disk space and bandwidth requirements were all an order of magnitude bigger. Also, even if using RSA for messages, it would still make sense to do all the bitcoin network with ECDSA and use RSA in parallel for only the message part. In that case, everything that's been implemented up to now would be implemented exactly as it has been.

We can figure out the best way to do this much later. It could use a separate (maybe existing) e-mail or IM infrastructure to pass messages, and instead of RSA, maybe just put a hash of the message in the transaction to prove that the transaction is for the order described in the message. The message would have to include a salt so nobody could brute force the hash to reveal a short message.

33

On the Proof-of-Work Difficulty

SATOSHI DISCUSSES The increasing proof-of-work difficulty as more miners begin to participate.

Proof-of-work difficulty increasing

Satoshi Nakamoto February 05, 2010 07:19:12 PM

We had our first automatic adjustment of the proof-of-work difficulty on 30 Dec 2009.

The minimum difficulty is 32 zero bits, so even if only one

person was running a node, the difficulty doesn't get any easier than that. For most of last year, we were hovering below the minimum. On 30 Dec we broke above it and the algorithm adjusted to more difficulty. It's been getting more difficult at each adjustment since then.

The adjustment on 04 Feb took it up from 1.34 times last year's difficulty to 1.82 times more difficult than last year. That means you generate only 55% as many coins for the same amount of work.

The difficulty adjusts proportionally to the total effort across the network. If the number of nodes doubles, the difficulty will also double, returning the total generated to the target rate.

For those technically inclined, the proof-of-work difficulty can be seen by searching on "target:" in debug.log. It's a 256-bit unsigned hex number, which the SHA-256 value has to be less than to successfully generate a block. It gets adjusted every 2016 blocks, typically two weeks. That's when it prints "GetNextWorkRequired RETARGET" in debug.log.

minimum 00000000ffff0000000000000000000000000000000000000000000000000000
30/12/2009 00000000d86a000000000000000000000000000000000000000000000000000000
11/01/2010 00000000c428000000000000000000000000000000000000000000000000000000
25/01/2010 00000000be710000000000000000000000000000000000000000000000000000
04/02/2010 000000008cc300000000000000000000000000000000000000000000000000000
14/02/2010 000000006546570000000000000000000000000000000000000000000000000000
24/02/2010 0000000043b3e500000000000000000000000000000000000000000000000000000
08/03/2010 00000000387f6f000000000000000000000000000000000000000000000000000000

21/03/2010 00000000381375000000000000000000000000000000000000000000000000
01/04/2010 000000002a11150000000000000000000000000000000000000000000000
12/04/2010 0000000020bca70000000000000000000000000000000000000000000000
21/04/2010 0000000016546f0000000000000000000000000000000000000000000000
04/05/2010 0000000013ec530000000000000000000000000000000000000000000000
19/05/2010 00000000159c240000000000000000000000000000000000000000000000
29/05/2010 000000000f67c00000000000000000000000000000000000000000000000
11/06/2010 000000000eba640000000000000000000000000000000000000000000000
24/06/2010 000000000d3142000000000000000000000000000000000000000000000
06/07/2010 000000000ae493000000000000000000000000000000000000000000000
13/07/2010 0000000005a3f400000000000000000000000000000000000000000000
16/07/2010 000000000168fd000000000000000000000000000000000000000000000
27/07/2010 00000000010c5a000000000000000000000000000000000000000000000
05/08/2010 0000000000ba180000000000000000000000000000000000000000000000
15/08/2010 0000000000800e000000000000000000000000000000000000000000000
26/08/2010 000000000069200000000000000000000000000000000000000000000000

date, difficulty factor, % change
2009 1.00
30/12/2009 1.18 +18%
11/01/2010 1.31 +11%

25/01/2010	1.34	+2%
04/02/2010	1.82	+36%
14/02/2010	2.53	+39%
24/02/2010	3.78	+49%
08/03/2010	4.53	+20%
21/03/2010	4.57	+9%
01/04/2010	6.09	+33%
12/04/2010	7.82	+28%
21/04/2010	11.46	+47%
04/05/2010	12.85	+12%
19/05/2010	11.85	-8%
29/05/2010	16.62	+40%
11/06/2010	17.38	+5%
24/06/2010	19.41	+12%
06/07/2010	23.50	+21%
13/07/2010	45.38	+93%
16/07/2010	181.54	+300%
27/07/2010	244.21	+35%
05/08/2010	352.17	+44%
15/08/2010	511.77	+45%
26/08/2010	623.39	+22%

34

On the Bitcoin Limit and Profitability of Nodes

THE ORIGINAL POSTS in this thread questioned the profitability of miners to mine when the difficulty level becomes high and the amount of bitcoin rewards decreases (it was 50 BTC at the time of these posts, but was reduced to 25 BTC later on in early 2013).

Re: Current Bitcoin economic model is unsustainable

Satoshi Nakamoto February 21, 2010 05:44:24 PM

xc wrote:

Nothing to sweat people. Nobody ever died of a 'deflationary spiral.' : -) I agree with "I-am-not-anonymous." The market will choose the best bitcoin-like currency. I happen to believe, however, that the rules that Satoshi has founded bitcoin on will be more than adequate for the future of a thriving bitcoin economy.

Everybody knows exactly how fast the supply of bitcoins will grow: it's set in stone in the rules of the programming and the bitcoin network. While it's true that there is not a currently existing fully-fleshed out market to truly price bitcoins, such markets and exchanges are being developed. As far as future would-be bitcoin generators are concerned, the question is not how much will he "demand….to compensate for his costs." The question he'll be asking himself is "given current market values and my ability to utilize electricity and CPU resources, is it worth it for me to generate bitcoins?" If the answer is yes, he participates. If it's no, he stops trying to mine for bitcoins and focuses on trading tangible assets with bitcoins serving as an appropriate intermediary. If he's not sure, he tries his hand at it for a while and then makes a final decision.

The number of nodes and associated computational cpu power will be in flux, and that competitive flux will allow for costs to approximate value (not the other way around.) Value being set by the markets and the demand for use of bitcoin as a trade intermediary (a money). In the far future, the competition of transaction costs will play a more important role for the would-be node operator.

Contrary to the paradox of thrift argument you present, collecting bitcoins and saving them with hopes of earning purchasing power through deflation is not a bad thing. It will allow for the pooling of bitcoin capital and make purchases of larger capital investments possible. In the future, there

> might even be bitcoin banks that lend out saved bitcoins with market-set interest rates, thereby diminishing the effects of hoarding. All this wonderful saving, however, comes at a price: delayed gratification of present desires. From the perspective of the would-be saver, the question will always be denying present desires to purchase real tangible assets now versus the future possibilities of purchasing more later. This time preference naturally varies with people and in different circumstances.
>
> Given the fact that bitcoins are by their electronic nature easily divisible, prices will be able to easily adjust to deflationary pressures. If too many are saving, prices will fall and the rate of interest will go down. This encourages demand (lower prices) and decreases the desire to save (less interest).
>
> XC

Excellent analysis, xc.

A rational market price for something that is expected to increase in value will already reflect the present value of the expected future increases. In your head, you do a probability estimate balancing the odds that it keeps increasing.

In the absence of a market to establish the price, NewLibertyStandard's estimate based on production cost is a good guess and a helpful service (thanks). The price of any commodity tends to gravitate toward the production cost. If the price is below cost, then production slows down. If the price is above cost, profit can be made by generating and selling more. At the same time, the increased production would increase the difficulty, pushing the cost of generating towards the price.

In later years, when new coin generation is a small percentage of the existing supply, market price will dictate the cost of production more than the other way around.

At the moment, generation effort is rapidly increasing, suggesting people are estimating the present value to be higher than the current cost of production.

35

On the Possibility of Bitcoin Address Collisions

BITCOIN ADDRESSES are created out of a hash of the public addresses, and concern was expressed about a possible collision, where two different individuals could by some random chance be assigned the same Bitcoin address. Note that a 160-bit hash yields 2 to the power of 160 or 1.46×10^{48} possibilities, and therefore the probability of a collision's occurring is extremely remote.

Re: Bitcoin Address Collisions

Satoshi Nakamoto February 23, 2010 09:22:47 AM

> NewLibertyStandard wrote:
>
> Although extremely unlikely, what would happen if two Bitcoin clients generated the same Bitcoin address? Would payments be delivered to whichever client encountered the payment first? If there is a mechanism in place to prevent such collisions, please explain it.

There's a separate public/private keypair for every bitcoin address. You don't have a single private key that unlocks everything. Bitcoin addresses are a 160-bit hash of the public key, everything else in the system is 256-bit.

If there was a collision, the collider could spend any money sent to that address. Just money sent to that address, not the whole wallet.

If you were to intentionally try to make a collision, it would currently take 2^126 times longer to generate a colliding bitcoin address than to generate a block. You could have got a lot more money by generating blocks.

The random seed is very thorough. On Windows, it uses all the performance monitor data that measures every bit of disk performance, network card metrics, cpu time, paging etc. since your computer started. Linux has a built-in entropy collector. Adding to that, every time you move your mouse inside the Bitcoin window you're generating entropy, and entropy is captured from the timing of disk ops.

36

QR Code

TWO CONVERSATIONS related to QR code for mobile phones came up. Based on an original suggestion from user *ec* on the forum, Satoshi suggested using QR code for the bitcoin address for payments at point of sale, a common practice today.

Re: URI-scheme for bitcoin

Satoshi Nakamoto February 24, 2010 05:57:43 AM

That would be nice at point-of-sale. The cash register displays a QR-code encoding a bitcoin address and amount on a screen and you photo it with your mobile.

https://bitcointalk.org/index.php?topic=177.msg1814#msg1814

Re: Bitcoin mobile

Satoshi Nakamoto February 24, 2010 05:57:43 AM

> Quote from: sirius-m on June 10, 2010, 01:51:16 PM
>
> You can of course use services like vekja.net or mybitcoin.com on a mobile browser, depositing money there to the extent you trust them.

I think that's the best option right now. Like cash, you don't keep your entire net worth in your pocket, just walking around money for incidental expenses.

They could make a smaller version of the site optimized for mobile. If there was an app, it could be a front end to one of those, with the main feature being QR-code reader, or maybe there's already a universal QR-code reading app that web sites can be designed to accept scans from.

If there was an iPhone app that was just a front end for vekja or mybitcoin, not a big involved P2P, would apple approve it and if not, on what basis? It could always be an Android app instead. An app is not really necessary though, just a mobile sized website.

A web interface to your own Bitcoin server at home wouldn't be a solution for everyone. Most users don't have a static IP, and it's too much trouble to set up port forwarding.

37

Bitcoin Icon/Logo

SATOSHI PRESENTS a logo/icon to use for Bitcoin and makes it copyright-free. This is no longer the logo being used by bitcoin.org. The current logo is this:

(See http://commons.wikimedia.org/wiki/File:Bitcoin.svg)

New icon/logo

Satoshi Nakamoto February 24, 2010 09:24:23 PM

New icons, what do you think? Better than the old one?

Full size 530x529 image for scaling down to custom sizes:
http://www.bitcoin.org/download/bitcoin530.png

The perspective shadow was too thick on the larger sizes. I updated 32, 48 and the full size.

I release these images into the public domain (copyright-free). I request that derivative works be made public domain.

> Quote from: Sabunir on February 25, 2010, 02:28:49AM
>
> Excellent. This would be a good resource for those participating in the banner contest. Why unequal dimensions?
>
> My only suggestion would be to make the coin's text stand out more. At tiny resolutions outlines tend to be come unworkable, so a better option may be to experiment with contrast. Making the text significantly darker than the rest of the coin would likely increase readability. Alternately, you could make the inner circle color darker, and the text lighter.

Good suggestion. I made the B slightly lighter and the background slightly darker. Very slightly. The foreground is now exactly the same colour as the BC in the old one.

It's kind of OK if you can't easily read the B in the 16x16. At that size, you just need to see that it's a coin. It doesn't matter so much what's embossed on it, just that there be some detail there because it wouldn't look like a coin if it was a blank smooth circle.

It's slightly wider than tall because the dark perspective under it goes more to the right than down.

I finished and posted the 32x31 and 48x47 versions in the first message. I like the 48 a lot.

How does everyone feel about the B symbol with the two lines through the outside? Can we live with that as our logo?

> Quote from: Cdecker on February 27, 2010, 03:24:07 AM
>
> How about an SVG version? That way we could automatically generate smaller and larger versions as needed.

I don't know how to do SVG, but I did the original very large, over 500 pixels across, so it can be scaled down. I'll give the original when I'm finished.

I had to custom tweak each icon size so the vertical lines land square on their pixels, otherwise they're ugly blurry and inconsistent. Such is the challenge of making icons. The original will be good for scaling to custom sizes between 48 and 500 but not smaller.

38

GPL License Versus MIT License

A SUGGESTION for creating a "we accept Bitcoin" logo had a GPL license. Here, Satoshi states that he prefers the MIT license, the same open source license that the Bitcoin software uses.

Re: Make your "we accept Bitcoin" logo

Satoshi Nakamoto February 24, 2010, 09:53:52 PM

If you GPL stuff, I have to avoid using it. Nothing against GPL per-se, but Bitcoin is an MIT license project. Anything GPL please clearly mark it as such.

39

On Money Transfer Regulations

In this post, Satoshi suggests a service wherein buyers and sellers of bitcoins could meet in person to complete the purchase/sale of bitcoins and thereby avoid any sort of regulations. Both parties would bring a device capable of Internet access or meet at a place having public-access computers (e.g., a library or Internet café). The buyer would presumably pay in cash and would provide the seller his/her address so that the transfer could be completed. A service allowing buyers and sellers to find each other does exist today (see for example *localbitcoins.com*).

Re: Money Transfer Regulations

Satoshi Nakamoto March 03, 2010, 04:28:56 AM

When there's enough scale, maybe there can be an exchange site that doesn't do transfers, just matches up buyers and sellers to exchange with each other directly, similar to how e-bay works.

To make it safer, the exchange site could act as an escrow for the bitcoin side of the payment. The seller puts the bitcoin payment in escrow, and the buyer sends the conventional payment directly to the seller. The exchange service doesn't handle any real world money.

This would be a step better than e-bay. E-bay manages to work fine even though shipped goods can't be recovered if payment falls through.

40

On the Possibility of a Cryptographic Weakness

SEVERAL THREADS covered different issues to which Satoshi suggested the same solution. Two of the threads below concern SHA-256, which is the cryptographic hash function used to create the "message digest" of the blocks used as the public ledger, each containing a set of bitcoin transactions. SHA-256 is used by the banking industry and other financial institutions. Were any weaknesses to one day be discovered in this encryption method, it would affect the whole financial industry, which would then be forced to change over to a new method. Satoshi suggests the same policy for Bitcoin.

The second thread was in regard to the discovery of a major cryptographic weakness. At first, Satoshi refers to his earlier post on SHA-256

Collisions, but user *llama* specifies the case where a major weakness is discovered in the elliptic curve cryptographic code that is used for the Bitcoin private key.

Re: Dealing with SHA-256 Collisions

Satoshi Nakamoto June 14, 2010, 08:39:50 AM

> Quote from: lachesis on June 14, 2010, 01:01:11 AM
>
> A mathematician friend of mine pointed out that there are very few if any hash protocols that have survived for 10 years or more. What would Bitcoin's solution be if SHA256 were to be cracked tomorrow?
>
> SHA-256 is very strong. It's not like the incremental step from MD5 to SHA1. It can last several decades unless there's some massive breakthrough attack.

If SHA-256 became completely broken, I think we could come to some agreement about what the honest block chain was before the trouble started, lock that in and continue from there with a new hash function.

If the hash breakdown came gradually, we could transition to a new hash in an orderly way. The software would be programmed to start using a new hash after a certain block number. Everyone would have to upgrade by that time. The software could save the new hash of all the old blocks to make sure a different block with the same old hash can't be used.

Re: Major Meltdown

Satoshi Nakamoto July 10, 2010, 04:26:01 PM

> Quote from: llama on July 01, 2010, 10:21:47 PM
>
> Satoshi, That would indeed be a solution if SHA was broken (certainly the more likely meltdown), because we could still recognize valid money owners by their signature (their private key would still be secure).
>
> However, if something happened and the signatures were compromised (perhaps integer factorization is solved, quantum computers?), then even agreeing upon the last valid block would be worthless.

True, if it happened suddenly. If it happens gradually, we can still transition to something stronger. When you run the upgraded software for the first time, it would re-sign all your money with the new stronger signature algorithm. (by creating a transaction sending the money to yourself with the stronger sig)

Re: Hash() function not secure

Satoshi Nakamoto July 16, 2010, 04:13:53 PM

SHA256 is not like the step from 128 bit to 160 bit.

To use an analogy, it's more like the step from 32-bit to 64-bit address space. We quickly ran out of address space with 16-bit computers, we ran out of address space with 32-bit computers at 4GB, that doesn't mean we're going to run out again with 64-bit anytime soon.

SHA256 is not going to be broken by Moore's law computational improvements in our lifetimes. If it's going to get broken, it'll be

by some breakthrough cracking method. An attack that could so thoroughly vanquish SHA256 to bring it within computationally tractable range has a good chance of clobbering SHA512 too.

If we see a weakness in SHA256 coming gradually, we can transition to a new hash function after a certain block number. Everyone would have to upgrade their software by that block number. The new software would keep a new hash of all the old blocks to make sure they're not replaced with another block with the same old hash.

41

On a Variety of Transaction Types

THIS POST is a bit more technical than other posts presented herein. However, I chose to include it because of it is useful in explaining why Satoshi's first implementation of the core design supported a variety of possible transaction types so as to avoid future major modifications.

Re: Transactions and Scripts: DUP HASH160…EQUALVERIFY CHECKSIG

Satoshi Nakamoto June 17, 2010, 06:46:08 PM

Quote from: Gavin Andresen June 17, 2010, 11:38:31 AM

So I'm writing a little tool that dissects the Bitcoin wallet.dat, mainly because I want to understand better exactly how Bitcoin works.

And I see that the outputs of transactions have a value (number of bitcoins) and a bunch of bytes that are run through the little Forth-like scripting language built in to bitcoin. E.g.: ['TxOut: value: 100.00 Script: DUP HASH160 6fad...ab90 EQUALVERIFY CHECKSIG']

First: it make me a little nervous that bitcoin has a scripting language in it, even though it is a really simple scripting language (no loops, no pointers, nothing but math and crypto). It makes me nervous because it is more complicated, and complication is the enemy of security. It also makes it harder to create a second, compatible implementation. But I think I can get over that.

Looking at the code, new transactions are verified by pushing the signature an then public key on the interpreter's stack and then running the TxOut script (did I get that right?).

Could I write code to create transactions with any valid script in the TxOut?
E.g. could I create a TxOut with a script of: OP_2DROP OP_TRUE . . . to create a coin that could be spent by anybody?

And is flexibility in the types of coins created the reason it is coded this way?

The nature of Bitcoin is such that once version 0.1 was released, the core design was set in stone for the rest of its lifetime. Because of that, I wanted to design it to support every possible transaction type I could think of. The problem was, each thing required special support code and data fields whether it was used or not, and only covered one special case at a time. It would have been an explosion of special cases. The solution was script, which generalizes the problem so transacting parties can describe their transaction as a predicate that the node network evaluates. The

nodes only need to understand the transaction to the extent of evaluating whether the sender's conditions are met.

The script is actually a predicate. It's just an equation that evaluates to true or false. Predicate is a long and unfamiliar word so I called it script.

The receiver of a payment does a template match on the script. Currently, receivers only accept two templates: direct payment and bitcoin address. Future versions can add templates for more transaction types and nodes running that version or higher will be able to receive them. All versions of nodes in the network can verify and process any new transactions into blocks, even though they may not know how to read them.

The design supports a tremendous variety of possible transaction types that I designed years ago. Escrow transactions, bonded contracts, third party arbitration, multi-party signature, etc. If Bitcoin catches on in a big way, these are things we'll want to explore in the future, but they all had to be designed at the beginning to make sure they would be possible later.

I don't believe a second, compatible implementation of Bitcoin will ever be a good idea. So much of the design depends on all nodes getting exactly identical results in lockstep that a second implementation would be a menace to the network. The MIT license is compatible with all other licenses and commercial uses, so there is no need to rewrite it from a licensing standpoint.

A second version would be a massive development and maintenance hassle for me. It's hard enough maintaining backward compatibility while upgrading the network without a second version locking things in. If the second version screwed up, the user experience would reflect badly on both, although it would at least reinforce to users the importance of staying with the official version. If someone was getting ready to fork a second version, I would have to air a lot of disclaimers about the risks of using a minority version. This is a design where the majority version wins

if there's any disagreement, and that can be pretty ugly for the minority version and I'd rather not go into it, and I don't have to as long as there's only one version.

I know, most developers don't like their software forked, but I have real technical reasons in this case.

> Quote from: gavinandresen on June 17, 2010, 07:58:14 PM
>
> I admire the flexibility of the scripts-in-a-transaction scheme, but my evil little mind immediately starts to think of ways I might abuse it. I could encode all sorts of interesting information in the TxOut script, and if non-hacked clients validated-and-then-ignored those transactions it would be a useful covert broadcast communication channel.
>
> That's a cool feature until it gets popular and somebody decides it would be fun to flood the payment network with millions of transactions to transfer the latest Lady Gaga video to all their friends . . .

That's one of the reasons for transaction fees. There are other things we can do if necessary.

> Quote from: laszlo on June 17, 2010, 06:50:31 PM
>
> How long have you been working on this design Satoshi? It seems very well thought out, not the kind of thing you just sit down and code up without doing a lot of brainstorming and discussion on it first. Everyone has the obvious questions looking for holes in it but it is holding up well : -)
>
> Since 2007. At some point I became convinced there was a way to do this without any trust required at all and couldn't resist to keep thinking about it. Much more of the work was designing than coding.
>
> Fortunately, so far all the issues raised have been things I previously considered and planned for.

42

First Bitcoin Faucet

GAVIN ANDRESEN, currently Lead Core Bitcoin Developer, announced that he has written a "Bitcoin faucet" giving away 5 bitcoins per customer for free. Satoshi replies that he has had the same idea in mind if nobody else came up with it.

Re: Get 5 free bitcoins from freebitcoins.appspot.com

Satoshi Nakamoto June 18, 2010, 11:08:34 PM

> Quote from: Gavin Andresen on June 11, 2010, 05:38:45PM
>
> For my first Bitcoin coding project, I decided to do something that sounds really dumb: I created a web site that gives away

> Bitcoins. It is at: https://freebitcoins.appspot.com/
>
> Five ฿ per customer, first come first served, I've stocked it with ฿1,100 to start. I'll add more once I'm sure it is working properly.
>
> Why? Because I want the Bitcoin project to succeed, and I think it is more likely to be a success if people can get a handful of coins to try it out. It can be frustrating to wait until your node generates some coins (and that will get more frustrating in the future), and buying Bitcoins is still a little bit clunky.
>
> Please try it out and get some free coins, even if you already have more Bitcoins than you know what to do with. You can get some and then donate them right back; the address is: ~~15VjRaDX9zpbA8LVnbrCAFzrVzN7ixHNsC~~

> Excellent choice of a first project, nice work. I had planned to do this exact thing if someone else didn't do it, so when it gets too hard for mortals to generate 50BTC, new users could get some coins to play with right away. Donations should be able to keep it filled. The display showing the balance in the dispenser encourages people to top it up.
>
> You should put a donation bitcoin address on the page for those who want to add funds to it, which ideally should update to a new address whenever it receives something.

Later, as the value was going up. Satoshi suggest reducing the bitcoin faucet to 1 BTC (1 bitcoin)

Re: Donations to freebitcoins.appspot.com needed!

Satoshi Nakamoto July 16, 2010, 02:02:07 AM

> Quote from: Gavin Andresen on June 12, 2010, 07:15:46PM
>
> The Bitcoin Faucet is handling the slashdotting really well... except that I'm running out of coins to give away. over 5,000 have flowed out of the Faucet since I refilled it last night.
>
> Any of you early adopters who generated tens of thousands of coins back in the early days, are you willing to send a few to the Faucet to be given away so more people can try out Bitcoin? I know that most of them are likely to be lost (I suspect there a lot of slashdot lookey-loos who won't stick around long enough to spend their 5 bitcoins), but if that's the case then that'll just increase the value of your other bitcoins, anyway...
>
> Fountain donation address is: ~~15VjRaDX9zpbA8LVnbrCAFzrVzN7ixHNsC~~
>
> Depending on donations and how long the slashdotting lasts, I might have to start giving away bitnickels...

5 BTC seems like a lot these days, maybe the normal amount should be 1 or 2 BTC.

This is an important service so new users can at least get something if generating is too hard.

43

Bitcoin 0.3 Released!

SATOSHI is not only technical in terms of what this new release offered, but he also gives this sales and marketing pitch: *"Escape the arbitrary inflation risk of centrally managed currencies! Bitcoin's total circulation is limited to 21 million coins."*

Bitcoin 0.3 released!

Satoshi Nakamoto June 06, 2010, 06:32:35 PM

Announcing version 0.3 of Bitcoin, the P2P cryptocurrency! Bitcoin is a digital currency using cryptography and a distributed network to replace the need for a trusted central server. Escape the arbitrary inflation risk of centrally managed currencies! Bitcoin's total circulation is limited to 21 million coins. The coins are gradually released to the network's nodes based on the CPU

power they contribute, so you can get a share of them by contributing your idle CPU time.

What's new:

- Command line and JSON-RPC control
- Includes a daemon version without GUI
- Transaction filter tabs
- 20% faster hashing
- Hashmeter performance display
- Mac OS X version (thanks to Laszlo)
- German, Dutch and Italian translations (thanks to DataWraith, Xunie and Joozero)

Get it at http://www.bitcoin.org or read the forum to find out more.

44

On the Segmentation or "Internet Kill Switch"

TWO THREADS involved the possibility of segmentation or a split of the network.

Re: Anonymity!

Satoshi Nakamoto June 08, 2010, 07:12:00 PM

It's hard to imagine the Internet getting segmented airtight. It would have to be a country deliberately and totally cutting itself off from the rest of the world.

Any node with access to both sides would automatically flow the block chain over, such as someone getting around the blockade with a dial-up modem or sat-phone. It would only take one node to do it. Anyone who wants to keep doing business would be motivated.

If the network is segmented and then recombines, any transactions in the shorter fork that were not also in the longer fork are released into the transaction pool again and are eligible to get into future blocks. Their number of confirmations would start over.

If anyone took advantage of the segmentation to double-spend, such that there are different spends of the same money on each side, then the double-spends in the shorter fork lose out and go to 0/unconfirmed and stay that way.

It wouldn't be easy to take advantage of the segmentation to double-spend. If it's impossible to communicate from one side to the other, how are you going to put a spend on each side? If there is a way, then probably someone else is also using it to flow the block chain over.

You would usually know whether you're in the smaller segment. For example, if your country cuts itself off from the rest of the world, the rest of the world is the larger segment. If you're in the smaller segment, you should assume nothing is confirmed.

This covers specifically the case of a network split.

What happens when network is split for prolonged time and reconnected?

Posted by em3rgentOrdr on August 01, 2010, 11:07:24 AM

Suppose that BitCoins are being widely used all across the globe. Suppose that all internet connections between two

countries are blocked (eg China and US go to war) and people still engage in transactions inside each network. Now all transactions within each network are broadcasted to all nodes inside its network, but not to the other network. Within each network, the longest chain in each would be considered valid, and the BitCoin economy would continue to exist inside each network.

Now after several years existing independently, what happens when the two networks are reconnected?

Re: What happens when network is split for prolonged time and reconnected?

Posted by kiba on August 02, 2010, 03:19:08 AM

Maybe they won't be reconnected. Instead, we will effectively have two currencies. This will lead to the creation of an Eastern-Western bitcoin currency exchange market(s).

Re: What happens when network is split for prolonged time and reconnected?

Posted by throughput on August 02, 2010, 06:07:08 PM

I, as a merchant, will only care about whether my network is a majority network, so after a reconnect my transactions will be accepted. So it will be enough for me to be able to monitor the current number of distinct nodes. Put that into a graph and stop processing transactions if that number suddenly halves. It may be a service on a web-server running a Bitcoin node.

But is there a way to monitor that number at all? If not, it

would be wise to add some feature to the standard, which will allow to determine in real time what is the number of distinct nodes running.

Re: What happens when network is split for prolonged time and reconnected?

Posted by creighto on August 03, 2010, 08:01:22 PM

> Quote from: throughput on August 03, 2010, 01:33:08 PM
>
> Yes...
> But what you describe is only possible after someone have noticed and prooved the network split is happening.
> Do you propose any method to detect the beginning of the network split?

I started another thread along this line elsewhere, but for an individual vendor, a simple watchdog daemon that tracks the average time between blocks since the last official change in difficulty and alerts the vendor if a single block takes more than twice as long as the average, perhaps suspending the acceptance of new coins until the vendor checks to see what is happening. Each block in a row that takes longer than the average increases confidence against a false positive. So if one block takes twice as long as average, followed by a series of blocks that take 75% longer than average, then you can be fairly certain that you are no longer on the majority network.

Re: What happens when network is split for prolonged time and reconnected?

Posted by satoshi on August 03, 2010, 10:45:07 PM

creighto: I agree with that idea. After a few hours, it should be possible for the client to notice if the flow of blocks has dropped off by more than would be likely just by chance. It could tell if it's not hearing the hum of the world anymore.

> Quote from: knightmb on August 03, 2010, 07:02:13 PM
>
> > Quote from: gavinandresen on August 03, 2010, 06:38:44 PM
> >
> > Or if the split lasted long enough (more than 100 blocks), transactions that involve generated coins on the shorter chain would be invalid at the merge.
>
> Interesting info, so other than some double-spending issues, as long as the block chain isn't separated for more than 100 or so blocks (or 16+ hours),

In practice, splits are likely to be very asymmetrical. It would be hard to split the world down the middle. More likely it would be a single country vs the rest of the world, lets say a 1:10 split. In that case, it would take the minority fork 10 times as long to generate 100 blocks, so about 7 days. Also it would be super easy for the client to realize it's hearing way too few blocks and something must be wrong.

> Quote from: knightmb on August 03, 2010, 07:02:13 PM
>
> If there a hard coded limit on split delay? Meaning if I had a small network split from the public network, spent some coin around, came back a few days later and got them sync up to the public network (other than coin generation if it happened) transactions should be fine?

There's no time limit. Assuming you weren't spending coins generated in the minority fork, or spending someone's double-spends you received, your transactions can get into the other chain at any time later.

45

On Cornering the Market

SATOSHI REPLIES to a comment about someone's trying to buy up all of the bitcoins and references the Hunt brothers and the silver market of the late 1970s. Note that the Hunt brothers' share of buying was actually a small percentage of the silver market. What doomed them was their trading on COMEX from a leveraged position on the future exchange. COMEX changed the rules by placing a cap on the total amount of contracts one could have, thus forcing anyone having more than the specified limit into a selling position, and so forced the Hunt brothers to liquidate. See the detailed writing on this subject by Mike Maloney at *WealthCycles.com*:

http://wealthcycles.com/features/the-hunt-brothers-capped-the-price-of-gold-not-50-silver

Re: BTC Vulnerability? (Massive Attack against BTC system. Is it really?)

Satoshi Nakamoto July 09, 2010, 03:28:46 PM

> Quote from: user on July 07, 2010, 06:15:28 PM
>
> Hi. (I'm sorry if I don't understand any concept).
> What you think if anyone intruder will buy up bitcoin currency and erase all binary data. This way can destroy bitcoin systems. Is btc network protected against that attack?

What the OP described is called "cornering the market". When someone tries to buy all the world's supply of a scarce asset, the more they buy the higher the price goes. At some point, it gets too expensive for them to buy any more. It's great for the people who owned it beforehand because they get to sell it to the corner at crazy high prices. As the price keeps going up and up, some people keep holding out for yet higher prices and refuse to sell.

The Hunt brothers famously bankrupted themselves trying to corner the silver market in 1979:
"Brothers Nelson Bunker Hunt and Herbert Hunt attempted to corner the world silver markets in the late 1970s and early 1980s, at one stage holding the rights to more than half of the world's deliverable silver.[1] During Hunt's accumulation of the precious metal silver prices rose from $11 an ounce in September 1979 to nearly $50 an ounce in January 1980.[2] Silver prices ultimately collapsed to below $11 an ounce two months later,[2] much of the fall on a single day now known as Silver Thursday, due to changes made to exchange rules regarding the purchase of commodities on margin.[3]"

http://en.wikipedia.org/wiki/Cornering_the_market

46

On Scalability and Lightweight Clients

As time passes, the block chain, which contains a record of all transactions made since January 2009, grows continuously. Given a Bitcoin wallet containing several Bitcoin addresses along with their corresponding private keys and bitcoin balances, Bitcoin must know which address or addresses should be used to cover a transaction. For example, assume that Bitcoin address A has 0.1 BTC, B has 0.2 BTC, and C has 0.3 BTC and that a payment of 0.5 BTC is to be made. The Bitcoin wallet will have to pick a combination of 2 or more Bitcoin addresses with which to cover the 0.5 BTC as none of them individually has sufficient bitcoins to make the entire payment. Unless a Bitcoin client has the full block chain and so is able to know the current balance of each address, it must interface with a server that does have the block chain. The Simplified Payment Verification,

first described in Satoshi's original Bitcoin paper, allows clients to rely on a server participating in the Bitcoin network that has the full block chain but may or may not be participating in the mining process. It was implemented later and benefits lightweight clients.

Re: Scalability

Satoshi Nakamoto July 14, 2010, 09:10:52 PM

> Quote from: jib on July 12, 2010, 11:36:17 AM
>
> Am I correct in understanding that every node receives information about every transaction (as the technical paper says)? Doesn't that make bitcoin completely impractical for use as a currency on a large scale?

The design outlines a lightweight client that does not need the full block chain. In the design PDF it's called Simplified Payment Verification. The lightweight client can send and receive transactions, it just can't generate blocks. It does not need to trust a node to verify payments, it can still verify them itself.

The lightweight client is not implemented yet, but the plan is to implement it when it's needed. For now, everyone just runs a full network node.

I anticipate there will never be more than 100K nodes, probably less. It will reach an equilibrium where it's not worth it for more nodes to join in. The rest will be lightweight clients, which could be millions.

At equilibrium size, many nodes will be server farms with one or two network nodes that feed the rest of the farm over a LAN.

47

On Fast Transaction Problems

HERE, Satoshi explains that a payment processing company would monitor the Bitcoin network for the transaction of interest to the merchant, as well as any other conflicting transactions. Since nodes will only accept the first transactions and will reject any other transactions which conflict with those, the merchant's transaction should be seen first. If any conflicting transactions are seen by the payment processing company, it will inform the merchant that the transaction is bad. Of course, if the correct transaction is officially accepted, the merchant can reimburse the client or process the sale.

Re: Bitcoin snack machine (fast transaction problems)

Satoshi Nakamoto July 17, 2010, 10:29:13 PM

> Quote from: Insti, July 17, 2010, 02:33:41 AM
>
> How would a Bitcoin snack machine work?
>
> 1. You want to walk up to the machine. Send it a bitcoin.
> 2. ?
> 3. Walk away eating your nice sugary snack. (Profit!)
>
> You don't want to have to wait an hour for you transaction to be confirmed.
> The vending machine company doesn't want to give away lots of free candy.
>
> How does step 2 work?

I believe it'll be possible for a payment processing company to provide as a service the rapid distribution of transactions with good-enough checking in something like 10 seconds or less.

The network nodes only accept the first version of a transaction they receive to incorporate into the block they're trying to generate. When you broadcast a transaction, if someone else broadcasts a double-spend at the same time, it's a race to propagate to the most nodes first. If one has a slight head start, it'll geometrically spread through the network faster and get most of the nodes.

A rough back-of-the-envelope example:

1	0
4	1
16	4
64	16
80%	20%

So if a double-spend has to wait even a second, it has a huge disadvantage.

The payment processor has connections with many nodes. When it gets a transaction, it blasts it out, and at the same time monitors the network for double-spends. If it receives a double-spend on any of its many listening nodes, then it alerts that the transaction is bad. A double-spent transaction wouldn't get very far without one of the listeners hearing it. The double-spender would have to wait until the listening phase is over, but by then, the payment processor's broadcast has reached most nodes, or is so far ahead in propagating that the double-spender has no hope of grabbing a significant percentage of the remaining nodes.

Another later thread revisited the scalability and transaction rate. Satoshi points back to the thread above.

Re: Scalability and transaction rate

Satoshi Nakamoto July 29, 2010, 02:00:38 AM

> Quote from: Red, July 22, 2010, 05:17:28 AM
>
> I'm curious about the developers feelings on scalability. For example, could the system handle a million users, doing say 5 transactions each per day. 5 million transactions per day is roughly 35,000 transactions per 10 minute period?
>
> Is there a bottle neck in propagating 35,000 transactions to a million nodes for block generation? Or has that issue been designed for?

The current system where every user is a network node is not the intended configuration for large scale. That would be like every Usenet user runs their own NNTP server. The design supports letting users just be users. The more burden it is to run a node,

the fewer nodes there will be. Those few nodes will be big server farms. The rest will be client nodes that only do transactions and don't generate.

> Quote from: bytemaster on July 28, 2010, 08:59:42 PM
>
> Besides, 10 minutes is too long to verify that payment is good. It needs to be as fast as swiping a credit card is today.

See the snack machine thread, I outline how a payment processor could verify payments well enough, actually really well (much lower fraud rate than credit cards), in something like 10 seconds or less. If you don't believe me or don't get it, I don't have time to try to convince you, sorry.
http://bitcointalk.org/index.php?topic=423.msg3819#msg3819

48

Wikipedia Article Entry on Bitcoin

We CANNOT IMAGINE that Wikipedia would consider deleting the entry on Bitcoin with its current level of interest. At the time of this post, Bitcoin was still under $1, but was generating sufficient interest to justify an article in Wikipedia. Satoshi comments here that he considers the timing strange, as coverage of Bitcoin was rapidly increasing in the media.

Re: They want to delete the Wikipedia article

Satoshi Nakamoto July 20, 2010, 06:38:28 PM

Quote from: Giulio Prisco July 14, 2010, 07:21:08 AM

http://en.wikipedia.org/wiki/Bitcoin

This article is being considered for deletion in accordance with Wikipedia's deletion policy.
Please share your thoughts on the matter at this article's entry on the Articles for deletion page.

This article needs references that appear in reliable third-party publications. Primary sources or sources affiliated with the subject are generally not sufficient for a Wikipedia article.
Please add more appropriate citations from reliable sources.

The recent Slashdot article should be considered as a reliable reference:
http://news.slashdot.org/story/10/07/11/1747245/Bitcoin-Releases-Version-03

I cannot edit at this moment, can you guys save the WP artcile?

Bitcoin is an implementation of Wei Dai's b-money proposal http://weidai.com/bmoney.txt on Cypherpunks http://en.wikipedia.org/wiki/Cypherpunks in 1998 and Nick Szabo's Bitgold proposal http://unenumerated.blogspot.com/2005/12/bit-gold.html

The timing is strange, just as we are getting a rapid increase in 3rd party coverage after getting slashdotted. I hope there's not a big hurry to wrap the discussion and decide. How long does Wikipedia typically leave a question like that open for comment?

It would help to condense the article and make it less promotional sounding as soon as possible. Just letting people know what it is, where it fits into the electronic money space, not trying to convince them that it's good. They probably want something that just generally identifies what it is, not tries to explain all about how it works.

If you post in http://en.wikipedia.org/wiki/Wikipedia:Articles_for_deletion/Bitcoin please don't say "yeah, but bitcoin is really

important and special so the rules shouldn't apply" or argue that the rule is dumb or unfair. That only makes it worse. Try to address how the rule is satisfied.

Search "bitcoin" on google and see if you can find more big references in addition to the infoworld and slashdot ones. There may be very recent stuff being written by reporters who heard about it from the slashdot article.

I hope it doesn't get deleted. If it does, it'll be hard to overcome the presumption. Institutional momentum is to stick with the last decision. (edit: or at least I assume so, that's how the world usually works, but maybe Wiki is different)

And later, on July 31st, the article was officially deleted, and then later restored.

Re: BitCoin Wikipedia page DELETED!!!

Posted by em3rgentOrdr, July 31, 2010, 02:17:41 AM

---

from http://en.wikipedia.org/wiki/Bitcoin

"This page has been deleted. The deletion and move log for the page are provided below for reference.

10:42, 30 July 2010 Polargeo (talk | contribs) deleted "Bitcoin" ↱ (Wikipedia:Articles for deletion/Bitcoin)"

Re: BitCoin Wikipedia page DELETED!!!

Posted by sirius, September 30, 2010, 04:45:26 PM

Can we just make different language versions of a deleted

page without getting them removed? Let's do it if we can. I can write a version in Finnish.

Re: BitCoin Wikipedia page DELETED!!!

Posted by satoshi, September 30, 2010, 05:50:32 PM

If you do, I think it should be a very brief, single paragraph article like 100 words or less that simply identifies what Bitcoin is.

I wish rather than deleting the article, they put a length restriction. If something is not famous enough, there could at least be a stub article identifying what it is. I often come across annoying red links of things that Wiki ought to at least have heard of.

The article could be as simple as something like:

"Bitcoin is a peer-to-peer decentralised /link/electronic currency/link/."

The more standard Wiki thing to do is that we should have a paragraph in one of the more general categories that we are an instance of, like Electronic Currency or Electronic Cash. We can probably establish a paragraph there. Again, keep it short. Just identifying what it is.

Re: BitCoin Wikipedia page DELETED!!!

Posted by ribuck, December 13, 2010, 11:23:41 AM

It looks like the article will be restored. But one point that keeps being raised is that many of the article's references are to pages in this forum. If anyone can replace a forum reference with a reference to a page that has no perceived conflict of interest, that would help.

49

On the Possibility of Stealing Coins

As stated before, Bitcoin uses asymmetric cryptography with a public and private key pair as a mechanism to receive and authorize spending of bitcoins. However, Satoshi decided to use as the Bitcoin address the hash of the public key rather than the public key itself. Satoshi did this for two reasons. One was to reduce the size of each transaction as the hash is only 160 bits long. The second benefit was that it conveniently added one more layer of security in case a "backdoor" or security flaw should one day be discovered in the asymmetric cryptography algorithm used by Bitcoin. To be able to spend bitcoins, a hacker would have to first derive the public key from the hash and then derive the private key from the public key. *Bitcoin Magazine* wrote an excellent article on this subject.[2]

This whole thread discusses the possibility that an attacker with a lot of computing power could spend the bitcoins stored in a bitcoin

[2] http://bitcoinmagazine.com/7781/satoshis-genius-unexpected-ways-in-which-bitcoin-dodged-some-cryptographic-bullet/

address. Since the bitcoin block chain is an open ledger, it can be inspected in order to identify a bitcoin address having a large balance, and so an attacker could focus on those addresses.

Satoshi concluded that this would be quite difficult, as it would require a brute force attack to find a public key with a matching hash. It also shows the value of open source code (code that is open for all to see) for security, as opposed to closed source.

The important parts of the threads, including Satoshi's entire posting, are reproduced here:

Stealing Coins

Posted by Red, July 25, 2010, 05:08:03 PM

I think there is a pretty significant crypto flaw in Bitcoin as currently implemented. I'm not sure it is exploitable now (I'm not a real cryptohacker) but it is more than plausible that will be in the near future.

The flaw would enable anonymous stealing of coins from arbitrary bitcoin addresses. And no it doesn't involve solving any of the hard problems that keep existing crypto systems secure. It is simply a *potential* correctable logic flaw in the implementation.

I would like bitcoins to succeed, so I'd rather not jump up and down in public yelling about flaws in public. Is there an appropriate place to discuss these types of issues?

Re: Stealing Coins

Posted by Satoshi, July 25, 2010, 05:45:22 PM

It's best if you tell it to me privately so it can be fixed first.

I just e-mailed you my e-mail address. (or you could PM me here)

Re: Stealing Coins

Posted by Satoshi, July 25, 2010, 07:06:23 PM

Red, thanks for telling me privately first! Please go ahead and post it (and relieve the suspense for everyone!)

His point is that transactions paid to a Bitcoin Address are only as secure as the hash function. To make Bitcoin Addresses short, they are a hash of the public key, not the public key itself. An attacker would only have to break the hash function, not ECDSA.

Re: Stealing Coins

Posted by Red, July 25, 2010, 07:09:43 PM

Thanks Satoshi,

Here is what I sent him.

———

Public key cryptography depends on the fact that it is hard to factor large prime numbers. Everyone knows that. If bitcoins were transfers were assigned to a well formed public key, and an associated private key signature was required for future

transfer I would concede that bitcoins crypto transfers were completely secure.

However, bitcoin transactions don't seem to work that way (by my reading). Transactions assign coin amounts to a particular "bitcoin address". Where the address is a hash of the public key.

To validate a transaction, nodes take the public key from the signature and use that to verify the actual signature. If the signature is valid, it then hashes the public key to confirm it matches the bitcoin address assigned in the previous transaction. If both match, by definition, the transaction is good.

The potential weakness is in associating the public key in the signature with the bitcoin address.

There is a many to one relationship between public keys and a given hash. Now, if finding a pair of prime numbers that creates a secure public/private key pair where the public key part hashes to a particular bitcoin address seems hard... it probably is.

However, that is not required.

All you need is ANYTHING representing a public key that hash collides with a know large bitcoin account. It does NOT have to be a secure key pair based on primes. It is simply has to work once and allow the transfer of the stolen money to another account. That is potentially much easier.

Some hashes are harder to collide than others. I'm not sure the strength of the hash being used. However, colliding any hash gets much easier if you don't have to care about the content being hashed.

Because of the nature of public keys they look like random data. As I understand them, you can't know if a public key is

based upon secure math unless you succeed in factoring it. Therefore clients don't try. They normally just do the validation of the signature and presume the public key was generated in a secure fashion if it worked.

NOTE: The following analysis needs double checking by a real cryptohacker. IANACR

So depending on the hash, you could use one of the up-and-coming hash collision algorithms to generate a colliding block of data which represents a public key. Then by reversing the public/private key math, generate an associated (but hardly secure at all) private key that would generate valid signatures.

You then take your insecure, easily factorable, key pair and generate a signed transaction that matches the target bitcoin address.

Since the transaction log, can't validate the full public key the coins were intended for, it simple presumes it must have been the one presented.

By recording the full public key of the transfer target in the block list you can regain the intended strength. However, you lose the ability to pass around 34 character addresses.

If I'm off base, I apologize for wasting your time.

Cheers!
Red

Re: Stealing Coins

Posted by Red, July 25, 2010, 07:22:14 PM

Satoshi pointed out that my scenario still required the hash function to be broken. That is true, but I was surprised to learn

how successful some have been with that. MD4 and MD5 are obvious examples. But work is well underway at colliding SHA-1 and siblings like SHA-256.

What hash is being used in this part of Bitcoin?

He is also skeptical that you could you could use something other than a generated keypair.

On this point, I'm pretty confident that it is a simple matter of mathematics. I didn't pay enough attention to this until I learned about "blind signing" of documents.

It turns out you can take a document and multiply it by a random number. Then have someone sign the jumbled file. Finally, you divide your random number out of their signature and the result is still a valid signature for the original document. Who'd figured that would work!

Anyway, if keypairs are only secure if they are based upon pairs of primes. Then nothing changes any of the math if the numbers are not prime. They are just much easier to factor.

I'd be perfectly happy for some crypto guy to prove me an idiot. It effects some features of a previous project I created that relied on the same association. I didn't think of this then either.

Re: Stealing Coins

Posted by knightmb, July 25, 2010, 07:34:42 PM

Very nice. *another reason why I love open source*

As I understand it then, and please correct me if I'm wrong

Since the hash of the public key is smaller than the actual

public key itself, one need only find a collision that matches the hash and when that collision is found you'll know the public/private key combo. Then you simply spend coin using the known ones and the other clients will think it's a valid transfer because the clients are only concerned that your hash matches the hash of the victim and the transaction is recorded for all time.

Currently the hash is 35 characters long, alpha-numeric
26 (upper case) +26 (lower case) +10 (numbers) = 62
possible per character

So we have 541,638,008,296,341,754,635,824,011,3
76,225,346,986,572,413,939,634,062,667,808,768
possible combinations.

So I think we have about half of much work to do compared to going brute force against the main private/public key. Never hurts to plan for the future : -)

Re: Stealing Coins

Posted by knightmb, July 25, 2010, 07:44:02 PM

> Quote from: Red on July 25, 2010, 07:22:14 PM
>
> Satoshi pointed out that my scenario still required the hash function to be broken. That is true, but I was surprised to learn how successful some have been with that. MD4 and MD5 are obvious examples. But work is well underway at colliding SHA-1 and siblings like SHA-256.

What they often don't mention though is *collision generating* still takes a lot of CPU time.

If I figure out that Public Key 123456 generates Hash ABCD

and Public Key 654321 also generates Hash ABCD

I'm still left without the Private Key.

But from what you are saying, all I need is Public Key 654321 and I can spend coin pretending to be Public Key 123456.

Re: Stealing Coins

Posted by Red, July 25, 2010, 07:52:23 PM

From what I was told, bitcoin is using one of the 160 bit hashes for generating bitcoin address.

The SHA-1 family of hash algorithms are some of the most commonly used. SHA-1 is a 160 bit hash.

Here is a paper that claims to find SHA-1 collisions in 2^52 crypto operations. And optimally secure hash would take 2^80 operations. 2^52 time is still large, but it is getting into cluster and botnet range.

http://www.ictlex.net/wp-content/iacrhash.pdf

The MD5 hashes can already be crashed in seconds on laptops. That was why it was retired from certificate based signatures.

And yes what I'm saying is **I think** you can think of a public key as two secret numbers mathematically combined together. And the private key as those two numbers kept separately. The thing that make the system secure requires that the two secret numbers be really large prime numbers.

But if they are really large non-prime numbers the combination math still works, it is just must faster to break the algorithm.

I'll do a little more googling and see if I can substantiate my claims. I was hoping someone could dismiss them out of hand though.

Re: Stealing Coins

Posted by Satoshi, July 25, 2010, 08:01:40 PM

> Quote from: knightmb on July 25, 2010, 07:44:02 PM
>
> If I figure out that Public Key 123456 generates Hash ABCD
> and
> Public Key 654321 also generates Hash ABCD
>
> *I'm still left without the Private Key.*
>
> But from what you are saying, all I need is Public Key 654321 and I can spend coin pretending to be Public Key 123456.

You would still have to sign it with public key 654321. You need to find a collision using a public key for which you know the private key.

When you claim a Bitcoin Address transaction, you give your public key that matches the hash, then you must sign it with that key.

Red's point is that it's easy to quickly generate insecure public keys which you could break and find the private key after you find a collision.

He points out that if the public key was required to be a secure one, one which must have required significant work to find the prime numbers, that would increase the strength above that of the hash function alone. Someone trying to brute force would have to take time generating a key for each attempt.

Re: Stealing Coins

Posted by knightmb, July 25, 2010, 08:20:41 PM

> Quote from: satoshi on July 25, 2010, 08:01:40 PM
>
> You would still have to sign it with public key 654321. You need to find a collision using a public key for which you know the private key.
>
> When you claim a Bitcoin Address transaction, you give your public key that matches the hash, then you must sign it with that key.
>
> Red's point is that it's easy to quickly generate insecure public keys which you could break and find the private key after you find a collision.
>
> He points out that if the public key was required to be a secure one, one which must have required significant work to find the prime numbers, that would increase the strength above that of the hash function alone. Someone trying to brute force would have to take time generating a key for each attempt.

Yeah, I thought the private key had to be in the mix somewhere. It kind of adds another randomness though, you have to find the hash that collides with another public key and at the same time, the private key has to be weak enough to break. I'm not saying it's impossible, but it introduces 2 variables into the reverse collision finding.

Basically, one would build a rainbow table of weak private keys and then have to compare those to public hashes and then have to hope that someone out there has a hash that happens to be a part of that attack. Not impossible of course, but how feasible even if computers were 100 times faster in 10 years?

[edit] ok, re-read what you wrote, the public key is generated

from the private key, not independently. So just finding a weak public key is the issue.

Re: Stealing Coins

Posted by satoshi, July 25, 2010, 08:48:01 PM

> Quote
>
> Here is a paper that claims to find SHA-1 collisions in 2^52 crypto operations. And optimally secure hash would take 2^80 operations. 2^52 time is still large, but it is getting into cluster and botnet range.

---

2 ^ 80 is if you can use a birthday attack. You can't use a birthday attack for this, so the difficulty is the full 2 ^ 160 bits. Although, if you were trying to crack any one of 1 million (2 ^ 20) transactions, you could do a partial birthday attack 2 ^ 160/2 ^ 20 = 2 ^ 140

Bitcoin Addresses are the only place where 160-bit hash is used. Everything else is SHA-256. They're calculated as:

bitcoinaddress = RIPEMD-160(SHA-256(publickey))

Correct me if I'm wrong (please, and I'll gladly eat crow) but I think it would be hard to use an analytical attack on RIPEMD-160 in this case. An analytical attack prescribes a certain range or pattern of inputs to try that will greatly increase your chance of finding a collision. Here, you don't have that kind of control over RIPEMD-160's input, because the input is the output of SHA-256. If an analytical attack helps you find an input to RIPEMD-160 that produces a collision, what are you going to do with it? You still have to get SHA-256 to output that value, so you would still have to break SHA-256 too.

For brute force, RIPEMD-160(SHA-256(x)) is no stronger than

RIPEMD-160 alone. But for analytical attack, it seems like you must analytical attack both RIPEMD-160 and SHA-256. If I'm wrong, then the strength is the same as RIPEMD-160 and the SHA-256 only serves as one round of key strengthening.

Re: Stealing Coins

Posted by Red, July 25, 2010, 09:04:01 PM

> Quote from: satoshi on July 25, 2010, 08:48:01 PM
>
> bitcoinaddress = RIPEMD-160(SHA-256(publickey))
>
> Correct me if I'm wrong (please, and I'll gladly eat crow) but I think it would be hard to use an analytical attack on RIPEMD-160 in this case.

I think you are correct on the analytical attack. At least a far as I understand (minimally) the mathematical genius that is analyzing them.

I was worried it was the simpler:

bitcoinaddress = RIPEMD-160(publickey)

Re: Stealing Coins

Posted by Red, July 25, 2010, 09:19:11 PM

So the way I read it.

Given two numbers p and q. Which for RSA are supposed to be large primes.

Then n = p*q

The public key is the two fields (n, e). e is called the public exponent and appears to be chosen from a set of common values.

The private key is also two fields (n, d). d is called the private exponent it it is derived by knowing e, p-1, and q-1.

The trick is, it is really hard to factor n into p & q. Therefore it is equally as hard to find p-1 and q-1

My postulation is that if n is arbitrary, and e is one of the common values, then there are lots of different p, q pairs that would work. The less prime the numbers the easier to find p and q, and therefore p-1 and q-1. And if you have a big block of arbitrary data that give you lots of flexibility in trying to collide a hash.

(That is the point where I could be totally off base though. Really interested, if a crypto geek knows better than me.)

I did read that the key generation algorithms create p and q such that they are "very likely prime" but it is too much work to know for sure. This leads me to believe non-primes don't cause any obvious FAILs. I could be wrong though.

Re: Stealing Coins

Posted by satoshi, July 25, 2010, 10:27:36 PM

Sorry, actually it's ECDSA (Elliptic Curve Digital Signature Algorithm) not RSA. I shouldn't have said "prime numbers". ECDSA doesn't take much time to generate a keypair.

Re: Stealing Coins

Posted by Red, July 26, 2010, 12:46:04 PM

> Quote from: satoshi on July 25, 2010, 10:27:36 PM
>
> Sorry, actually it's ECDSA (Elliptic Curve Digital Signature Algorithm) not RSA. I shouldn't have said "prime numbers". ECDSA doesn't take much time to generate a keypair.
>
> I'll learn how elliptic curves work one day, but not today. I should have taken more finite math when I was I college. Who'd a thought it would have come in handy for anything!
>
> By the way, nice idea and implementation of BitCoin Satoshi!
>
> It opens a whole new world of possibilities. I particularly like the concept of distributed agreement without relying upon trust. I think that is the breakthrough concept.
>
> Also, I think the idea of BitCoin mining was brilliant! I doubt you could have gotten the network bootstrapped any other way. I disagree that it's a "fair way" to distribute coins, but hey the world is not fair! And really, I don't think any other way would have generated as much user excitement.
>
> By the way, I concede that there is no thread of stealing bitcoins from my earlier postulation. The double hash seems to assure that from my perspective. Nice call!
>
> Incidentally, I'd still like to know what happens if you generate RSA keys based upon non-prime numbers though. I figure there are other systems out there that didn't double hash. :-)

Re: Stealing Coins

Posted by Bitcoiner, July 27, 2010, 02:01:16 AM

I'm glad that there's guys like Red out there keeping a sharp eye out on things! This thread also makes me appreciative of open source software, since there's so many smart and interested people on this forums that can validate the software and place an additional degree of trust in it. Not sure that Bitcoin could be too successful if it was closed source!

Re: Stealing Coins

Posted by bytemaster, July 28, 2010, 09:42:17 PM

It would seem to me that the obvious solution to minimize the risk of any potential attack is to make the potential "reward" small. Thus never keep too many coins in one address. If the economic value of the "prize" is less than the cost of breaking it then no one will bother trying. After saying that, I still think that it is best to keep things as hard as possible to crack.

Re: Stealing Coins

Posted by knightmb, July 28, 2010, 10:45:16 PM

It would certainly be hard by both luck and CPU/storage power to do this.

If you found a collision and a private key, that would do you no good since you would have to peg an account out of the 541,638,008,296,341,754,635,824,011,376,225,346,986,572,413,939,634,062,667,808,768 possible combinations of people using accounts.

So look at it two-fold. I find a collision in the hash and I find the private key. Now I have to hope my odds are that someone else is using that hash. Since there are more possible hash account numbers than every person every born on this planet and was each using a million addresses, the attack by it's own nature, while interesting, just isn't really feasible on a large scale.

50

Major Flaw Discovered

A MAJOR FLAW was discovered in the Bitcoin software/protocol that allowed a sender to send invalid transactions, one where the sender creates new bitcoins. By the time it was fixed, several millions of invalid bitcoins had been created. They were later erased from the block chain.

*** ALERT *** Upgrade to 0.3.6

Posted by satoshi, July 29, 2010, 07:13:06 PM

Please upgrade to 0.3.6 ASAP! We fixed an implementation bug where it was possible that bogus transactions could be displayed as accepted. Do not accept Bitcoin transactions as payment until you

upgrade to version 0.3.6!

If you can't upgrade to 0.3.6 right away, it's best to shut down your Bitcoin node until you do.

Also in 0.3.6, faster hashing:

- midstate cache optimisation thanks to tcatm
- Crypto++ ASM SHA-256 thanks to BlackEye

Total generating speedup 2.4x faster.

Download:
http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.3.6/

Windows and Linux users: if you got 0.3.5 you still need to upgrade to 0.3.6.

51

On Flood Attack Prevention

THE CONCERN RAISED HERE is the equivalent of a denial of service attack on the Bitcoin network where one entity could send millions of transactions, each transferring a small amount, 1 satoshi (0.00000001 BTC) for instance. This thread is more technical than some others, and not all of the posts have been recopied here only those relevant to the topic and those about concerns addressed by Satoshi.

Flood attack 0.00000001 BC

Posted by Mionione, July 12, 2010, 12:04:24 PM

hi, what would happen if someone sends millions of 0.00000001 BC to millions of address please ?

=> all of the networks peers must store all transactions ?
=> are each 0.00000001 owner/hash stocked in blocks on all peers?

i don't really understand how bitcoin handle fractions of bc

Re: Flood attack 0.00000001 BC

Posted by Gavin Andresen, July 12, 2010, 12:08:45 PM

From the source code:

main.h: // To limit dust spam, require a 0.01 fee if any output is less than 0.01

Re: Flood attack 0.00000001 BC

Posted by llama, July 12, 2010, 02:23:46 PM

Hmm, I didn't realize that was in there, and I really don't like that approach.

That pretty much ruins the possibility of using bitcoin for true micropayments. Wouldn't it be better for clients to just ignore a spammy IP? Sure an attacker could get more, but he couldn't get millions.

Re: Flood attack 0.00000001 BC

Posted by Gavin Andresen, July 12, 2010, 02:45:54 PM

But how would you distinguish between a legitimate micropayment-processing IP and a spammy "I want to make Bitcoin use so much bandwidth nobody is willing to run it any more" IP?

Really small micropayments seem to me to be a really hard problem, and I don't think Bitcoin should try to solve too many very hard problems all at once.

Re: Flood attack 0.00000001 BC

Posted by Gavin Andresen, July 12, 2010, 02:45:54 PM

But how would you distinguish between a legitimate micropayment-processing IP and a spammy "I want to make Bitcoin use so much bandwidth nobody is willing to run it any more" IP?

Re: Flood attack 0.00000001 BC

Posted by Insti, August 04, 2010, 02:58:31 PM

What exactly is this 'dust spam' that this 0.01BTC transaction fee "solving"?
It seems to do more harm than good because it prevents micropayment implementations such as the one bytemaster is suggesting.

I'm not aware that the network is straining under the weight of the existing transaction volume.
Anyone wishing to send a lot of transactions can already do this by sending x BTC to themselves a lot.

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 04, 2010, 04:25:36 PM

> Quote from: Insti on August 04, 2010, 02:58:31 PM
>
> It seems to do more harm than good because it prevents micropayment implementations such as the one bytemaster is suggesting.

Bitcoin isn't currently practical for very small micropayments. Not for things like pay per search or per page view without an aggregating mechanism, not things needing to pay less than 0.01. The dust spam limit is a first try at intentionally trying to prevent overly small micropayments like that.

Bitcoin is practical for smaller transactions than are practical with existing payment methods. Small enough to include what you might call the top of the micropayment range. But it doesn't claim to be practical for arbitrarily small micropayments.

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 05, 2010, 04:03:21 PM

Forgot to add the good part about micropayments. While I don't think Bitcoin is practical for smaller micropayments right now, it will eventually be as storage and bandwidth costs continue to fall. If Bitcoin catches on on a big scale, it may already be the case by that time. Another way they can become more practical is if I implement client-only mode and the number of network nodes consolidates into a smaller number of professional server farms. Whatever size micropayments you need will eventually be practical. I think in 5 or 10 years, the bandwidth and storage will seem trivial.

I am not claiming that the network is impervious to DoS attack. I think most P2P networks can be DoS attacked in numerous ways. (On a side note, I read that the record companies would like to DoS all the file sharing networks, but they don't want to break the anti-hacking/anti-abuse laws.)

If we started getting DoS attacked with loads of wasted transactions back and forth, you would need to start paying a 0.01 minimum transaction fee. 0.1.5 actually had an option to set that, but I took it out to reduce confusion. Free transactions are nice and we can keep it that way if people don't abuse them.

That brings up the question: if there was a minimum 0.01 fee for each transaction, should we automatically add the fee if it's just the minimum 0.01? It would be awfully annoying to ask each time. If you have 50.00 and send 10.00, the recipient would get 10.00 and you'd have 39.99 left. I think it should just add it automatically. It's trivial compared to the fees many other types of services add automatically.

> Quote from: FreeMoney on August 04, 2010, 07:30:32 PM
>
> Does including more slow down your hashing rate?
>
> No, not at all.

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 05, 2010, 04:30:20 PM

> Quote from: bytemaster
>
> Payments would generally be advanced, say 1 BTC at a time and when the connection closes any "change" would be returned. This rule makes it impossible to pay for a simple "search query" with no further transactions.

One alternative is to use a round-up system. You pay for, say, 1000 pages or images or downloads or searches or whatever at a time. When you've used up your 1000 pages, you pay for another 1000 pages. If you only use 1 page, then you have 999 left that you may never use, but it's not a big deal because the cost per 1000 is still small.

Or you could pay per day. The first time you access the site on a given day, you pay for 24 hours of access.

Per 1000 or per day may be easier for consumers to get their heads around too. They worry about per item because it's harder to figure if it might add up too fast. Unlimited for 24 hours they know what the cost will be. Or if 1000 seems like plenty, they're not worrying that it's costing more with each click if they figure 1000 is more than they'll probably use.

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 05, 2010, 04:39:58 PM

> Quote from: bytemaster on August 05, 2010, 03:39:19 PM
>
> The only solution to this problem is to make broadcasting of a transaction "non free". Namely, if you want me to include it you have to pay me. The net (no pun intended) result is that each client would need to pay other clients to whom they even send their transaction, not just the individual who gets it in a block. In this way the laws of economics take over and no one gets a free ride on the transaction broadcast system.
>
> I don't know a way to implement that. The transaction fee to the block creator uses a special trick to include the transaction fee without any additional size. If there was a transaction for each transaction fee, then what about the transactions fees for the transaction fee's transaction?

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 05, 2010, 05:49:43 PM

> Quote from: bytemaster on August 05, 2010, 04:46:52 PM
>
> Right now the transaction fee address is left "blank" and the block generator fills it out.
> Now you would fill it in with the address of the person you are asking to build the block.

If you're only going to have one person work on building the block, that could take days. Oh, do you mean send a different variation to each node with the tx fee written to them?

The way it is now, it's whoever builds this gets it.

If we needed to, we could have a BitTorrent-esque tit-for-tat for transaction broadcast. Relay paying transactions to me, or I won't relay them to you. It probably won't be an actual problem though. It only takes one node relaying like it should to cancel out 7 others greedily not relaying.

Re: Flood attack 0.00000001 BC

Posted by satoshi, August 11, 2010, 11:28:50 PM

It would be nice to keep the blk*.dat files small as long as we can.

The eventual solution will be to not care how big it gets.

But for now, while it's still small, it's nice to keep it small so new users can get going faster. When I eventually implement client-only mode, that won't matter much anymore.

There's more work to do on transaction fees. In the event of a flood, you would still be able to jump the queue and get your

transactions into the next block by paying a 0.01 transaction fee. However, I haven't had time yet to add that option to the UI.

Scale or not, the test network will react in the same ways, but with much less wasted bandwidth and annoyance.

52

Drainage of Bitcoin Faucet

As THE VALUE of bitcoins increased, the Bitcoin faucet (see previous reference) was becoming more attractive. Gavin Andresen reports that the value of a bitcoin has increased by a factor of 10 since he created the Faucet.

Who's the Spanish jerk draining the Faucet?

Posted by Gavin Andresen, August 04, 2010, 08:40:55 PM

I just shut down freebitcoins.appspot.com; it looks like somebody in Spain is being a jerk and getting a new IP address, bitcoin address, and solving the captcha. Over and

over and over again:

> Code:
>
> 79.154.133.217 - - [04/Aug/2010:12:46:55 -0700]
> "POST / HTTP/1.1" 200 1294 "https://freebitcoins.appspot.
> com/"
> "Opera/9.80 (Windows NT 6.0; U; es-LA) Presto/2.6.30
> Version/10.60,gzip(gfe)"
>
> 79.146.112.13 - - [04/Aug/2010:12:45:20 -0700]
> "POST / HTTP/1.1" 200 1294 "https://freebitcoins.appspot.
> com/"
> "Opera/9.80 (Windows NT 6.0; U; es-LA) Presto/2.6.30
> Version/10.60,gzip(gfe)"
>
> 81.44.159.81 - - [04/Aug/2010:12:42:20 -0700]
> "POST / HTTP/1.1" 200 1294 "https://freebitcoins.appspot.
> com/"
> "Opera/9.80 (Windows NT 6.0; U; es-LA) Presto/2.6.30
> Version/10.60,gzip(gfe)"

Those IP addresses all map to Telefonica de Espana. If it was you: give them back, please: 15VjRaDX9zpbA8LVnbrCAFzrVzN7ixHNsC

Now that 5 bitcoins is worth a fair bit, I'm thinking I need more cheating countermeasures. I can think of four things to try:

1. Rate limit based on the first byte of the IP address (79. or 81. in this case).
2. Rate limit based on the USER-AGENT string ("Opera/9.8..." in this case).
3. Rate limit based on last two domains of reverse DNS lookup of the IP address (rima-tde.net in this case).

4. Make the standard amount given away 0.5 Bitcoins (Bitcoins have gone up 10 times in value since I started the Faucet).

If you get rate limited, you'll get a message that asks you to try again tomorrow.

BitcoinFX: thanks again for the donation to the faucet; I'm going to drain the Faucet below 500 coins temporarily, and will refill it with your donation after the new cheating countermeasures are in place.

Re: Who's the Spanish jerk draining the Faucet?

Posted by satoshi, August 04, 2010, 08:40:55 PM

Silently failing would look bad.

> Quote from: gavinandresen on August 04, 2010, 08:40:55 PM
>
> 1. Rate limit based on the first byte of the IP address (79. or 81. in this case).

Definitely needed. What rate are you thinking of? Ultimately, it's better to rate limit it than to let it all drain out.

> Quote from: gavinandresen on August 04, 2010, 08:40:55 PM
>
> 3. Rate limit based on last two domains of reverse DNS lookup of the IP address (rima-tde.net in this case).

That might work surprisingly well. If it works, it keeps them from hitting the rate limit, but the rate limit is there as the last line of defence.

> Quote from: gavinandresen on August 04, 2010, 08:40:55 PM
>
> 4. Make the standard amount given away 0.5 Bitcoins (Bitcoins have gone up 10 times in value since I started the Faucet).

Definitely time to lower it.

53

Transaction to IP Address Rather than Bitcoin Address

IN THE BEGINNING, the ability to send to an IP address rather than (or perhaps in addition to) a Bitcoin address was considered.

BITCOIND TRANSACTION TO IP ADDRESS

Posted by lfm, August 05, 2010, 02:22:14 PM

I cant figure out how to send a transaction to an ip address from bitcoind command line interface. Has the function been implemented yet? (linux 64 if it matters)

Re: bitcoind transaction to ip address

Posted by satoshi, August 05, 2010, 05:28:40 PM

It's not implemented.

It turned out nobody liked that mode of transfer anyway, so it hasn't had much development attention.

54

On Escrow and Multi-Signature Transactions

TRANSACTIONS requiring multiple signatures are built in to the Bitcoin protocol and can be used by escrow services. For example, three keys are involved, but only two of these are required to sign the transaction. In such case, one key is owned by the payer, the second by the payee, and the third by the escrow agent. When there are no disputes or conflicts, the payer and the payee sign the transaction so that the payee can receive the funds.

If there is a dispute, the escrow agent reviews the dispute and, after deciding for either the payer or the payee, signs the transaction over to whichever party the escrow agent has decided in favor of. This is analogous to a bank check that requires two signatures from any of three

persons, in this case the payer, payee, and the escrow agent. Escrow services for Bitcoin transactions do exist today. The following three threads contain discussions related to how escrow could be handled and the implications of escrow to Bitcoin.

A PROPOSAL FOR A SEMI-AUTOMATED ESCROW MECHANISM

Posted by Olipro, July 30, 2010, 07:29:08 PM

So, the basic escrow works by two people working through a third party to exchange (usually money) for some other form of goods or services.

In a transaction where both people are honest, the escrow business can essentially be automatic since the buyer gets his goods and approves release of funds, only when there is a dispute does human interaction become necessary. Therefore, I propose the following system:

1) you create an escrow transaction for the amount, authorised by your key and containing the recipient's key/ data etc - this block cannot be claimed until a subsequent block is issued by the buyer to approve it, it's also impossible for the buyer to reclaim it without the seller approving it to be returned.

2) it enters the network, gets verified and the seller sends the goods, once the buyer gets them, he creates a release transaction and the seller gets his bitcoins.

3) if a dispute occurs and both parties are refusing to release the money one way or the other, clearly it's now necessary to get a third party to arbitrate - in this situation, a signature from both the buyer and seller authorising a third party is required

which will give that third party ownership of the original escrow transaction and they can then arbitrate the matter.

Re: A proposal for a semi-automated Escrow mechanism

Posted by satoshi, August 05, 2010, 06:08:30 PM

A transaction can be written that requires two signatures to spend it next. You write a payment that requires the signature of both the recipient and the sender to spend it. To release the escrow, you give the recipient the signature for your half, or the payee can return it by giving you his signed half. There's no mediator in this simple case. The recourse is to refuse to ever release it, essentially burning the money.

Re: A proposal for a semi-automated Escrow mechanism

Posted by satoshi, August 07, 2010, 08:04:59 PM

> Quote from: jgarzik on August 05, 2010, 07:00:30 PM
>
> Due to that recourse, it is unlikely to be used as an escrow mechanism : -)

Really? Do you think people won't be able to understand the benefit? (If your response is an argument that there's no benefit at all, I guess that will reinforce the case that people won't be able to understand it.)

Here, Satoshi creates a specific thread regarding escrow handling.

Escrow

Posted by satoshi, August 07, 2010, 08:13:52 PM

Here's an outline of the kind of escrow transaction that's possible in software. This is not implemented and I probably won't have time to implement it soon, but just to let you know what's possible.

The basic escrow: The buyer commits a payment to escrow. The seller receives a transaction with the money in escrow, but he can't spend it until the buyer unlocks it. The buyer can release the payment at any time after that, which could be never. This does not allow the buyer to take the money back, but it does give him the option to burn the money out of spite by never releasing it. The seller has the option to release the money back to the buyer.

While this system does not guarantee the parties against loss, it takes the profit out of cheating.

If the seller doesn't send the goods, he doesn't get paid. The buyer would still be out the money, but at least the seller has no monetary motivation to stiff him.

The buyer can't benefit by failing to pay. He can't get the escrow money back. He can't fail to pay due to lack of funds. The seller can see that the funds are committed to his key and can't be sent to anyone else.

Now, an economist would say that a fraudulent seller could start negotiating, such as "release the money and I'll give you half of it back", but at that point, there would be so little trust and so much spite that negotiation is unlikely. Why on earth would the fraudster keep his word and send you half if he's already breaking his word to steal it? I think for modest amounts, almost everyone would refuse on principle alone.

Re: Escrow

Posted by jgarzik, August 07, 2010, 09:25:40 PM

Buyer not having recourse except burning the money will limit the utility, I think.

Re: Escrow

Posted by aceat64, August 08, 2010, 02:55:59 AM

> Quote from: jgarzik on August 07, 2010, 09:25:40 PM
>
> Buyer not having recourse except burning the money will limit the utility, I think.

Perhaps we could work in a way to do arbitration. If both the buyer and seller agree, the money can be diverted to a 3rd party. That person could then arbitrate and either return the money to the buyer, give it to seller or steal it (obviously you'd want to choose a trustworthy arbitrator).

Re: Escrow

Posted by jgarzik, August 08, 2010, 03:58:03 AM

> Quote from: aceat64 on August 08, 2010, 02:55:59 AM
>
> > Quote from: jgarzik on August 07, 2010, 09:25:40 PM
> >
> > Buyer not having recourse except burning the money will limit the utility, I think.

> Perhaps we could work in a way to do arbitration. If both the buyer and seller agree, the money can be diverted to a 3rd party. That person could then arbitrate and either return the money to the buyer, give it to seller or steal it (obviously you'd want to choose a trustworthy arbitrator).

That's how online escrow operates today. Buyer and seller agree to let a 3rd party physically hold the money. Buyer and seller both agree to rules that the neutral 3rd party will follow, for transaction resolution / redemption. The neutral third party is the one who disburses funds to one party or the other.

This is a pretty decent overview: https://www.escrow.com/solutions/escrow/process.asp

Some people might choose to use the bitcoin-specific signed escrow method... but I think the "burn the money" recourse serves as a incentive to *avoid* bitcoin escrow entirely, rather than an incentive to use bitcoin escrow honestly.

Re: Escrow

Posted by aceat64, August 08, 2010, 05:49:44 AM

I like Olipro's suggestion is this thread: http://bitcointalk.org/index.php?topic=645.0

The buyer and seller both an equal amount of bitcoins into escrow and the seller can't retrieve both sets until the buyer signs off on it. Optionally if both parties agree the funds are returned to their original owners or both sets are transfered to an agreed upon arbitrator. I deviate from his suggestion that the arbitrator only have control over the buyers half, I think they should have control of both so that both parties still have a bitcoin stake in the issue.

Re: Escrow

Posted by jgarzik, August 10, 2010, 06:53:57 PM

> Quote from: nimnul on August 10, 2010, 05:51:49 PM
>
> The Satoshi solution is good, because if customer can take money back, it will be a big problem to sellers. See current situation with internet credit card payments and chargebacks. Chargebacks are major PITA for sellers, bitcoin must avoid that at all cost : -)

Ask some real-world business owners if they want to tell their customers about the chance of the money being lost forever, unrecoverable by either party.

Re: Escrow

Posted by nelisky, August 10, 2010, 08:20:36 PM

Regardless of what the technical options are, I think that an escrow will always need to be, by definition, a trusted entity. I can see the automated workflow being easy enough when things go well:

- Buyer sends btc to escrow, stating the recipient address
- Seller sees btc in escrow, marked to send to his address
- Buyer can release funds to seller
- Escrow will automatically do so after x days
- Both parties can open a complaint

And that's all I would automate. When things go bad, both parties should have a fee to pay to the escrow (that fee may be paid in advance to open account there?) so everyone looses something. Then the escrow will just have to mediate.

> Because there's a fee *and* a human intermediary, the chances of successful fraud will probably not be economically interesting in the long run. Someone already trusted would make the ideal person for this, and maybe for a small fee some of us 'common guys' could help assert allegations from either side, if we are local to them.
>
> But the money burning solution, while great at preventing economically viable fraud, does nothing to prevent revenge and actually makes everyone loose if one side is dishonest. I would certainly not endorse that.

Re: Escrow

Posted by satoshi, August 11, 2010, 01:30:02 AM

> Quote from: jgarzik on August 10, 2010, 06:53:57 PM
>
> Ask some real-world business owners if they want to tell their customers about the chance of the money being lost forever, unrecoverable by either party.

That makes it sound like it might somehow get lost and the parties can't get it even if they want to cooperate.

When you pay for something up front, you can't get it back either. Consumers seem comfortable with that. It's no worse than that.

Either party always has the option to release it to the other.

> Quote from: nelisky on August 10, 2010, 08:20:36 PM
>
> But the money burning solution, while great at preventing economically viable fraud, does nothing to prevent revenge and actually makes everyone loose if one side is dishonest. I would certainly not endorse that.

Then you must also be against the common system of payment up front, where the customer loses.

Payment up front: customer loses, and the thief gets the money. Simple escrow: customer loses, but the thief doesn't get the money either.

Are you guys saying payment up front is better, because at least the thief gets the money, so at least someone gets it?

Imagine someone stole something from you. You can't get it back, but if you could, if it had a kill switch that could be remote triggered, would you do it? Would it be a good thing for thieves to know that everything you own has a kill switch and if they steal it, it'll be useless to them, although you still lose it too? If they give it back, you can re-activate it.

Imagine if gold turned to lead when stolen. If the thief gives it back, it turns to gold again.

It still seems to me the problem may be one of presenting it the right way. For one thing, not being so blunt about "money burning" for the purposes of game theory discussion. The money is never truly burned. You have the option to release it at any time forever.

Re: Escrow

Posted by ribuck, August 11, 2010, 11:13:12 AM

> Quote from: Inedible on August 11, 2010, 01:52:53 AM
>
> . . . It's just a shame there's nothing that can be done to mitigate malicious intent by offering to sell something, only to 'burn' the payment and never send the goods (assuming they even existed).

> This would just be a case of spite but a very real threat none the less.
>
> E.g.
>
> A offers to sell laptop
> B offers to buy and escrows 2000 bitcoins
> A confirms that item is sent but never sends it
> B never receives it so never release the bitcoins
> A doesn't care because their intent was to make B 'spend' their bitcoins with no recompense
>
> ---
>
> How about this:
>
> A offers to sell laptop for 2000 bitcoins, and escrows 2500 bitcoins as security
> B offers to buy and escrows 2500 bitcoins
> A confirms that item is sent but never sends it
> B never receives it so never release the bitcoins
> A now cares because he has 2500 bitcoins in escrow as security
>
> In this scenario, it's in A's interest to send the laptop, otherwise he loses his BTC 2500 security. It's also in B's interest to confirm receipt of the laptop, otherwise he loses his BTC 500 "excess".
>
> The awkward situations are going to arise if both A and B are honest, but an uninsured delivery service loses or breaks the laptop, or if one of the participants dies before releasing the escrow.

And another thread surfaced later:

How to Make a Distributed Bitcoin Escrow Service

Posted by harding, September 26, 2010, 01:16:18 AM

Summary: Giving BitCoin a decentralized escrow would give it an advantage over all other exchange mediums, which might increase its adoption rate. Details follow.

For a *decentralized* currency, *centralized* escrows seem to be the norm for BitCoin today. An example:

Alice wants to buy $5 USD worth of BitCoins from Bob, but neither Alice nor Bob fully trust the other, so they go to a site they both trust–say Mt. Gox. There they deposit their respective monies and there they have Mt. Gox make the exchange for them.

No offense to Mt. Gox (a site I like), but can we do without its escrow service?

An almost distributed alternative:

Charlie, a trusted third-party, generates a BitCoin private key.

Charlie then uses the Unix command `split` to split the private key in half–giving one half to Alice and one half to Bob.

Bob deposits $5 USD worth of BitCoins into the split BitCoin account;

Alice verifies the transaction using the public block;

Alice sends $5 USD to Bob by PayPal;

Bob verifies the PayPal transaction;

Bob sends Alice his half of the split private key so Alice can

access the BitCoins he deposited earlier.

(For simplicity I omit part of the PayPal details like who pays the transaction fee and how long you should wait to avoid chargeback fraud. I also omit any incentive for Bob to perform the final step.)

More advanced almost-distributed examples can be made if we substitute something more sophisticated for the Unix command `split`. For example: a Shamir's secret sharing scheme implementation like `ssss`[1]. A utility like `ssss` allows Alice and Bob to appoint an arbiter in case they get in a disagreement.

The problem with all of this, of course, is that we must trust Charlie to not abuse the full copy of the private key he creates.

The ideal solution would be for Alice and Bob to each generate half of the private key on their own. I don't fully understand the math used in modern keypairs, but I doubt this is possible with the current algorithm.

Is there an alternative way for Alice and Bob to each acquire half of a private key without giving the whole key to any party?

—Dave

[1] See: http://en.wikipedia.org/wiki/Shamir's_Secret_Sharing

Re: How to Make a Distributed Bitcoin Escrow Service

Posted by satoshi, September 26, 2010, 05:34:26 PM

It's not implemented yet, but the network can support a transaction that requires two signatures. It's described here:
http://bitcointalk.org/index.php?topic=750.0

It's absolutely safer than a straight payment without escrow, but not as good as a human arbitrated escrow, assuming you trust the human enough.

In this kind of escrow, a cheater can't win, but it's still possible for you to lose. It at least takes away the profit motive for cheating you. The seller is assured that the money is reserved for him, while the buyer retains the leverage that the seller hasn't been paid yet until completion.

55

On Bitcoin Mining as a Waste of Resources

THE ARGUMENT that bitcoin mining is a waste of resources has often been reported in the media. If Satoshi were not anonymous and were still involved, his interviews would inevitably include this question. Thus, seeing the answer he would likely give presented in these posts is illuminating.

Bitcoin minting is thermodynamically perverse

Posted by gridecon, August 06, 2010, 01:52:00 PM

Let me begin by saying that Bitcoin is an amazing project and I am very impressed with the implementation and the

goals. From reading these forums it seems to be understood that debate about the design and operation of the bitcoin economy ultimately serves to strengthen it, so I hope these comments are taken in that spirit. *EDIT - I have been convinced by further research and discussion that Bitcoin is actually highly efficient compared to most traditional currencies, because the infrastructure required to support a government issued fiat currency represents a much larger investment of resources than Bitcoin's cpu power consumption. I am leaving this thread active though because it has been generating a lot of interesting discussion.*

I believe that the amount of energy input required to the bitcoin economy represents a serious obstacle to its growth. I think in the long-term, transactions may be even more serious than minting in this regard, but I will for the moment discuss minting because it is more precisely bounded and defined. The idea that the value of bitcoins is in some way related to the value of the electricity required, on average, to mint a winning block is generally accepted, but the precise nature of this relationship is contentious.

One argument is that anyone who chooses to generate coins is actually making the choice to purchase bitcoins with electricity/computational resources, and that because some/many people are in fact making that choice, bitcoins have at least that much "value" to the generators, who can be assumed to be maximizing their utility. A contrasting argument is that cost of production is different than market value, and the most objective measure is the current market conversion price to a more liquid and widely traded currency such as the US dollar.

My contention is that both of these arguments miss the point and the real problem, which is the fundamental perversity of wasting large amounts of energy and computations in generating the winning blocks for the minting process. The

minting process exists because of the necessity of actually "printing" the currency, and certain desirable properties of crypto-math for making the currency's behavior predictable. The fact that the current minting process requires a large energy input of computational work is highly unfortunate and has the perverse consequence that bitcoin may actually be "destroying wealth" in the sense of wasting energy producing a digital object worth less than the resources invested in it.

As is often pointed out, a currency does not necessarily have, or need to have, any inherent value - a medium of exchange is a useful tool and can have value purely as a consequence of social convention. The cost of production of bitcoins in electricity consumed represents a waste, a "thermodynamic burden" that the currency has to carry. Consider a hypothetical alternative digital currency called "compucoin", which purchases cpu cycles from nodes on the network. The market value of this currency would converge very closely with the cost of electricity required to generate cpu cycles. Instead of costing cpu cycles to mint, the value of the cpu cycles the coins could be exchanged for would create a rational basis for the currency's value and integrate it with an existing market. I imagine that alternatives to Bitcoin (many of them probably sharing a lot of Bitcoin's source code) will inevitably emerge and Bitcoin's current minting process makes the currency "expensive" in terms of energy input. I believe this places it at a competitive disadvantage to other currencies and can only hinder its widespread adoption and long-term value. *Edit - as mentioned above, I am now much more optimistic about Bitcoin long term. I still think compucoins would be a cool idea, though!*

Re: Bitcoin minting is thermodynamically perverse

Posted by satoshi, August 07, 2010, 05:46:09 PM

It's the same situation as gold and gold mining. The marginal cost of gold mining tends to stay near the price of gold. Gold mining is a waste, but that waste is far less than the utility of having gold available as a medium of exchange.

I think the case will be the same for Bitcoin. The utility of the exchanges made possible by Bitcoin will far exceed the cost of electricity used. Therefore, *not* having Bitcoin would be the net waste.

> Quote from: gridecon on August 06, 2010, 04:48:00 PM
>
> As an overall point, I also do not agree with the idea that the very high computational burden of coin generation is in fact a necessity of the current system. As I understand it, currency creation is fundamentally metered by TIME - and if that is the fundamental controlling variable, what is the need for everyone to "roll as many dice as posible" within that given time period? The "chain of proof" for coin ownership and transactions doesn't depend on the method for spawning coins.

Each node's influence on the network is proportional to its CPU power. The only way to show the network how much CPU power you have is to actually use it.

If there's something else each person has a finite amount of that we could count for one-person-one-vote, I can't think of it. IP addresses . . . much easier to get lots of them than CPUs.

I suppose it might be possible to measure CPU power *at certain times*. For instance, if the CPU power challenge was only run for an average of 1 minute every 10 minutes. You could still prove

your total power at given times without running it all the time. I'm not sure how that could be implemented though. There's no way for a node that wasn't present at the time to know that a past chain was actually generated in a duty cycle with 9 minute breaks, not back to back.

Proof-of-work has the nice property that it can be relayed through untrusted middlemen. We don't have to worry about a chain of custody of communication. It doesn't matter who tells you a longest chain, the proof-of-work speaks for itself.

Re: Bitcoin minting is thermodynamically perverse

Posted by satoshi, August 09, 2010, 09:28:39 PM

The heat from your computer is not wasted if you need to heat your home. If you're using electric heat where you live, then your computer's heat isn't a waste. It's equal cost if you generate the heat with your computer.

If you have other cheaper heating than electric, then the waste is only the difference in cost.

If it's summer and you're using A/C, then it's twice.

Bitcoin generation should end up where it's cheapest. Maybe that will be in cold climates where there's electric heat, where it would be essentially free.

Re: Bitcoin minting is thermodynamically perverse

Posted by throughput, August 10, 2010, 12:27:30 PM

I think the discussion have eventually lost the ethic aspects of motivating the botnet creators to
invest even more resources in their business in case when BTCs generated will deliver the value,
comparable to the current uses of botnets.
What if Bitcoin operation will outperform the other activities?
How can you imagine, that botnet building process is done in a way, that benefit the community?

Quote from: jgarzik on August 06, 2010, 07:53:25 PM

> Participation in the network as an honest node helps everyone.
>
> Yes, but only when it is not against the computer owner's will, he pays the electricity bill.
>
> If it is, then he loses REAL money for an extra power consumption caused by 100% CPU load.
> So, Bitcoin motivates behavior of stealing computing power from innocent computer owners.
>
> Well, you may now try to compare the social harm to the benefits, but do you really feel you have the moral right to do so?

Re: Bitcoin minting is thermodynamically perverse

Posted by Gavin Andresen, August 10, 2010, 09:26:14 PM

Quote from: throughput on August 10, 2010, 12:27:30 PM

> So, Bitcoin motivates behavior of stealing computing power from innocent computer owners.

Sure, in exactly the same way the existence of credit cards motivates behavior of stealing credit card numbers from innocent credit card users.

Or the existence of bank accounts motivates hackers to try to break into your system to find out your bank account number.

Or the existence of cars motivates some people to steal gasoline from innocent service station owners.

I believe the benefits of Bitcoin will outweigh the harm, and I further believe that I **am** capable of making that moral judgment. I might be wrong, and I might regret I ever got involved, but if I only ever did things that I was 100% certain were going to work out for the best I would never accomplish anything new and interesting.

56

On an Alternate Type of Block Chain with Just Hash Records

HERE, a suggestion that Satoshi considered interesting is discussed. This suggestion relies on giving less information in the block chain, with the intent of providing a greater level of privacy.

Not a suggestion

Posted by Red, August 10, 2010, 05:45:45 AM

As some might have noticed, one of the things that bugs me about bitcoin is that the entire history of transactions is

completely public. I totally understand the benefits of how this simplifies things and makes it easy for everyone to prove coins are valid.

So this is not a suggestion for a change to bitcoin. Rather it is a question about what could be possible, and what couldn't be possible.

The general question is, could the block list be/have been implemented in a way that didn't store the full transactions in the list? Specifically, *perhaps* it would be possible to store only hashes of the in-points, out-points in the block list. These would be time stamped (notarized) in the blocklist exactly as is being done now.

The major difference is that it would be the coin receiver's responsibility to store the full transaction. And perhaps he might have to store previous transactions (X) deep to show history.

Then when he wanted to transfer the coins to the next party, he would create a transaction exactly as is being done now, except he would have to submit the antecedents to the transaction for validation as well. For validation, each antecedent of the in-points would be hashed and validated as existing in the block list. The in-points would be hashed and identified in the blocklist as not yet spent. Then the transaction would be validated as is currently done.

If everything validated correctly, the additional in/out-point hashes would be added to the block. This closes the transaction's in-points, and marks the new out-point hashes as unspent.

Once a node completes the block (by winning the hashing contest), he then broadcasts the block of hashes and the related transactions+plus antecedents to the other nodes for confirmation and acceptance.

as a rough example:

{block-9
 hash-a, hash-b, hash-c, hash-x
}
{block-12
 hash-a, hash-y, hash-c, hash-d
}
{block-17
 hash-b, hash-d, hash-e, hash-z, hash-f
}

{Transaction
 {in-points: hash-x, hash-y, hash-z}
 {address, signature and other transactions stuff}
 {out-points: hash-payed, hash-change
}

{generating-block
 hash-x, hash-y, hash-z, hash-payed, hash-change
}

So basically, if the i/o-point hash existed twice in the block list, it has been spent. If it exists only once it has not been spent.

So in after block-17:
 a, b, c & d are spent.
 e, f, x, y, z are unspent.

The transaction spends x, y & z and creates hash-payed & hash-change, so the transaction is valid.

After the generating-block:
 a, b, c, d, x, y, & z are spent.
 e, f, payed, change are unspent.

====

The Goal:

The goal is to provide all the same security of the existing system, but to avoid creating a public graph of every transaction that is easily correlated. In this case, the hashes don't even have to associate in the block. The block could simply sort all hashes in ascending order.

In effect, I want to create real gold coins. I can give my coins to you, but everyone in the world doesn't know I did. You can give them to the next guy and prove they are pure gold coins, because you have the pedigree of the coins AND every generation in the pedigree was notarized in the public record.

====

The Question:

Satoshi showed that you can remove transactions from the block list through the Merkle tree structure, without compromising security. I guess my real question is:

"What is the earliest you can remove the transactions?"

You could argue that nodes could remember everything anyway (the web never forgets). But if you structured the protocol so that new nodes would only receive a block list of hashes, they could only remember from this moment forward. That would give a little additional privacy. (Maybe)

====

Any thoughts? Is there an obvious way that people could cheat and get rich?

Re: Not a suggestion

Posted by Insti, August 10, 2010, 09:34:14 AM

In your system, Rather than just getting transactions from the block chain I just have to watch every transaction (which I'll see anyway) and log them to my secret server.

You're just advocating security through obscurity.

Re: Not a suggestion

Posted by Red, August 10, 2010, 02:09:36 PM

> Quote from: Insti on August 10, 2010, 09:34:14 AM
>
> You're just advocating security through obscurity.

I did mention that. I wouldn't count on this for monetary security. I would like the system to be equivalent to the current one.

However, privacy obscurity is known to add value. Your neighbors, or the FBI could me watching everything you do all day long. But they probably aren't. If you happen to become "of interest", sure they could start watching you now and from this time forward.

But the most asked for additional legal powers seems to be, "let me examine everyone's logs!" (phone calls, cell towers, email connections, facebook connections, credit/debit card transactions, Google history, browser history.) The other systems are "security though authority." Bitcoin doesn't have that.

By the way, I'd rather not broadcast every transaction to every node either. But that is for another thread.

By the way, this is the way most digital notary services work. You send them a hash of a signed document and they log it permanently. Then they create a hash chain like bitcoin does. They periodically publish the current hash chain value in a newspaper or other offline redundant record.

You don't have to send your private documents/transaction to the notary for them to be time stamped and recorded. The notary is just certifying that something that matched this hash existed at this point in time.

Re: Not a suggestion

Posted by Insti, August 10, 2010, 03:06:16 PM

> Quote from: Red on August 10, 2010, 02:22:09 PM
>
> By the way, this is the way most digital notary services work. You send them a hash of a signed document and they log it permanently. Then they create a hash chain like bitcoin does. They periodically publish the current hash chain value in a newspaper or other offline redundant record.
>
> You don't have to send your private documents/transaction to the notary for them to be time stamped and recorded. The notary is just certifying that something that matched this hash existed at this point in time.

You also don't have to prove to the notary that you have X BTC in your account to spend.

Although I was recently reading about Zero-knowledge proofs ((http://en.wikipedia.org/wiki/Zero-knowledge_proof) if you could use something like that to prove that your account had X BTC in it without revealing anything else it might be what you're looking for.

I'm just worried what you want is theoretically impossible.

Re: Not a suggestion

Posted by Red, August 10, 2010, 05:29:44 PM

> Quote from: Insti on August 10, 2010, 03:06:16 PM
>
> Although I was recently reading about Zero-knowledge proofs (http://en.wikipedia.org/wiki/Zero-knowledge_proof)

Interesting idea to revisit! Thanks. Hadn't thought of them in a while.

Re: Not a suggestion

Posted by satoshi, August 11, 2010, 12:14:22 AM

This is a very interesting topic. If a solution was found, a much better, easier, more convenient implementation of Bitcoin would be possible.

Originally, a coin can be just a chain of signatures. With a timestamp service, the old ones could be dropped eventually before there's too much backtrace fan-out, or coins could be kept individually or in denominations. It's the need to check for the absence of double-spends that requires global knowledge of all transactions.

The challenge is, how do you prove that no other spends exist? It seems a node must know about all transactions to be able to verify that. If it only knows the hash of the in/outpoints, it can't check the signatures to see if an outpoint has been spent before. Do you have any ideas on this?

It's hard to think of how to apply zero-knowledge-proofs in this case.

We're trying to prove the absence of something, which seems to require knowing about all and checking that the something isn't included.

Re: Not a suggestion

Posted by Red, August 11, 2010, 04:58:50 AM

Satoshi: I know you know the first part of what I'm writing, but I want others to be able to follow and for you to correct any misconceptions I might have.

I was looking at the current Merkle tree implementation trying to figure out when transactions could be removed without losing security.

In transaction graph terms, the transactions represent the nodes. The edges of the transaction graph are represented by the in-points which point to previous transactions using a BlockHash->TransHash->OutPoint kind of structure. It is the existence of an in-point that marks a previous out-point spent.

So for a transaction to be valid, you most show for every in-point in a transaction that BOTH, a previous out-point exists AND no previous in-point exists that references that out-point. So for every out-point, there are zero or one in-points referring to it. zero = unspent. one = spent.

That also means that no transaction can be culled from the block list, until both its out-points are spent. Otherwise coins will disappear.

You can however, delete all double-bound transactions as

soon as you are confident the 2nd binding block will stick around. (earliest possibility)

However, as you delete transactions and replace them with their tree hashes, you lose the graph structure present in the block list. In effect, all transactions undeleted from the block list have unspent value purely because they still exist. They can no longer prove validity by ancestry since that part of the graph was culled.

Which got me thinking, is there a way to prove validity if you never put the whole transactions into the graph to begin with?

Quote from: satoshi on August 11, 2010, 12:14:22 AM

The challenge is, how do you prove that no other spends exist? It seems a node must know about all transactions to be able to verify that. If it only knows the hash of the in/outpoints, it can't check the signatures to see if an outpoint has been spent before. Do you have any ideas on this?

The key is to hash the transaction information as part of the out-point hash. So instead of creating a single transaction hash, you represent the transaction as two out-point hashes. (I originally considered an in-point/transaction/out-point structure using hashes, but that proved unnecessary.)

Only transaction validators need to know the bitcoin address associated with a recorded out-point hash. That comes from the submitted antecedent transaction for an in-point of the current transaction. The antecedent transaction and out-point is hashed and presumed BOTH valid and unspent if that hash appears one-and-only-one time in the block list.

The current transaction must be signed by the key for the address in the antecedent transactions of course. If this proves valid, two new out-point hashes are generated and inserted in the current block. The in-point hashes are marked spent by including them in the current block as well. (If a hash exists

twice it is spent.) If you want to represent the transaction as a unit (and the currently visible transaction graph), the in-point hashes and out-point hashes could be grouped. However, this is not strictly necessary to prove validity.

Quote from: satoshi on August 11, 2010, 12:14:22 AM

We're trying to prove the absence of something, which seems to require knowing about all and checking that the something isn't included.

In this case we are trying to prove the presence of ONE matching hash and the absence of TWO matching hashes. It does require knowing all of them to prove.

I think the prohibitions against double spending are as strong as in the current version.

==== CAUTION! ====

However, you have to consider the case where a node causes mischief by deliberate adding random "canceling hashes". In this case, the node wouldn't be able to gain access to the coins, as he has no signed transaction hashing to a valid unspent out-point hash. However, the current owner wouldn't be able to spend the coins either. The in-point would be presumed already spent.

That means the validation conditions are EXACTLY THE SAME as with the current implementation. All validating nodes must examine and validate all transactions represented in a block before accepting it and building on it.

If there exist any hashes in the proposed block that are not represented by valid transactions, the block must be rejected. That is exactly the same as the current system's, if any transaction doesn't validate, the block must be rejected.

I had hoped the condition to pass all transactions to all

validators could be weakened but I can't see how (yet) without relying on trusted delegation.

———

An interesting feature is that this simplifies the validation process. All that needs to be done is to parse the block list (of hashes) once. As each hash is parsed you simply look it up in a hash-set. If it doesn't exist you add it. If it does exist you delete it. When you are done parsing the block list, you will have the minimal set of valid and unspent out-points. You might even be able to keep the whole set in memory. (at least for a while!)

Quote from: satoshi on August 11, 2010, 12:14:22 AM

It's hard to think of how to apply zero-knowledge-proofs in this case.

It's hard for me too! :-) Was interesting to re-read though!

Was hoping it would spawn some insight on a way for nodes to demonstrate that they "always follow" the block generating rules, in absence of everyone needing to have the set of all transactions to double check.

It didn't. :-)

Re: Not a suggestion

Posted by satoshi, August 11, 2010, 09:07:59 PM

Still thinking this idea through…

The only job the network needs to do is to tell whether a spend of an outpoint is the first or not.

If we're willing to have clients keep the history for their own money, then some of the information may not need to be stored by the network, such as:

- the value
- the association of inpoints and outpoints in one transaction

The network would track a bunch of independent outpoints. It doesn't know what transactions or amounts they belong to. A client can find out if an outpoint has been spent, and it can submit a satisfying inpoint to mark it spent. The network keeps the outpoint and the first valid inpoint that proves it spent. The inpoint signs a hash of its associated next outpoint and a salt, so it can privately be shown that the signature signs a particular next outpoint if you know the salt, but publicly the network doesn't know what the next outpoint is.

I believe the clients would have to keep the entire history back to the original generated coins. Someone sending a payment would have to send data to the recipient, as well as still communicating with the network to mark outpoints spent and check that the spend is the first spend. Maybe the data transfer could be done as an e-mail attachment.

The fact that clients have to keep the entire history reduces the privacy benefit. Someone handling a lot of money still gets to see a lot of transaction history. The way it retrospectively fans out, they might end up seeing a majority of the history. Denominations could be made granular to limit fan-out, but a business handling a lot of money might still end up seeing a lot of the history.

Re: Not a suggestion

Posted by Red, August 12, 2010, 01:10:19 AM

> Quote from: satoshi on August 11, 2010, 09:07:59 PM
>
> Still thinking this idea through...

It's a bit of a brain twisting idea isn't it. :-)

It turns out the notion of a cancelable notarization generalizes nicely.

For example this system is not limited to bitcoin transactions. Since the signed contracts are kept externally, with additional validation/notarization rules, you could easily implement things like IOUs/claim checks.

If someone gave you $5, you could give him a $5 IOU. Its IOU hash would be notarized into the blocks list (of hashes). When you pay them back you could have them sign the IOU for confirmation. Then have the notary insert an IOU hash cancellation. Then no one could show back up with a copy of the IOU and demand double payment.

> Quote from: satoshi on August 11, 2010, 09:07:59 PM
>
> I believe the clients would have to keep the entire history back to the original generated coins. The fact that clients have to keep the entire history reduces the privacy benefit.

I thought this too at first. But then I convinced myself otherwise.

It is really a matter of how much trust you place in the verifiers and the process of verification. People like the warm fuzzys that having every transaction available lets them trace the roots of their money back to its creation. However that is not required.

If you are confident in the process that validated the transactions during block creation (> 50% CPU agreement). And if you are confident the previous blocks can't be changed (you proved this). Then you only need to check that related out-points have not been spent. The security features remain in the block list and procedure, even if the transactions themselves are stored externally and the predecessors are not stored at all. You showed this yourself by proving old transactions can be deleted using the Merkle tree to maintain consistency.

> Quote from: satoshi on August 11, 2010, 09:07:59 PM
>
> Someone handling a lot of money still gets to see a lot of transaction history. The way it retrospectively fans out, they might end up seeing a majority of the history. Denominations could be made granular to limit fan-out, but a business handling a lot of money might still end up seeing a lot of the history.
>
> True, privacy is directly related to observability. If there is a central party like a money changer, he can relate a lot of out-points. But if we get away from the notion that every coin must be traced back to creation, the observation horizons will be much closer.
>
> —
>
> It's really weird getting used to the notion that this coin is valid simply because the process wouldn't let it be included otherwise. But really, that is exactly how bitcoin generation works. The transaction has no inputs, but everyone decides the out-point must be valid purely because otherwise, it wouldn't be in the block at all. :-)

Re: Not a suggestion

Posted by satoshi, August 12, 2010, 02:46:56 AM

> Quote from: Red on August 12, 2010, 01:10:19 AM
>
> > Quote from: satoshi on August 11, 2010, 09:07:59 PM
> >
> > I believe the clients would have to keep the entire history back to the original generated coins. The fact that clients have to keep the entire history reduces the privacy benefit.
>
> I thought this too at first. But then I convinced myself otherwise.

Are you back to talking about the existing Bitcoin system here?

I was talking about in the hypothetical system I was describing, if the network doesn't know the values and lineage of the transactions, then it can't verify them and vouch for them, so the clients would have to keep the history all the way back.

If a client wasn't present until recently, the two ways to convince it that a transaction has a valid past is:
1) Show it the entire history back to the original generated coin.
2) Show it a history back to a thoroughly deep block, then trust that if so many nodes all said the history up to then was correct then it must be true.

But if the network didn't know all the values and lineage of the transactions, it couldn't do 2), I don't think

Re: Not a suggestion

Posted by Red, August 12, 2010, 04:25:51 AM

> Quote from: satoshi on August 12, 2010, 02:46:56 AM
>
> > Quote from: Red on August 12, 2010, 01:10:19 AM
> >
> > I thought this too at first. But then I convinced myself otherwise.
>
> Are you back to talking about the existing Bitcoin system here?

Yes, I am talking about the hypothetical system.

The way I proposed the system, each time a block gets generated every validating node must accept or reject that block by validating the transactions and confirming the hashes in the block. In effect, the same work that is being done with the current system, plus the out-point hash checks. Since the other validators were already competing to generate the block, they already have (at least most of) the transactions.

As with the current system, if the transactions don't validate (plus match included out-point hashes) the other nodes will reject the block. If the block doesn't get acceptance by at least 50% of the CPU power, it doesn't make the block list.

So the presence of the hashes in the block list, signifies that at least 50% of the existing validators at that time saw and validated all the containing transactions and out-point hashes.

Therefore (barring hash crashes) if someone submits an antecedent transaction that matches an unspent out-point, it must be valid.

That antecedent's antecedent must have been valid as well, otherwise the antecedent would have been rejected. And so on and so on.

For that not to be the case, you have to postulate that there was a period in time where blocks weren't being validated against out-point hashes. But that's plausibly implausible with the CPU competition system.

Quote from: satoshi on August 12, 2010, 02:46:56 AM

If a client wasn't present until recently, the two ways to convince it that a transaction has a valid past is:

1) Show it the entire history back to the original generated coin.

2) Show it a history back to a thoroughly deep block, then trust that if so many nodes all said the history up to then was correct then it must be true.

If a client joined the network recently, it did so presuming that prior validators followed the rules and all pre-existing blocks are valid. (No one would join a known corrupt network)

Sure, in the current system, if transactions were never purged, a new node could validate all prior blocks for self consistency. But they still couldn't prove absolute truth. A bot net could have taken over and erased some transactions leaving "a new truth" and unhappy users. Equivalent to case 1) above.

In the current system, if transactions were Merkle tree purged then you have case 2) above. New comers must trust in the process. Anything missing, they don't need to worry about. Everyone must presume it was valid.

The unique thing I'm saying is that, if you have confidence in the bitcoin validation competition process (and we do!), then you really don't need "a 2) thoroughly deep block" to be very deep at all. Someone said in another thread that clients reject any changes to blocks more than two hours old. So we can have absolute confidence in all blocks buried 12 deep.

So if a transaction is unspent and buried 12 deep, we can purge all it's ancestors. They add warm fuzzies but no additional validation. We have to rely on them. There is simply no way to back up and change course.

After that, every succeeding block presumes all the preceding blocks are true. Otherwise it would be a fork and not a succeeding block. So for any transaction validated against out-points in a preceding block, if those out-points exist and are unspent, they must be presumed valid. If those are presumed valid, their ancestors must be presumed valid even if purged.

—

In the proposed system, exactly the same things are true.

If an antecedent out-point hash is unspent and buried 12 blocks deep, then it is absolutely unspent. Nothing can change that fact. No point in checking its ancestors. You can finish validating the transaction, cancel the in-points hashes and create new out-point hashes.

Interestingly, if an antecedent out-point hash is unspent and buried LESS THAN 12 blocks deep, then it is RELATIVELY unspent. Curiously, there is still no point in checking its ancestors. The only thing that could change the antecedent's validity is a branch swap to a longer chain. If an ancestor of the antecedent you are validating this transaction against was swapped out, this transaction would be swapped out as well.

It's one of those cheesy time machine movie plots. Someone when back in time and spent my ancestor. Now I don't exist!

=====

So what I'm saying is that in BOTH systems (existing and proposed) the only thing validators need to do is to validate that the antecedent out-points exist and are unspent (for the current block chain). The process assures that everything else remains

relatively or absolutely valid.

The rest is just warm fuzzies.

– PS –

I know this is too long and redundant, but I'm to tired to edit. :-)

Re: Not a suggestion

Posted by satoshi, August 13, 2010, 07:28:47 PM

I'm not grasping your idea yet. Does it hide any information from the public network? What is the advantage?

If at least 50% of nodes validated transactions enough that old transactions can be discarded, then everyone saw everything and could keep a record of it.

Can public nodes see the values of transactions? Can they see which previous transaction the value came from? If they can, then they know everything. If they can't, then they couldn't verify that the value came from a valid source, so you couldn't take their generated chain as verification of it.

Does it hide the bitcoin addresses? Is that it? OK, maybe now I see, if that's it.

Crypto may offer a way to do "key blinding". I did some research and it was obscure, but there may be something there. "group signatures" may be related.

There's something here in the general area:
http://www.users.zetnet.co.uk/hopwood/crypto/rh/

What we need is a way to generate additional blinded variations

of a public key. The blinded variations would have the same properties as the root public key, such that the private key could generate a signature for any one of them. Others could not tell if a blinded key is related to the root key, or other blinded keys from the same root key. These are the properties of blinding. Blinding, in a nutshell, is x = (x * large_random_int) mod m.

When paying to a bitcoin address, you would generate a new blinded key for each use.

Then you need to be able to sign a signature such that you can't tell that two signatures came from the same private key. I'm not sure if always signing a different blinded public key would already give you this property. If not, I think that's where group signatures comes in. With group signatures, it is possible for something to be signed but not know who signed it.

As an example, say some unpopular military attack has to be ordered, but nobody wants to go down in history as the one who ordered it. If 10 leaders have private keys, one of them could sign the order and you wouldn't know who did it.

Re: Not a suggestion

Posted by Red, August 13, 2010, 09:48:56 PM

I'm going to reply to this in two parts.

> Quote from: satoshi on August 13, 2010, 07:28:47 PM
>
> I'm not grasping your idea yet.

That's my fault, because I was trying not to make too many claims at once. I was also not trying to introduce too many new "features" at once for analysis.

My mental goal is to incrementally constrain the horizon of

transaction visibility. Both in time and in space. Time meaning say to only nodes running at a particular instant. Space meaning to less than the set of all nodes running at the time. Optimally, a transaction would only be known to the sender and the receiver. Then all proof would disappear.

I hand you a $10 bill. Then we walk away forever. As long as no one observed me handing you the bill at that moment, no one can ever discover it by examining the bill itself.

> Quote from: satoshi on August 13, 2010, 07:28:47 PM
>
> Does it hide any information from the public network? What is the advantage?
>
> If at least 50% of nodes validated transactions enough that old transactions can be discarded, then everyone saw everything and could keep a record of it.
>
> I initially hoped that all transactions would be validated only between the parties concerned. In effect the block generating nodes would just record the hashes that got told to them.
>
> However, at the last minute I realized that since the hashes were not signed or otherwise verified, it became possible to easily falsify a "cancel the previous out-point hash". You couldn't steal someone's coins but you could invalidate them.
>
> I can see three possible ways forward on that pesky detail. 1) let all verifiers see the transactions, minimize what is saved. 2) come up with some way to minimize the number of validators that need to see each transaction. 3) create a single use keypair for each new out-point. Sign the hashes. (Last minute entry!)
>
> 1) I initially wrote about the first case, because it introduced less variables at once. I wanted to be sure recording only hashes wasn't an obvious FAIL.
>
> I tried to quantify what bit of privacy we would gain. It is minimal

in the worst case, (everyone saves everything anyway) but it is considerable in the nominal case, most people don't save anything they don't need for themselves.

So in this increment, the benefit is, any new threats can only observe a transactions that occur after they join. They can't look back in time, unless they can both identify an earlier adopter who recorded everything from when they joined, and convince them to share. So minimal protection, but at least your Ex isn't going to be snooping around after the fact. :-)

2) However, it is possible to minimize the space horizon with a clever use of a DHT. All details are not worked out yet, but you can visualize it by splitting the block list into say 1024 identical block lists each with 10 redundant validating nodes. Rather than one blocks list with 10,000 redundant validating nodes. Each randomly chosen set of nodes is responsible for a segment of the hash space.

But instead of guaranteeing that 50% of all CPU power is required to fake something, you might aim for 100% consensus and a complete broadcast of the chain checksum and/or blocks. So upon periodic DHT re-org any new node can verify that the chain has always remained 100% consistent. (Similar to publishing each of the 1024 checksums in the newspaper each day)

This restricts an attacker's visibility to know what hash he would want to cancel. (I only see 1/1024th of the transactions) And it limits his time window to submit a fraudulent cancelation to a time window when he controls 100% of a bucket's verifiers.

So there is a potential path to gain some privacy by restricting some visibility. It comes at some potential risk.

3) So in reality I need to give you credit for sparking the best case idea. Kudos! I initially dismissed the idea of signing the out-point hashes, because it seems so much like the existing bitcoin addresses. I assumed the public key required in the signature

would associate too many things.

However, if you use a one-time public key where you sign a combination of the out-point hash and the current block number. Then when the out-point hash is initially created it is recorded with a public key. When it is spent the hash is verified by having a different but related signature, signed by the same key.

I think that solves the problem completely. There are no additional associations because the two single use instances of the out-point hash in the block list HAVE TO be related. Adding a second single use public key identifier adds nothing.

To simplify the "current block number" issue, the submitter might submit signatures for the next 3-4 block numbers. The validator would only record the appropriate one. To the block

It does add more bits to the block list than I was hoping to. I thought a hash only was optimal.

====

What is the smallest crypto construct that has the following properties? Might be able consider that instead of a hash and full signature.

1) I give you something that appears arbitrary.
2) I give you something that appears easily related to your #1 but unrelated to anyone else's #1.
3) Nobody else could figure out your #2 from #1.

====

For example

1) I give you Z where Z = X * Y and both X & Y are large primes
2) I give you the tuple (X, Y)
3) Nobody can factor X and Y from Z

In that case, when sending an offline transaction, the sender

would enclose (X,Y) for each in-point.
The receiver would privately create a new (X,Y) for each new out-point.
The receiver then broadcasts each in-point's (X,Y) to cancel them.
It broadcasts each out-point's Z to create them.

Does that work, or is it too naive?

Re: Not a suggestion

Posted by Red, August 13, 2010, 10:20:50 PM

Quote from: satoshi on August 13, 2010, 07:28:47 PM

Crypto may offer a way to do "key blinding". I did some research and it was obscure, but there may be something there. "group signatures" may be related.

There's something here in the general area:
http://www.users.zetnet.co.uk/hopwood/crypto/rh/

What we need is a way to generate additional blinded variations of a public key. The blinded variations would have the same properties as the root public key, such that the private key could generate a signature for any one of them. Others could not tell if a blinded key is related to the root key, or other blinded keys from the same root key. These are the properties of blinding. Blinding, in a nutshell, is x = (x * large_random_int) mod m.

When paying to a bitcoin address, you would generate a new blinded key for each use.

Then you need to be able to sign a signature such that you can't tell that two signatures came from the same private key. I'm not sure if always signing a different blinded public key would already give you this property. If not, I think that's where group

> signatures comes in. With group signatures, it is possible for something to be signed but not know who signed it.
>
> As an example, say some unpopular military attack has to be ordered, but nobody wants to go down in history as the one who ordered it. If 10 leaders have private keys, one of them could sign the order and you wouldn't know who did it.

This is a really cool idea. I think I see where you were going with it. It took me a few tries to fit it all together. I'm a bit slow.

I'm correct, you were suggesting that you could sign an out-point hash with a single-use blinded key.

Where the blinded public key is equivalent to the public key of the transaction's bitcoin address. Say the bitcoin address' public/private key pair was P/p. The the blinded public keys would be P1, P2, P3...Pn. Where each can validate anything signed with the private key (p).

So upon creation when you submit the out-point hash for validation it appears as signed by P1. However, when receiver submits the out-point for cancellation it would be signed P2 or anything besides P1 (since it is already of public record). Both calculated signatures would be the same, but the public key would change. That would signify only someone in possession of the common private key could have generated it.

That is genius!

57

On the Higher Cost of Mining

THIS THREAD DISCUSSES the increase in difficulty of mining following an increase in the amount of computer power, when the increase in computer power is followed by a drop in computer power. Then, the miners left in the network would then have to deal with a much higher difficulty level, which increases the time per block until the next adjustment.

This problem has not affected Bitcoin, but it did greatly affect some alternative cryptocurrencies such as Feathercoin. A solution called Kimoto's Gravity Well was developed for alternative currencies to integrate. The thread below addresses this potential problem.

Satoshi specifically addresses the market response to the cost of mining.

Potential disaster scenario

Posted by gebler, August 14, 2010, 12:43:54 PM

The difficulty for generating bitcoins is periodically adjusted using a method that has worked well this far. However, I am afraid there are plausible scenarios where the current method would misbehave quite spectacularly.

One scenario goes like this:

1) As bitcoins become more known, competition among minters continues to increase, with corresponding increases in difficulty. The increased difficulty will eventually make bitcoin minting clearly unprofitable for those who do not have access to good energy prices and cheap access to an energy-efficient HW/SW combination.

2) Some bitcoin users may continue to mint bitcoins even though it is not profitable for them. This could be due to ideology, the fun factor, or just ignorance. But it is quite plausible that the vast majority of bitcoins will be minted by those who profit from it. Let's say that 99% of all bitcoins are eventually minted by for-profit-minters.

3) The competition among for-profit-minters will drive profit margins down, to the point where it is profitable to continue minting, but barely so. Let's say that the typical profit margin during one difficulty adjustment period (2016 blocks) is 10%.

4) Since bitcoin minting is a decentralized uncoordinated process, we can expect random fluctuations in bitcoin minting activity. This does not affect the difficulty during a specific 2016-block period, so the minting activity can e.g. increase by 20% during a period without making minting unprofitable within that period.

Given the above assumptions, we now have a disaster at

hand at the next difficulty adjustment. As bitcoin production was 20% more than target, difficulty is adjusted upwards by 20%. But the profit margin was only 10%, so for-profit-minters would now lose money if they continued minting. They will therefore stop minting, and as they make up 99% of the minting capacity, generating the next 2016 blocks will take 100 times longer than normal. Everything that depends on block generation will slow to a crawl, and this slowness will persist for a very long time, since the next 2016 blocks will take 100 times longer to generate (almost 4 years rather than two weeks).

Now, if this was to happen, I guess a new client could be released that resets the difficulty to some sensible number and started using a better algorithm for difficulty adjustment. But it would be much better to do it proactively before it becomes a problem (perhaps with a predetermined "flag day" activating the new algorithm at a certain time in the future, giving the new client a chance to propagate).

A simple(?) modification of the algorithm would be to apply the adjustment after a certain amount of time rather than at a certain block number. The switch could still be synced to take effect for the next block, so time synchronization between clients would not need to be super exact to have the vast majority of them agree on when the new difficulty is to be applied.

Also, the difficulty adjustment should probably take into account the adjustments of the number of bitcoins minted per event (now 50, halved every 4 years). Halving the number of bitcoins generated each time is equivalent to doubling the difficulty as far as profitability is concerned, and such a drastical drop in profitability is unnecessary if it can be avoided easily.
I'm not sure if the current adjustment algorithm already takes that into

account somehow, but I couldn't see any obvious adjustment for it in the source code.

Re: Potential disaster scenario

Posted by satoshi, August 15, 2010, 04:37:16 PM

Some places where generation will gravitate to:

1) places where it's cheapest or free
2) people who want to help for idealogical reasons
3) people who want to get some coins without the inconvenience of doing a transaction to buy them

There are legitimate places where it's free. Generation is basically free anywhere that has electric heat, since your computer's heat is offsetting your baseboard electric heating. Many small flats have electric heat out of convenience.

How expensive is heating oil? With the price of oil so high, if it's actually more expensive than electric, then generating would have negative cost.

There's also kids putting it on their parent's power bill, employees their employer, botnets, etc.

Case 3 comes into play for small amounts. The overhead of doing an exchange doesn't make sense if you just need a small bit of pocket change for incidental micropayments. I think this is a nice advantage vs fiat currency, instead of all the seigniorage going to one big entity, let it go in convenience amounts to people who need to scrape up a small amount of change.

58

On the Development of an Alert System

SATOSHI DISCUSSES his development of an alert system where important messages can be delivered across the Bitcoin network only by those who own a private key, in this case Satoshi himself. This could be used, for example, to report to all miners the requirement for an important software upgrade after a bug (issue) has been found.

Development of alert system

Posted by satoshi, August 22, 2010, 11:55:06 PM

I've been working on writing the alert system. Alerts are broadcast through the network and apply to a range of version

numbers. Alert messages are signed with a private key that only I have.

Nodes can do two things in response to an alert:

- Put a warning message on the status bar.
- Make the money handling methods of the json-rpc interface return an error.

In cases like the overflow bug or a fork where users may not be able to trust received payments, the alert should keep old versions mostly safe until they upgrade. Manual users should notice the status bar warning when looking for received payments, and the json-rpc safe mode stops automated websites from making any more trades until they're upgraded.

The json-rpc methods that return errors during an alert are:

sendtoaddress
getbalance
getreceivedbyaddress
getreceivedbylabel
listreceivedbyaddress
listreceivedbylabel

In a reply to someone regarding the alert system:

Re: Development of alert system

Posted by satoshi, August 24, 2010, 11:51:12 PM

If you're so paranoid that you're getting hysterical over this, then surely you're paranoid enough that if a warning message displays on the status bar, you'll check the website and forum.

I think if another bug like the overflow bug occurs, it's important

that automated websites stop trading until their admins can check out what's going on and decide what to do. If you decide it's a false alarm and want to take your chances, you can use the "-disablesafemode" switch.

Re: Development of alert system

Posted by satoshi, August 25, 2010, 03:17:37 PM

It can't do arbitrary actions remotely. Maybe some of you are responding to other posters who suggested the alert system should do more?

If there is an alert, the following json-rpc methods return an error:

sendtoaddress
getbalance
getreceivedbyaddress
getreceivedbylabel
listreceivedbyaddress
listreceivedbylabel

The remaining 14 methods function as normal.

I believe the safer option should be enabled by default. If you want your server to keep trading and ignore an alert saying the money its receiving might be like the money from the overflow bug, then you can use the switch and not blame anyone else if you lose your money.

Worst case if you leave alerts enabled, your site stops trading until you upgrade or add the -disablesafemode switch.

Getting surprised by some temporary down time when your node would otherwise be at risk is better than getting surprised by a thief draining all your inventory.

Someday when we haven't found any new bugs for a long time and it has been thoroughly security reviewed without finding anything, this can be scaled back. I'm not arguing that this is the permanent way of things forever. It's still beta software.

Re: Development of alert system

Posted by satoshi, August 25, 2010, 04:56:15 PM

> Quote from: jimbobway on August 25, 2010, 04:45:22 PM
>
> > Quote from: BioMike on August 23, 2010, 05:15:43 AM
> >
> > @mizerydearia, I think the quote button is easier to find then the reply one.
> >
> > So, theoretical this is a first control system where <some goverment> can arrest satoshi and demand
> > that he hands over his key (or get it from his computer) and shut down the complete network?
> >
> > Or is that not possible? How far would <some goverment> get?
>
> A few rhetorical questions for satoshi:
>
> Can you resist waterboarding?
> Can you endure electric shock?
> All forms of torture?
> Lastly, are you Jack Bauer by any chance? Seriously.

WRT the alert system, who cares? The most the key can do is temporarily disable six json-rpc commands until the site owners either add the -disablesafemode switch or upgrade. All nodes keep running and generating, the network stays up. If I'm not available, any script kiddie can figure out how to add two characters and make a new version that disables the alert system. It would be a temporary inconvenience only.

> Quote from: BioMike on August 23, 2010, 05:15:43 AM
>
> So, theoretical this is a first control system where <some goverment> can arrest satoshi and demand
> that he hands over his key (or get it from his computer) and shut down the complete network?

This is what makes me think the people objecting don't know what they're talking about. It can't "shut down the complete network".

Re: Development of alert system

Posted by satoshi, August 25, 2010, 04:56:15 PM

> Quote from: BioMike on August 25, 2010, 06:23:45 PM
>
> > Quote from: satoshi on August 25, 2010, 04:56:15 PM
> >
> > This is what makes me think the people objecting don't know what they're talking about. It can't "shut down the complete network".
>
> I've never objected this change/idea, just asking if this was possible and to what extent.
> What's wrong with getting informed? : -)

My apologies, your post was indeed a question not a statement.

59

On the Definition of Money and Bitcoin

SATOSHI RESPONDS to a thread regarding Bitcoin and Murray Rothbard's view on money. Rothbard was part of the Austrian School of Economy, an economic school of thought whose many founders originated from Vienna during the late 19th century. Its distinctive trait is its belief that the workings of the broader economy are the sum of all individuals' decisions and actions. In contrast to most other economic schools, the Austrian school believes that no central planners could possibly be able to properly estimate the resulting aggregate offer and also demand of any product or service. If central planners change any economic parameters that they control (typically applicable to interest rate under central banking), how could they properly estimate the resulting sum of all decisions on spending habits by consumers as well as investing decisions of businesses

and investors. No matter how many charts and statistics are collected, deviations between expectation and outcome are unavoidable and will lead to eventual disruptions.

Bitcoin does NOT violate Mises' Regression Theorem

Posted by xc, July 27, 2010, 02:09:27 AM

The Money Regression and Emergence of Money from the Barter Economy

The entire purpose of the regression theorem was to help explain an apparent paradox of money: how does money have value as a medium of exchange if it is valued because it serves as a medium of exchange? Menger and Mises helped break this apparent circularity by explaining the essential time component missing from the phrasing of the paradox.

As Rothbard explains in Man, Economy, and State (p 270), "…a money price at the *end* of day X is determined by the marginal utilities of money and the good as they existed at the *beginning* of day X. But the marginal utility of money is based, as we have seen above, on a *previously existing array of money prices*. Money is demanded and considered useful because of its already existing money prices. Therefore, the price of a good on day X is determined by the marginal utility of the good on day X and the marginal utility of money on day X, which last in turn depends on the prices of goods on day X – 1. **The economic analysis of money prices is therefore not circular. If prices today depend on the marginal utility of money today, the latter is dependent on money prices yesterday.**" [all emphasis added]

Rothbard then goes on to explain that in order for money to

emerge from a ***barter*** economy, it must have a preexisting commodity value. This commodity value arises from barter demand for the potential money in **direct consumption** (i.e. ornamentation). This value ***seeds*** future estimations of the value of the money as a medium of exchange. The natural market emergence of money is thus fully explained.

The Monetary Economy

However, once an economy has been monetized and a memory of price ratios for goods and services has been established, a money may lose its direct commodity value and still be used as a money (medium of indirect exchange). Rothbard explains (p 275):

"On the other hand, it does not follow from this analysis that if an extant money were to lose its direct uses, it could no longer be used as money. Thus, if gold, after being established as money, were suddenly to lose its value in ornaments or industrial uses, it would not necessarily lose its character as a money. Once a medium of exchange has been established as a money, **money prices continue to be set**. If on day X gold loses its direct uses, there will still be previously existing money prices that had been established on day X – 1, and these prices form the basis for the marginal utility of gold on day X. Similarly, the money prices thereby determined on day X form the basis for the marginal utility of money on day X + 1. From X on, gold could be demanded for its exchange value alone, and not at all for its direct use. Therefore, while it is absolutely necessary that a money **originate as a commodity** with direct uses, it is not absolutely necessary that the direct uses continue after the money has been established."

This explains the history of fiat currencies. They originally started off as simple names for weights of commodity money (silver) that developed out of the pre-monetary barter economy. Despite later losing their ties to direct commodity value through state interference, paper currency retained

status as money because of **memory of previous money prices**. This factor is so strong that the relationship between gold and the USD, for example, is somewhat inverted. Gold no longer circulates as a common medium of exchange. Prices are set in USD, not in gold. Most individuals wishing to trade in gold do so based on their knowledge of USD/gold price ratios. ("Hey, let me buy that $100 couch from you in gold?" "Ok, USD/gold is $1000/oz. Give me 1/10oz of gold.") Legal tender laws, state taxation, and the entire financial regulatory environment maintain this inertia of USD prices and make it challenging to return to gold money directly, despite the destructive inflationary nature of fiat currencies.

The Emergence of the Bitcoin Economy

The very first businesses in the Bitcoin economy were exchangers (NewLibertyStandard, BitcoinMarket, BitcoinExchange,....). This is not an accident, but flows from the analysis above. In order for Bitcoins to serve as a medium of exchange without commodity value for uses besides indirect exchange, there must be a translated knowledge of money prices. Market exchangers fill this gap and give Bitcoin users access to this knowledge. Bitcoins may therefore currently serve as a money intermediary for paypal dollars\pecunix\ euros. But why is there demand for Bitcoin over USD?? This is a subjective valuation arising from properties such as anonymity, decentralized system of clearance, cryptographic trust, predetermined and defined rate of growth, built in deflation, divisibility, low transaction fees, etc.... inherent to the Bitcoin system.

The essential point is that once exchange can occur between a money (USD) and Bitcoins, providers of goods have a means by which to value Bitcoins as a potential medium of exchange. The money regression is satisfied, because taken back far enough we reach traditional commodity money: BITCOINS -> USD -> MONETIZED GOLD & SILVER [start

monetary economy] -> [end barter economy] COMMODITY GOLD & SILVER.

Of course, if a major meltdown occurred and knowledge of all price ratios was wiped out, Bitcoin probably would NOT directly emerge as a money (assuming Bitcoins have limited value outside of exchange). Fiat currencies with zero direct barter value certainly would not. Commodities such as gold and silver that have widely recognized direct value in barter would likely emerge first. The economy would then be monetized with price ratios in gold and silver. Bitcoins then, being valued for intrinsic properties amenable to exchange, might then become prevalent in trade. Initially, creators of value would continue to make their price value ratios in terms of the true money (gold oz/BTC ratio), but with time Bitcoin prices (BTC) can emerge (see vekja.net as example). We are in this initial phase now.

Therefore, so long as exchange of BTC and USD/Euros/etc… occurs, knowledge of existing price ratios can be utilized in the Bitcoin economy. In time as Bitcoins become increasingly marketable, these fiat<->BTC price ratios will seed direct BTC price ratios. The Bitcoin Economy thus emerges. The Misean regression theorem is satisfied.

XC

edit: clarified possibility of direct emergence of bitcoin as money from barter economy.

Re: Bitcoin does NOT violate Mises' Regression Theorem

Posted by satoshi, August 27, 2010, 05:32:07 PM

As a thought experiment, imagine there was a base metal as

scarce as gold but with the following properties:

- boring grey in colour
- not a good conductor of electricity
- not particularly strong, but not ductile or easily malleable either
- not useful for any practical or ornamental purpose

and one special, magical property:

- can be transported over a communications channel

If it somehow acquired any value at all for whatever reason, then anyone wanting to transfer wealth over a long distance could buy some, transmit it, and have the recipient sell it.

Maybe it could get an initial value circularly as you've suggested, by people foreseeing its potential usefulness for exchange. (I would definitely want some) Maybe collectors, any random reason could spark it.

I think the traditional qualifications for money were written with the assumption that there are so many competing objects in the world that are scarce, an object with the automatic bootstrap of intrinsic value will surely win out over those without intrinsic value. But if there were nothing in the world with intrinsic value that could be used as money, only scarce but no intrinsic value, I think people would still take up something.

(I'm using the word scarce here to only mean limited potential supply)

Another post on the same subject:

Re: Bitcoin does NOT violate Mises' Regression Theorem

Posted by epaulson, August 17, 2010, 06:45:18 PM

There has been a lot of debate about what Bitcoins are – i.e. currency vs. commodity. Also there has been a lot of debate about inflation vs. deflation with respect to Bitcoins, whether people would lend them, at what rates, etc.

I think the most apt description of Bitcoins is that they are shares of stock in this communal Bitcoin enterprise we are undertaking. It is a lot like being part of a company (right now a very small company) and being paid in stock shares. There are a fixed number of Bitcoins, as there are a fixed number of shares in a company (barring new issues/etc.).

The primary value of Bitcoins right now is the hope that they will someday be worth significantly more than they are right now. For that to happen, the Bitcoin enterprise as a whole needs to gain collective value. We, as employee/owners of Bitcoin need to generate that added value. The most obvious way is to facilitate internet commerce by bartering shares of Bitcoin for other goods. The collective computational effort of all the employee/owners helps ensure that the barter is fair by keeping a record of each transaction. The individual efforts of some Bitcoiners are helping to make the barter of Bitcoins easier or more useful.

Regarding lending/borrowing of Bitcoins, to me it is analogous to lending/borrowing stock. The primary reason to borrow Bitcoins would be because you think they are overvalued and will be worth less when you have to return them. When you borrow the Bitcoins, you can sell them now (barter them now) and hopefully it will cost you less to buy them back at a later date so that you can return them to your lender (probably plus a fee).

In essence, Bitcoins are like a "direct public offering" of stock in the Bitcoin enterprise.

Re: Bitcoins are most like shares of common stock

Posted by satoshi, August 27, 2010, 04:39:26 PM

Bitcoins have no dividend or potential future dividend, therefore not like a stock.

More like a collectible or commodity.

60

On the Requirement of a Transaction Fee

SATOSHI'S ADVICE is to allow some transactions to be processed even if they do not have transaction fees. Currently, miners are still being rewarded with bitcoins, but this is scheduled to stop when all 21 million bitcoins have been mined—somewhere in the middle of the 22nd century. At that point, transaction fees might become mandatory so that miners are appropriately rewarded for use of their resources.

Always pay transaction fee?

Posted by jgarzik, September 07, 2010, 03:17:34 AM

To accurately reflect that processing a transaction has certain resource costs across the network, I propose that tx fee be

required for *every* transaction after X datetime (where X is a few months in the future).

Re: Always pay transaction fee?

Posted by satoshi, September 07, 2010, 04:32:21 PM

Another option is to reduce the number of free transactions allowed per block before transaction fees are required. Nodes only take so many KB of free transactions per block before they start requiring at least 0.01 transaction fee.

The threshold should probably be lower than it currently is.

I don't think the threshold should ever be 0. We should always allow at least some free transactions.

Re: Always pay transaction fee?

Posted by satoshi, September 08, 2010, 05:30:14 PM

Currently, paying a fee is controlled manually with the -paytxfee switch. It would be very easy to make the software automatically check the size of recent blocks to see if it should pay a fee. We're so far from reaching the threshold, we don't need that yet. It's a good idea to see how things go with controlling it manually first anyway.

It's not a big deal if we reach the threshold. Free transactions would just take longer to get into a block.

I did a rough tally of 4000 blocks from around 74000-78000. This is excluding the block reward transactions:

There were average 2 transactions per block, 17 transactions per hour, 400 transactions per day.

Average transaction bytes per block was 428 bytes, or 214 bytes per transaction.

The current threshold is 200KB per block, or about 1000 transactions per block. I think it should be lowered to 50KB per block. That would still be more than 100 times the average transactions per block.

The threshold can easily be changed in the future. We can decide to increase it when the time comes. It's a good idea to keep it lower as a circuit breaker and increase it as needed. If we hit the threshold now, it would almost certainly be some kind of flood and not actual use. Keeping the threshold lower would help limit the amount of wasted disk space in that event.

Re: Always pay transaction fee?

Posted by satoshi, September 23, 2010, 04:08:35 PM

> Quote from: satoshi on September 08, 2010, 05:30:14 PM
>
> The current threshold is 200KB per block, or about 1000 transactions per block. I think it should be lowered to 50KB per block. That would still be more than 100 times the average transactions per block.

I implemented this change in SVN rev 157.

The reason I previously made it so high was to allow very large transactions without hitting the transaction fee. The threshold was around 26,000 BTC for transactions made of 50 BTC generated coins. Even though it was 100 times easier to generate back then, only a few people ever encountered the fee at that level. The new threshold puts it at around 11,000 BTC for sending generated

coins. It would mostly only be reached with generated bitcoins. If you bought your bitcoins, they'll be denominated in larger transactions and won't be anywhere near the fee limit, unless you bought them in several hundred separate transactions. Even if you do reach the fee level, you only have to pay it once to bundle your little transactions together.

61

On Sites with CAPTCHA and Paypal Requirements

SOMEONE PROPOSES a few other possible ways in which Bitcoin could be useful. Satoshi's reply addresses the one regarding websites with both CAPTCHA and PayPal requirements.

The Niche List

Posted by kiba, September 23, 2010, 04:00:16 PM

This is Operation Economic Growth. Our mission is to grow the bitcoin economy by making everyone specialize in a narrow range of good and services.

Simply put, announce what you want to consume and I'll add it to the list. Somebody then will announce that he will try to enter that niche. There can be competition within niches too, but there are other niches to fill.

We'll hold those people "accountable" for their niches by plodding, encouraging, starting a thread and then getting disappointed when the service didn't come online etc.

Wanted Niches:

1. craigslist like classified ads for locale.
2. "Mechanical Turk"-like site that list simple jobs for people to do. Suggested by noagendamarket in the Stable Exchange Rate? topic of the Economic forum.
3. Beer supply store. Malt, yeasts, hops, etc.
4. Plant store for selling various herbs and stuff.
6. Hacker Academy. Free educational video. Flat tutiion fee classes. Pay as you go for personal tutors.
7. Dating site that accept bitcoins.
8. Easy encryption and backup service.

Niche filled or being worked on:

1. Advertising clearing house like http://projectwonderful.com. Suggested by mskwik.(I used projectwonderful to make tiny bit of money. I wonder if I can get more money from a bitcoin advertising clearing house) noagenda offered a large bounty on it and being worked on Biomike.
2. Download site like rapidshare and other crappy host. Inconvenient captcha and required paypal. Bitcoin can possibly take both roles and streamline the whole process. Suggested by Kiba. Taken by Hippich. Eventually sprawned 3 competitors.
3. Freelancer site. Taken by whichspace.
4. Pizza order system. You can order on the web, from the

commandline, from your smartphone, sms, etc. Taken by mizerydearia.

Re: The Niche List

Posted by satoshi, October 06, 2010, 11:10:31 PM

> Quote from: kiba on September 23, 2010, 04:00:16 PM
>
> 1. Download site like rapidshare and other crappy host. Inconvenient captcha and required paypal. Bitcoin can possibly take both roles and streamline the whole process.

Repeating myself here, but there is open source software for that, so it would just be a matter of bolting on a Bitcoin payment mechanism. One good one I found was Mihalism Multi Host. It's designed as a free host, so it would just need a few tweaks to loosen up restrictions consistent with paid use.

62

On Short Messages in the Block Chain

THE BLOCK CHAIN is the public ledger of all bitcoin transactions and is shared within the peer-to-peer network. At present, it contains only the transactions themselves. In this thread, someone proposes adding another piece of information within each transaction contained in the block chain that would be the equivalent of the "Note" section on bank checks. Unlike these, however, this note would be public and so visible to all. Satoshi expressed concerned that someone could publish in this note some information that was meant to be kept private, such as a customer account number.

Nevertheless, this feature is currently being considered for a future update of Bitcoin but is not yet available as of this writing. For now, only a third party service such as *blockchain.info* allows users to add extra text information, but it is not part of the block chain itself.

Miners have the ability to add some extra text in the block. As a matter of fact, the very first block created by Satoshi Nakamoto, block 0, has the following message in it:

"The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

The message is in ASCII encoded form but easy to extract for those who know how.

Suggestion: Allow short messages to be sent together with bitcoins ?

Posted by ShadowOfHarbringer, October 23, 2010, 03:11:17 PM

Bitcoin is great, but it misses one thing that usual bank transfers have: payment title.

Perhaps it should be possible to include short (<=512 bytes) message for each transaction.
The message could be encrypted with public/private keys so only the receiver can see its contents.

What do You think ?

PS.
I might be wrong, but the messages could also be used to increase randomness of hashing process by the way, couldn't they ? If not, never mind.

Re: Suggestion: Allow short messages to be sent together with bitcoins ?

Posted by satoshi, October 23, 2010, 07:02:57 PM

ECDSA can't encrypt messages, only sign signatures.

It would be unwise to have permanently recorded plaintext messages for everyone to see. It would be an accident waiting to happen.

If there's going to be a message system, it should be a separate system parallel to the bitcoin network. Messages should not be recorded in the block chain. The messages could be signed with the bitcoin address keypairs to prove who they're from.

63

On Handling a Transaction Spam Flood Attack

In this exchange, Satoshi talks about the introduction of changes in the software that will make it more economically difficult for someone to "spam" the network with multiple transactions.

Transaction / spam flood attack currently under way

Posted by jgarzik, November 19, 2010, 07:02:38 PM

Someone is apparently "testing" the main bitcoin network by flooding it with 0.01 BTC transactions from A->A and B->B,

where A and B are two random public keys. You can watch at http://theymos.ath.cx:64150/bbe

We've hit the free transaction limit on each block, for many blocks now – appears to be ~219 free transactions per block. "real" transactions do not appear DoS'd at this time, presumably due to logic that prioritizes, in part, based on transaction value.

<soapbox>
Free TX's are just asking for permanent level of spam. There should be a cost to each TX, even if it's only 0.001 BTC or so.
</soapbox>

Re: Transaction / spam flood attack currently under way

Posted by satoshi, November 19, 2010, 11:50:24 PM

Quote from: creighto on November 19, 2010, 08:29:12 PM

Perhaps in addition to the age priority rule recently implimented, there should be a minimum age rule without a transaction fee. Said another way, perhaps a generation rule that says that a free transaction must be 3 blocks deep before it can be transfered again for free. This will still allow real users to immediately spend new funds if they have to, while still permitting real users to reshuffle funds to suit their needs without an overhead cost. I think that this would significantly inhibit the type of spamming attack that is currently underway.

I'm doing something like that. Priority is a more formalised version of the concept you're describing.

> Quote from: FreeMoney on November 19, 2010, 05:39:44 PM
>
> As it stands now 3.15 has a lot of free transaction space and that space is given first to transactions with the highest [age]*[value]/[size] correct? Would it be reasonable to make some arbitrary portion of the free space require [age]*[value]/[size] > C ?
>
> Maybe set C so that a standard 1BTC transaction can get into the main free area on the next block. And a .1 can get in after waiting about 10 blocks. And make the area which allows [age]*[value]/[size] < C to let in about a dozen transactions or so.

Yes, like this. And the no-priority-requirement area is 3K, about a dozen transactions per block.

I just uploaded SVN rev 185 which has a minimal priority requirement for free transactions. Transaction floods are made up of coins that are re-spent over and over, so they depend on their own 0 conf transactions repeatedly. 0 conf transactions have 0 priority, so free transactions like that will have to wait for one transaction to get into a block at a time.

Version 0.3.15 doesn't write transactions using 0 conf dependencies unless that's all it has left, so normal users shouldn't usually have a problem with this.

I think this is a good compromise short of making the default fee 0.01. It's not so much to ask that free transactions can only be used to turn coins over so often. If you're using free transactions, you're taking charity and there has to be some limit on how often you can use it with the same coins.

We've always said free transactions may be processed more slowly. You can help ensure your transactions go through quickly by adding -paytxfee=0.01.

64

On Pool Mining Technicalities

IN THIS THREAD, the concept of how Bitcoin pool mining works is discussed and how it should be done to avoid cheaters becoming part of the pool without sharing. Today, mining pools are the largest contributors to mining. Mining pools were not initially a concept that Satoshi Nakamoto described. They came up later as a suggestion by someone on the forum when the difficulty of mining began to increase as interest in Bitcoin increased. The best analogy to a bitcoin mining pool is co-workers sharing lottery tickets.

Cooperative mining

Posted by slush, November 27, 2010, 01:45:41 PM

Hi all,

since the bitcointalk has been hacked few months ago I temporary lost the access to this forum. Now the access is recovered, but I'm not planning to continue with the pool support in this thread anymore, for more reasons. Mostly as newbies cannot post here, it cannot work as a full customer support; I've received many complains about this particularly. Then, it is spaghetti-style forum and it is very hard to follow the discussion.

Few days ago we started official pool support ticket system at http://support.bitcoin.cz. This support system is integrated also with support@bitcoin.cz, so writing an email to support@bitcoin.cz is the right place if you need authorized reply from one of pool admins anytime soon. Right now we're processing quite long backlog of emails there, but our target is to reply to all tickets in 24 hours. On http://support.bitcoin.cz is also knowledge base where we're filling more and more Q&A every day.

I would like to invite you also to IRC #mining.bitcoin.cz, where is quite many people online, ready to chat and provide basic help with all the stuff.

I'll leave this thread open for unofficial discussion, but it is out of my time possibilities to follow the discussion here.

Join us on http://mining.bitcoin.cz!
EDIT 27.12.2010: wiki page about pooled mining
EDIT 17.03.2011: DaCoinMinster published GreaseMonkey script which tweaks the pool website - it's 3rd party tool, use it on your own risk.

What is Pooled Mining?
Pooled mining is a way for multiple users to work together to mint bitcoins, and to share the benefits fairly.

Why do I need it?
Bitcoins are ordinarily only ever created in chunks of 50 at a time, with the whole 50 paid to a single person. Furthermore, the race to get the 50 BTC prize in a given block is highly competitive.

If you set out mining on your own, it may be a long time before you can make a return. Pooled mining allows you to receive smaller, more frequent, steadier payouts instead. If you have a slower computer, or a CPU miner, then pooled mining may be the only way that you will ever mint any bitcoins at all.

How do I get started?
You need less than 10 minutes to start mining in pool. Visit http://mining.bitcoin.cz and follow instructions.

Original post:

Once people started to use GPU enabled computers for mining, mining became very hard for other people. I'm on bitcoin for few weeks and didn't find block yet (I'm mining on three CPUs). When many people have slow CPUs and they mining separately, each of them compete among themselves AND against rich GPU bastards ;-), because everybody counts sha256 hashes from the same range. Two separate CPUs with 1000khash/s isn't the same as one 2000khash/s machine!. But new feature of the official bitcoin client called 'getwork' now enables work of many computers together, so they don't compete. Because there is now standalone CPU miner (thanks to jgarzik!) and 'getwork' patch is in official client now, I have an idea:

Join poor CPU miners to one cluster and increase their chance to find a block!

How that should work? There will be web page where you can register, enter your wallet address and get URL and your personal rpcuser/rpcpassword for your CPU/GPU miners. When you start own miner with these credentials, server will send you work which was not calculated yet by other members of cluster.

But when your client find winning hash, you do not get full reward for block (50BTC right now), but only proportional part, which you calculate. When you offer 1000khash/s for one day and whole cluster performance will be 20000khash/s and it takes two days to find a block, your reward will be (50/20/2=)1.25BTC.

Advantages? When you have poor standalone computer, you need to wait many weeks or even months for finding full 50BTC reward. When you join cluster like this, you will constantly receive small amount of bitcoins every day or week (depends on full cluster performance).

Disadvantages? You need to trust in central authority (me) that I don't steal block for myself. But I'm goofing around for few week and I'm amazed with bitcoin idea, so I don't plan to steal anybody right now :-).

~~Another possible problem is that somebody will ask for new work very often, but in fact he will not count real hashes. In this case it will look like he has very strong CPU and he should get big part of reward if cluster find a block. But there is a simple defense against cheaters: Central server sometimes send work which leads to 'winning' hash. Worker which don't return this hash as matching will be permanently banned (login/password and IP address).~~ This was succesfully solved by letting miners calculate proof-of-work. It is not anymore possible to be a part of cluster and not count hashes.

Are you interested in?

Re: Cooperative mining

Posted by ribuck, November 27, 2010, 10:21:02 PM

> Quote from: grondilu on November 27, 2010, 10:21:27 PM
>
> To me it seems that cooperative mining is a tough task, because the honnesty of participants has to be checked. What's preventing someone to run a modified version of the client, that would just keep generated bitcoin for himself, while receiving bitcoins from others ?

<sigh>

Either I haven't been very good at explaining why there's no possibility to cheat, or I'm wrong. But if I'm wrong, no-one has posted a specific objection. So I'll try to explain it again, by presenting a specific design to show that a dishonest client cannot cheat.

Suppose I operate a pooled mining server, and I recruit some clients who wish to pool their mining.

My server asks each client to do some hashing for it. It asks each client to submit any hashes they find that are above a certain threshold of difficulty. The server chooses a difficulty that is one-fortieth (1/40th) of the current "official" difficulty level.

My server gets a constant trickle of candidate hashes sent back by the remote mining clients. Every now and then, one of those hashes meets the official difficulty level and my server can generate a block, which earns my server 50 bitcoins.

I now distribute bitcoins to the remote mining clients, at the rate of one bitcoin for each hash that was submitted for the current block that was at or above 1/40th of the official difficulty level.

In the long run, I would expect to distribute 40 coins out of every 50 that my server generates, although there will be some fluctuation from block to block. Nothing in this scheme requires the clients to be honest, because there is no way that a dishonest client can cheat!

The client is calculating hashes that will generate 50 BTC for my server. Those same hashes are not of any use to a dishonest client. They cannot be used to generate 50 BTC for the dishonest client, because a different hash code is needed to encode the payment of the generated bitcoins to someone else. And if the dishonest client tries to cheat by generating hashes that will pay the generated bitcoins to themselves, then the hash codes they submit won't validate at my server and I won't distribute any share of the payouts to them.

So this scheme requires absolutely no trust of the client.

This scheme also does not require the mining client to have faith that the server is honest. If the server advertises that it is paying out 1BTC for each hash that is at least 1/40th of the official difficulty level, then every client that submits an "easy" hash for a block that was generated can check that they received their bitcoin. Any fraud would show up immediately.

Re: Cooperative mining

Posted by satoshi, November 28, 2010, 04:03:30 PM

---

ribuck's description is spot on.

Pool operators can modify their getwork to take one additional parameter, the address to send your share to.

The easy way for the pool operator would be to wait until the next block is found and divy it up proportionally as:

user's near-hits/total near-hits from everyone

That would be easier and safer to start up. It also has the advantage that multiple hits from the same user can be combined into one transaction. A lot of your hits will usually be from the same people.

The instant gratification way would be to pay a fixed amount for each near-hit immediately, and the operator takes the risk from randomness of having more or less near-hits before a block is found.

Either way, the user who submits the hit that solves the block should get an extra amount off the top, like 10 BTC.

New users wouldn't really even need the Bitcoin software. They could download a miner, create an account on mtgox or mybitcoin, enter their deposit address into the miner and point it at anyone's pool server. When the miner says it found something, a while later a few coins show up in their account.

Miner writers better make sure they never false-positive near-hits. Users will depend on that to check if the pool operator is cheating them. If the miner wrongly says it found something, users will look in their account, not find anything, and get mad at the pool operator.

65

On WikiLeaks Using Bitcoin

In late 2010, governments of the world were attempting to exert pressure on WikiLeaks by cutting off its sources of funding, which consisted primarily of online donations through credit card payments and PayPal.

When PayPal announced that it would block service to WikiLeaks, Satoshi stated that, in his opinion, Bitcoin was not yet ready to act as a replacement.

http://www.wired.com/threatlevel/2010/12/paypal-wikileaks/

Wikileaks contact info?

Posted by genjix, November 10, 2010, 12:49:16 PM

Hey, I wanted to send a letter to Wikileaks about Bitcoin since unfortunately they've had several incidents where their funds have been seized in the past.
http://wikileaks.org/media/support.html

Anyone know where to send a message to them?

Re: Wikileaks contact info?

Posted by wumpus, December 04, 2010, 08:47:59 AM

Paypal just blocked them, and they're trying to get other US banks do the same. This would be a great moment to open bitcoin donations.

Re: Wikileaks contact info?

Posted by RHorning, December 04, 2010, 10:17:44 PM

> Quote from: Hal on December 04, 2010, 08:43:07 PM
>
> Looking on the bright side, if Bitcoin did get known as the Wikileaks currency, attacked by governments all over the world, at least we'd get our Wikipedia page back!

This is so true. There certainly wouldn't be a shortage of "reliable sources" about Bitcoins at that point. I think it would likely show up on the front page of most newspapers and be talked about extensively on both radio talk shows and the other broadcast networks too.

For myself, I'm getting to the point to say "bring it on" in regards to Wikileaks. Note that I'm using my real name here instead of a psuedonym and I'm willing to personally say "bring it on" in terms of being associated with Bitcoins as a project. I've had police come into my house without my permission already and do all kind of stupid stuff, so for me that line being crossed has already happened. I am also connected to enough people politically that if something was to happen to me that it would be noted and things would happen too.

It is the morally correct thing to be supporting Wikileaks, and if they'll take a few of my bitcoins, I not only want to donate but to let the world know that they can donate to Wikileaks through Bitcoins as well.

I can't speak for everybody here in the Bitcoins community but I am speaking for myself on this matter, and I'm not afraid of anything that the U.S. government might do to me if I was associated with backing Wikileaks financially. If anything, it would show that I no longer live under a constitutional government any more. If the U.S. government wants to tip their hand to expose themselves in that way, so be it. If the U.S. government kills me or puts me in jail, I'll certainly set a way for this community to find out. I really don't think it would come to that either, but I don't care if it did.

If I have to "vote" on this matter, I would encourage the Bitcoin community to take up the plate like we did with the EFF and encourage Wikileaks to put up a Bitcoin address on their website for donations. It would bring in some new blood into the Bitcoin community regardless, and it might be beneficial to Wikileaks as well. Leave it to Wikileaks to see if they want to use Bitcoins or not. In terms of governmental review of Bitcoins, we know that is going to happen sooner or later, so why are we fighting that inevitable result? Anything other than a low-key investigation is only going to make more people interested in

Bitcoins, which is only going to help the project even more. It can't be killed as a project, only slowed down a little bit in its growth at this point and more likely its adoption would be accelerated by any kind of publicity that would happen.

The only possible concern I would have is over how sound the protocol itself is right now. If anything, a major flux of new people into Bitcoins would help there too, and the worst that could happen is that Bitcoins itself would be broken in some way where a new cryptocurrency would have to be created fixing the problems of Bitcoins. It is the idea of cryptocurrency that would then persist, and it is incredibly hard to censor an idea.

Basically, bring it on. Let's encourage Wikileaks to use Bitcoins and I'm willing to face any risk or fallout from that act.

– Robert S. Horning
Logan, Utah

Re: Wikileaks contact info?

Posted by satoshi, December 05, 2010, 09:08:08 AM

Quote from: RHorning on December 04, 2010, 10:17:44 PM

Basically, bring it on. Let's encourage Wikileaks to use Bitcoins and I'm willing to face any risk or fallout from that act.

No, don't "bring it on".

The project needs to grow gradually so the software can be strengthened along the way.

I make this appeal to WikiLeaks not to try to use Bitcoin. Bitcoin is a small beta community in its infancy. You would not stand to get more than pocket change, and the heat you would bring would likely destroy us at this stage.

66

On a Distributed Domain Name Server

SOMEONE SUGGESTED creating a Bitcoin clone (an alternative currency) that would allow for a distributed peer-to-peer domain name server system (DNS). In addition to containing currencies, transactions stored in the block chain would also contain DNS information and could be updated with new transactions.

Such alternative currency exists today and is called Namecoin (see http://www.namecoin.org/), which allows people to register domain names ending in ".bit" and associate them with an IP address. Satoshi shares his thoughts regarding this type of system here. One of the major benefits of such decentralized domain name servers would bypass a government attempt at disrupting Internet communications to its citizens, as we have seen occur in Egypt in 2011.

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 09, 2010, 09:02:42 PM

I think it would be possible for BitDNS to be a completely separate network and separate block chain, yet share CPU power with Bitcoin. The only overlap is to make it so miners can search for proof-of-work for both networks simultaneously.

The networks wouldn't need any coordination. Miners would subscribe to both networks in parallel. They would scan SHA such that if they get a hit, they potentially solve both at once. A solution may be for just one of the networks if one network has a lower difficulty.

I think an external miner could call getwork on both programs and combine the work. Maybe call Bitcoin, get work from it, hand it to BitDNS getwork to combine into a combined work.

Instead of fragmentation, networks share and augment each other's total CPU power. This would solve the problem that if there are multiple networks, they are a danger to each other if the available CPU power gangs up on one. Instead, all networks in the world would share combined CPU power, increasing the total strength. It would make it easier for small networks to get started by tapping into a ready base of miners.

Re: BitDNS and Generalizing Bitcoin

Posted by nanotube, December 09, 2010, 09:20:40 PM

> Quote from: satoshi on December 09, 2010, 09:02:42 PM
>
> I think it would be possible for BitDNS to be a completely separate network and separate block chain, yet share CPU power with Bitcoin. The only overlap is to make it so miners can search for proof-of-work for both networks simultaneously.

sounds excellent in theory...

> Quote from: satoshi on December 09, 2010, 09:02:42 PM
>
> The networks wouldn't need any coordination. Miners would subscribe to both networks in parallel. They would scan SHA such that if they get a hit, they potentially solve both at once. A solution may be for just one of the networks if one network has a lower difficulty.
>
> I think an external miner could call getwork on both programs and combine the work. Maybe call Bitcoin, get work from it, hand it to BitDNS getwork to combine into a combined work.

---

seems that the miner would have to basically do "extra work". and if there's no reward from the bitdns mining from the extra work (which of course, slows down the main bitcoin work), what would be a miner's incentive to include bitdns (and whatever other side chains) ?

very curious to hear your further thoughts on this. : -)

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 09, 2010, 10:46:50 PM

> Quote from: nanotube on December 09, 2010, 09:20:40 PM
>
> seems that the miner would have to basically do "extra work". and if there's no reward from the bitdns mining from the extra work (which of course, slows down the main bitcoin work), what would be a miner's incentive to include bitdns (and whatever other side chains) ?

The incentive is to get the rewards from the extra side chains also for the same work.

While you are generating bitcoins, why not also get free domain names for the *same work*?

If you currently generate 50 BTC per week, now you could get 50 BTC and some domain names too.

You have one piece of work. If you solve it, it will solve a block from both Bitcoin and BitDNS. In concept, they're tied together by a Merkle Tree. To hand it in to Bitcoin, you break off the BitDNS branch, and to hand it in to BitDNS, you break off the Bitcoin branch.

In practice, to retrofit it for Bitcoin, the BitDNS side would have to have maybe ~200 extra bytes, but that's not a big deal. You've been talking about 50 domains per block, which would dwarf that little 200 bytes per block for backward compatibility. We could potentially schedule a far in future block when Bitcoin would upgrade to a modernised arrangement with the Merkle Tree on top, if we care enough about saving a few bytes.

Note that the chains are below this new Merkle Tree. That is, each of Bitcoin and BitDNS have their own chain links inside their blocks. This is inverted from the common timestamp server arrangement, where the chain is on top and then the Merkle Tree, because that creates one common master chain. This is two timestamp servers not sharing a chain.

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 10, 2010, 05:29:28 PM

Piling every proof-of-work quorum system in the world into one dataset doesn't scale.

Bitcoin and BitDNS can be used separately. Users shouldn't have to download all of both to use one or the other. BitDNS users

may not want to download everything the next several unrelated networks decide to pile in either.

The networks need to have separate fates. BitDNS users might be completely liberal about adding any large data features since relatively few domain registrars are needed, while Bitcoin users might get increasingly tyrannical about limiting the size of the chain so it's easy for lots of users and small devices.

Fears about securely buying domains with Bitcoins are a red herring. It's easy to trade Bitcoins for other non-repudiable commodities.

If you're still worried about it, it's cryptographically possible to make a risk free trade. The two parties would set up transactions on both sides such that when they both sign the transactions, the second signer's signature triggers the release of both. The second signer can't release one without releasing the other.

Re: BitDNS and Generalizing Bitcoin

Posted by Hal, December 10, 2010, 07:14:04 PM

Satoshi, are you endorsing the idea that additional block chains would each create their own flavor of coins, which would trade with bitcoins on exchanges? These chain-specific coins would be used to reward miners on those chains, and to purchase some kinds of rights or privileges within the domain of that chain?

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 10, 2010, 07:55:12 PM

> Quote from: Hal on December 10, 2010, 07:14:04 PM
>
> additional block chains would each create their own flavor of coins, which would trade with bitcoins on exchanges? These chain-specific coins would be used to reward miners on those chains, and to purchase some kinds of rights or privileges within the domain of that chain?

Right, the exchange rate between domains and bitcoins would float.

A longer interval than 10 minutes would be appropriate for BitDNS.

So far in this discussion there's already a lot of housekeeping data required. It will be much easier if you can freely use all the space you need without worrying about paying fees for expensive space in Bitcoin's chain. Some transactions:

Changing the IP record.

Name change. A domain object could entitle you to one domain, and you could change it at will to any name that isn't taken. This would encourage users to free up names they don't want anymore. Generated domains start out blank and the miner sells it to someone who changes it to what they want.

Renewal. Could be free, or maybe require consuming another domain object to renew. In that case, domain objects (domaincoins?) could represent the right to own a domain for a year. The spent fee goes to the miners in the next block fee.

Re: BitDNS and Generalizing Bitcoin

> Posted by Hal, December 10, 2010, 08:12:02 PM
>
> OK so if there are going to be bitdnscoins (aka DCCs, DomainChain Coins) then they have to be useful for

something. Otherwise every BitDNS miner is going to fill every block with his own domain name registrations, rather than replace one with someone else's registration in exchange for a transaction fee in a useless currency.

The rules have to be that you have to spend a certain amount of bitdnscoins/DCCs in order to register your names and/or do other BitDNS transactions. That's the only way to make this alternative currency desirable and valuable.

(Well we could do like Bitcoin and say there would only ever be 22 million DCCs ever created, so they'd get valuable from scarcity just like bitcoins. But that seems weak.)

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 10, 2010, 08:19:39 PM

I agree. All transactions, IP changes, renewals, etc. should have some fee that goes to the miners.

You might consider a certain amount of work to generate a domain, instead of a fixed total circulation. The work per domain could be on a schedule that grows with Moore's Law. That way the number of domains would grow with demand and the number of people using it.

Re: BitDNS and Generalizing Bitcoin

Posted by dtvan, December 11, 2010, 07:43:08 AM

After reading through this whole thread, I've got a couple of comments that I think would be helpful:

1) Everyone in the thread seems intent on replacing the entire DNS infrastructure in one fell swoop, which I think is the wrong approach. The real problem with the DNS system as it exists today is that somebody has to own the root. At the end of the day, you have to trust ICANN. What the DomainChain/BitDNS system should strictly focus on is establishing ownership of domain names. All it needs to track is that the holder of Key A owns domain foo.bar. Once we've established this shared trust, we can support many different DNS infrastructures that can be implemented independently from this project. Whatever new systems are created use DomainChain/BitDNS to establish which key owns the domain, and only allows that individual to insert records for that domain. This works out well, since all participants in the system can validate that the record they've looked up is valid. Right now it is easy to get bogged down in all the details of managing DNS records, when all we need to do is establish a trusted, distributed authority that can form the root of DNSSEC, some new p2p DNS, or whatever.

I'm also thinking this could be used to solve the CA problem with HTTPS, since signing your certificate with the same key would prove that I've reached the correct server. But I digress . . .

2) Limiting the TLDs should be a requirement. If this system doesn't inter-operate with the existing DNS infrastructure by preventing name collisions, it will undermine the trust you are trying to generate. Even I'm not sure I'm ready to sign up for a distributed DNS system if someone new can pick up www.mylocalbank.com and cause havok. I'd humbly suggest .web as the TLD to use, but anything will work as long as it is short and not currently in use.

Right now the focus should be on getting this up and running in a way that doesn't conflict with the existing system. If this system becomes dominant at some point and needs to tackle

additional TLDs, that is a "problem" that can be dealt with then.

3) Personally, I think expiring domain names are the way to go. Even with relatively expensive renewals today, there is still a ton of junk out there. I can't imagine how bad it would be if you owned a domain forever. It isn't asking much to say that you have to renew your domain periodically to keep it, especially since this shouldn't be the ripoff that the existing system is today.

I'd like to close out by saying that this is really exciting stuff. I've read a number of different ideas about how to solve the DNS problem, and this is the first one I've seen that could actually solve it (and not just replace ICANN with pick-your-new-benevolent-dictator).

Re: BitDNS and Generalizing Bitcoin

Posted by satoshi, December 10, 2010, 08:19:39 PM

@dtvan: all 3 excellent points.
1) IP records don't need to be in the chain, just do registrar function not DNS. And CA problem solved, neat.
2) Pick one TLD, .web +1.
3) Expiration and significant renewal costs, very important.

> Quote from: joe on December 11, 2010, 10:53:58 AM
>
> However, thinking more about this now I support inclusion of additional coinbases / tracking systems in the main network. The reason for doing this is so as not to water down CPU power into multiple networks. We want one strong network, so the network should be versatile.

Avoiding CPU power fragmentation is no longer a

reason. Independent networks/chains can share CPU power without sharing much else. See: http://bitcointalk.org/index.php?topic=1790.msg28696#msg28696 and http://bitcointalk.org/index.php?topic=1790.msg28715#msg28715

(Note, two of Satoshi's earlier posts are included in this thread)

Another thread came up on the same subject:

Re: Fees in BitDNS confusion

Posted by galeru, December 09, 2010, 07:45:38 PM

A bunch of the current debate about including BitDNS or BitX makes assumptions that miners will include transactions or not based on some rather fine-grained conditions, while none of the standard code includes any sort of implementation that allows non-programmers to make decisions like that. How will I, the user, figure out what sort of fees have to go into a transaction?

Re: Fees in BitDNS confusion

Posted by jgarzik, December 09, 2010, 11:07:04 PM

> Quote from: davout on December 09, 2010, 09:08:55 PM
>
> Just wondering about the following example :
> I broadcast a transaction, sending X coins to some address.
> Doesn't get included in blocks for a while because I don't include a fee.
>
> Do I have a way to cancel it and broadcast it again with a fee this time ?

See the discussion of locktime (https://bitcointalk.org/index.php?topic=1786.msg22119#msg22119)

for transaction replacement.

Re: Fees in BitDNS confusion

Posted by satoshi, December 09, 2010, 11:58:54 PM

Not locktime.

There's a possible design for far in the future:

You intentionally write a double-spend. You write it with the same inputs and outputs, but this time with a fee. When your double-spend gets into a block, the first spend becomes invalid. The payee does not really notice, because at the moment the new transaction becomes valid, the old one becomes invalid, and the new transaction simply takes its place.

It's easier said than implemented. There would be a fair amount of work to make a client that correctly writes the double-spend, manages the two versions in the wallet until one is chosen, handles all the corner cases. Every assumption in the existing code is that you're not trying to write double-spends.

There would need to be some changes on the Bitcoin Miner side also, to make the possibility to accept a double-spend into the transaction pool, but only strictly if the inputs and outputs match and the transaction fee is higher. Currently, double-spends are never accepted into the transaction pool, so every node bears witness to which transaction it saw first by working to put it into a block.

67

On a *PC World* Article on Bitcoin and WikiLeaks Kicking the Hornet's Nest

IN LIGHT OF THE ISSUES that WikiLeaks was bringing to the fore and the possible role that Bitcoin could play in helping with WikiLeaks's funding, *PC World* posted an article related to Bitcoin. Clearly, Bitcoin was starting to garner attention in the press. What is interesting is Satoshi's comment referencing WikiLeaks "kicking the hornet's nest". Here is a link to the article in *PC World,* followed by Satoshi's comment.

"Could the Wikileaks Scandal Lead to New Virtual Currency?"

http://www.pcworld.com/article/213230/could_wikileaks_scandal_lead_to_new_virtual_currency.html

Re: PC World Article on Bitcoin

Posted by satoshi, December 11, 2010, 11:39:16 PM

It would have been nice to get this attention in any other context. WikiLeaks has kicked the hornet's nest, and the swarm is headed towards us.

68

Satoshi's Last Forum Post: Release of Bitcoin 0.3.19

NINETEEN HOURS after the "Hornet's nest" post, Satoshi writes his last forum post before retreating from "public life". He posted it on *bitcointalk.org*. It was his last forum post prior to that made in March 2014 on the p2pfoundation forum.

Added some DoS limits, removed safe mode (0.3.19)

Posted by satoshi, December 12, 2010, 06:22:33 PM

There's more work to do on DoS, but I'm doing a quick build of what I have so far in case it's needed, before venturing into more complex ideas. The build for this is version 0.3.19.

- Added some DoS controls
As Gavin and I have said clearly before, the software is not at all resistant to DoS attack. This is one improvement, but there are still more ways to attack than I can count.

I'm leaving the -limitfreerelay part as a switch for now and it's there if you need it.

- Removed "safe mode" alerts
"safe mode" alerts was a temporary measure after the 0.3.9 overflow bug. We can say all we want that users can just run with "-disablesafemode", but it's better just not to have it for the sake of appearances. It was never intended as a long term feature. Safe mode can still be triggered by seeing a longer (greater total PoW) invalid block chain.

Builds:
http://sourceforge.net/projects/bitcoin/files/Bitcoin/bitcoin-0.3.19/

As of March 14, 2014, this is Satoshi's last post on p2pfoundation on Friday March 7th, 2014:

I am not Dorian Nakamoto

This was in reaction to a magazine's publication that claimed to have identified Bitcoin Satoshi Nakamoto as a man living in California with the name of Dorian Satoshi Nakamoto.

69

Emails to Dustin Trammell

THE FOLLOWING are direct email exchanges between Satoshi Nakamoto and Dustin Trammell that Dustin Trammell has generously made available for publication.

Email 1—Timestamp and bitcoin maturity

This first exchange concerns timestamp document services and Bitcoin mining maturity. These were discussed later in a public forum, but Satoshi addressed them first in a private conversation with Dustin Trammell.

From: "Satoshi Nakamoto" satoshi@vistomail.com

To: dtrammell@dustintrammell.com
Date: Tue, 13 Jan 2009 02:33:28 +0800
Subject: Re: Bitcoin v0.1 released

> I'm currently reading through your paper. At the timestamp server section you mention newspapers and usenet, so I thought you might be interested in this if you have not seen it already:
>
> http://www.publictimestamp.org/

Thanks, I hadn't seen that yet. It looks very well presented.

There was an older one that's been running for a long time that publishes its hashes to Usenet. I'm surprised this one isn't using Usenet, although it is kind of difficult to get access to post to Usenet in an automated way these days. If they can get a magazine or newspaper to publish their hashes, it would work a lot easier in court for their purposes. Bitcoin and all timestamp servers share the basic functionality of periodically collecting things into blocks and hashing them into a chain.

> By the way, I'm also currently running the alpha code on one of my workstations. So far it has two "Generated" messages, however the "Credit" field for those is 0.00 and the balance hasn't changed. Is this due to the age/maturity requirement for a coin to be valid?

Right, the credit field stays 0.00 until it matures, then it'll be

50.00. Do you think it would be clearer if I left the credit field blank until it matures? I should put some text in the transaction details (when you double click on it) explaining how it works. (was it obvious you can doubleclick on a line for details?)

Be sure to upgrade to v0.1.3 if you haven't already. This version has really stabilized things.

Satoshi

Email 2—Follow up

From: "Satoshi Nakamoto" satoshi@vistomail.com

To: dtrammell@dustintrammell.com
Date: Tue, 13 Jan 2009 15:55:13 +0800
Subject: Re: Bitcoin v0.1 released

> It actually posts the hash blocks to a Google Group called 'proof-hashes', so similar result as if it were posting to Usenet.
>
> http://groups.google.com/group/proof-hashes
>
> Since I run that group, and it's sole purpose is to archive proof-of-work hashes, feel free to join an account to have your system post there as well if you like.

Sweet, I was looking for a group like that on Usenet at one point to see what I would use if I needed, and nothing really fit. I'm sure Google groups is a lot easier to post to.

There are some scenarios where a Usenet or Google group could be used as a supplemental defence. Bitcoin is at its most vulnerable in the beginning when the total network CPU power is small. That's offset by the fact that the incentive to attack it is also low when it's small.

Hopefully the easy solution of just growing up and getting past that stage will work. If not, there are ways a Google group could help, if it really came to that.

> Electronic currency and cryptography are two things that I am very interested in so as you would assume I was drawn to this project immediately when I saw it posted to the Cryptography email list. Feel free to ping me for feedback or to test out new features, I'll be happy to help out.

We definitely have similar interests!

You know, I think there were a lot more people interested in the 90's, but after more than a decade of failed Trusted Third Party based systems

(Digicash, etc), they see it as a lost cause. I hope they can make the distinction, that this is the first time I know of that we're trying a non-trust based system.

> When the coins mature, will that generate a new 'credit' transaction, or will the existing generation transaction line's credit field be updated?

The existing transaction line will change.

> Upon opening version 0.1.3, all four of my transaction entries still say 'unconfirmed', but now the Descriptions say 'Generated (not accepted)'.
>
> Does this mean that some other node had extended the chain first and my coins were generated in a dead branch? If so, why did the previous instance of the software not detect this immediately and begin generating coins in the winning branch? Bug in 0.1.0?

You're right, sorry about that. It's the bug that was fixed in 0.1.3.

The communications thread would get blocked, so you would make connections, but they would go silent after a while. When you found a block, you couldn't broadcast it to the network, so it didn't get into the chain. You weren't receiving anything either to know that the network had gone on without you, until you restarted it.

The bug is also what caused bitcoin.exe to fail to exit. The communications thread was blocked and failed to exit. Bitcoin does a careful shutdown in case it might be in the middle of an important transaction, but actually it's completely safe to kill it.

This is all fixed in 0.1.3. If you give me your IP, I'll send you some coins.

> One other question I had... What prevents the single node with the most CPU power from generating and retaining the majority of the BitCoins?
>
> If every node is working independently of all others, if one is significantly more powerful than the others, isn't it probable that this node will reach the proper conclusion before other nodes? An underpowered node may get lucky once in a while, but if they are at a significant horsepower advantage I would expect the majority of BitCoins to be generated by the most powerful node.

It's not like a race where if one car is twice as fast, it'll always win. It's an SHA-256 that takes less than a microsecond, and each guess has an independent chance of success. Each computer's chance of finding a hash collision is linearly proportional to it's CPU power. A computer that's half as fast would get half as many coins.

> I'll watch this instance and see how it goes...

Let me know how it goes. If you have any trouble with it, send me your debug.log file. I can often figure out what went wrong just from that.

Satoshi

Email 3—On Bitcoin's potential

This exchange seems to indicate that Satoshi was not expecting such rapid acceptance of Bitcoin.

From: "Satoshi Nakamoto" satoshi@vistomail.com

To: dtrammell@dustintrammell.com
Date: Fri, 16 Jan 2009 03:15:14 +0800
Subject: Re: Bitcoin v0.1 released

> I've had that address for a while though so hopefully my dhcp client is being successful at renewing and not losing my address. It does change from time to time, but that address should be good for a while.

There's at least one node who's inbound IP keeps changing all the time within the same class B. Maybe every time the program is run. I wasn't expecting that.

Do you mind if I CC the rest of this to bitcoin-list or Cryptography?

BTW, bitcoin-list is:

bitcoin-list@lists.sourceforge.net
Subscribe/unsubscribe page:
http://lists.sourceforge.net/mailman/listinfo/bitcoin-list
Archives:
http://sourceforge.net/mailarchive/forum.php?forum_name=bitcoin-list

> Dustin D. Trammell wrote:
>
> > Satoshi Nakamoto wrote:
> >
> > You know, I think there were a lot more people interested in the 90's, but after more than a decade of failed Trusted Third Party based systems (Digicash, etc), they see it as a lost cause.
> >
> > I hope they can make the distinction that this is the first time
> >
> > I know of that we're trying a non-trust-based system.
>
> Yea, that was the primary feature that caught my eye. The real trick will be to get people to actually value the BitCoins so that they become currency.

Hal sort of alluded to the possibility that it could be seen as a long-odds investment. I would be surprised if 10 years from now we're not using electronic currency in some way, now that we know a way to do it that won't inevitably get dumbed down when the TTP gets cold feet.

Even if it doesn't take off straight away, it's now available for use by the next guy who comes up with a plan that needs some kind of token or electronic currency. It could get started in a closed system or narrow niche like reward points, donation tokens, currency for a game or micropayments for adult sites. Once it gets bootstrapped, there are so many applications if you could effortlessly pay a few cents to a website as easily as dropping coins in a vending machine.

It can already be used for pay-to-send e-mail. The send dialog is resizeable and you can enter as long of a message as you like. It's sent directly when it connects. The recipient doubleclicks on the transaction to see the full message. If someone famous is getting more e-mail than they can read, but would still like to have a way for fans to contact them, they could set up Bitcoin and give out the IP address on their website. "Send X bitcoins to my priority hotline at this IP and I'll read the message personally."

Subscription sites that need some extra proof-of-work for their free trial so it doesn't cannibalize subscriptions could charge bitcoins for the trial.

Satoshi

Email 4—On attacks and IP addresses involved in sending bitcoins

From: "Satoshi Nakamoto" satoshi@vistomail.com
To: dtrammell@dustintrammell.com
Date: Fri, 16 Jan 2009 03:46:30 +0800

Subject: Re: A few thoughts…

I group attacks into two classes:

1) Attacks that can only be done by someone actually in the chain of communication

2) Attacks that can be done by anyone on the Internet from anywhere

Type 1 exposes you to people in your house or company on your local LAN, admins at ISPs in between, and the LAN on the recipient's side. Type 2 exposes you to a billion people who can self-select to be attackers and get economy of scale when they develop one technique to attack multiple victims.

Sending by IP requests a new public key, so yes, it's vulnerable to type 1 man-in-the-middle. If that's a concern, sending to a

Bitcoin address doesn't have that vulnerability, although there's a small privacy tradeoff. I have a feeling most of the time people will get Bitcoin addresses off of non-SSL websites and unsigned cleartext e-mail, which is already vulnerable to type 1 and type 2 through DNS poisoning.

One solution would be to use both the IP and Bitcoin addresses when sending (maybe 1.2.3.4-1Kn8iojk...), where the recipient uses the public key of the Bitcoin address to sign the new public key to prove that you're sending to who you think you are. If the system starts to be used for real business purposes, I will certainly implement that. Another solution is to use SSL.

For now, it's pretty obvious that if you send to an IP, you didn't give any other identifying information about the recipient, so you're blindly sending to whoever answers that IP.

Another feature for later is an option to encrypt your wallet.

If I understand how that is done correctly, you just compute the

> transaction into the block chain and let the intended recipient 'discover' it, correct?

That's correct.

> An alternative could be to allow the network nodes to provide a resolution service, where they ask around for the network address of a
>
> BitCoin address, and if that node is online, once a consensus is agreed upon by the network for that address the sending BitCoin application connects directly there.

It would be nice to only need the Bitcoin address and have the IP worked out behind the scenes. Might have privacy or denial of service issues. Certainly before another sending method is implemented, there's plenty of time now to fully think through the design and make sure it's the best way.

Satoshi

Email 5—On potential loss and the need for backup

From: "Satoshi Nakamoto" satoshi@vistomail.com
To: dtrammell@dustintrammell.com
Date: Sat, 17 Jan 2009 02:32:48 +0800
Subject: Re: A few thoughts…

> One thing that came to mind on this topic is the potential for BitCoin
>
> loss if you have a system failure. The application doesn't seem to store any data in the directory that it runs in, so I assume it's stored in the registry and other places (haven't cracked out ProcessExplorer yet to check myself), so it may be a good

> idea to have the application be able to export everything that it needs for recovery to a file that could be backed up off of the system.

The files are in "%appdata%\Bitcoin", that's the directory to backup. The data is stored in a transactional database DBM, so it should be safe from loss if there's a crash or power failure.

%appdata% is per-user access privilege. Most new programs like Firefox store their settings files there, despite the headwind of Microsoft changing the directory name with every Windows release and being full of spaces and so long it runs off the screen.

> One other thing I noticed today is that if you close the application it doesn't appear to cleanly close it's network sockets (TCP RST's start flying). Probably an item for the low-priority todo list (:

Just now added code to the next release for that.

Satoshi

Email 6—Satoshi sent bitcoins

From: "Satoshi Nakamoto" satoshi@vistomail.com
To: dtrammell@dustintrammell.com
Date: Mon, 19 Jan 2009 00:54:32 +0800
Subject: Re: Bitcoin Transfer

It should be your Bitcoin address at home that you received it with. There's no way for it to know who it's from, so the best it can do is tell which of your addresses it was received on.

You can create multiple addresses and give a different address to each person and label them to help figure out who's sending to you.

It doesn't know any names other than what you tell it. The name printed there is what's associated in your address book for that address, either under the Address Book button or the "Change..." button to the right of your Bitcoin address.

Hey Satoshi,

After that first transfer of 25.00, you didn't send me another 100.00

did you? I sent myself 100.00 from my BitCoin application at work to my one at home using the BitCoin address rather than by IP. My application at home has a 100.00 transfer received, however it's transaction details say

"Received with: Satoshi
12higDjoCCNXSA95xZMWUdPvXNmkAduhWv".

That is not my BitCoin address from work, so I assume this means that I received the payment encoded with a block that was computed by your client? If so, how did it know your name in addition to the BitCoin address that generated it? I don't recall there being a place in my application to even put my name.

–

Dustin D. Trammell
dtrammell@dustintrammell.com
http://www.dustintrammell.com

70

Last Private Correspondence

ALLEGEDLY, Gavin Andresen is the last person to have had a private exchange with Satoshi Nakamoto. It took place four months after his final, December 2010 forum post on *bitcointalk.org*. Gavin Andresen had a few private email exchanges with Satoshi after his retreat from public life. However, Gavin has decided that only the very last email from Satoshi will be shared publicly.

Email from Satoshi

Gavin Andresen, April 26, 2011

On Tue, Apr 26, 2011, Satoshi Nakamoto <satoshin@gmx com> wrote:

I wish you wouldn't keep talking about me as a mysterious shadowy figure, the press just turns that into a pirate currency angle. Maybe instead make it about the open source project and give more credit to your dev contributors; it helps motivate them.

You must've read the Forbes article . . . yeah, I'm not happy with the 'wacky pirate money' tone, either.

More credit for the rest of the contributors is a very good idea.

On a completely different subject: I did something that I hope turns out to be smart, but might be stupid.

I was contacted by http://www.iqt.org/ – they're a US-govt-funded 'strategic investment' company, and part of what they do is holding an annual conference on emerging technologies for US intelligence agencies. This year the theme is "Mobility of Money".

They asked if I'd be willing to talk about Bitcoin, and I committed to giving a 50-minute presentation and participating in a panel discussion.

I hope that by talking directly to "them" and, more importantly, listening to their questions/concerns, they will think of Bitcoin the way I do– as a just-plain-better, more efficient, less-subject-to-political-whims money. Not as an all-powerful black-market tool that will be used by anarchists to overthrow The System.

It might be really stupid if it just raises Bitcoin's visibility on their radar, but I think it is way too late for that; Bitcoin is already on their radar.

I plan on posting about this on the forums soon, because "Gavin secretly visits the CIA" would spin all sorts of conspiracy theories. "Gavin openly visits the CIA" will create enough conspiracy theories as it is.

71

Bitcoin and Me (Hal Finney)

SINCE HE WAS THE RECIPIENT of the very first Bitcoin transaction and someone who was involved early, it is worth adding this wonderful post from Hal Finney on the *bitcointalk.org* forum dated March 19, 2013.

Bitcoin and me (Hal Finney)

Hal Finney, March 19, 2013, 08:40:02PM

I thought I'd write about the last four years, an eventful time for Bitcoin and me.

For those who don't know me, I'm Hal Finney. I got my

start in crypto working on an early version of PGP, working closely with Phil Zimmermann. When Phil decided to start PGP Corporation, I was one of the first hires. I would work on PGP until my retirement. At the same time, I got involved with the Cypherpunks. I ran the first cryptographically based anonymous remailer, among other activities.

Fast forward to late 2008 and the announcement of Bitcoin. I've noticed that cryptographic graybeards (I was in my mid 50's) tend to get cynical. I was more idealistic; I have always loved crypto, the mystery and the paradox of it.

When Satoshi announced Bitcoin on the cryptography mailing list, he got a skeptical reception at best. Cryptographers have seen too many grand schemes by clueless noobs. They tend to have a knee jerk reaction.

I was more positive. I had long been interested in cryptographic payment schemes. Plus I was lucky enough to meet and extensively correspond with both Wei Dai and Nick Szabo, generally acknowledged to have created ideas that would be realized with Bitcoin. I had made an attempt to create my own proof of work based currency, called RPOW. So I found Bitcoin facinating.

When Satoshi announced the first release of the software, I grabbed it right away. I think I was the first person besides Satoshi to run bitcoin. I mined block 70-something, and I was the recipient of the first bitcoin transaction, when Satoshi sent ten coins to me as a test. I carried on an email conversation with Satoshi over the next few days, mostly me reporting bugs and him fixing them.

Today, Satoshi's true identity has become a mystery. But at the time, I thought I was dealing with a young man of Japanese ancestry who was very smart and sincere. I've had the good fortune to know many brilliant people over the course of my life, so I recognize the signs.

After a few days, bitcoin was running pretty stably, so I left it running. Those were the days when difficulty was 1, and you could find blocks with a CPU, not even a GPU. I mined several blocks over the next days. But I turned it off because it made my computer run hot, and the fan noise bothered me. In retrospect, I wish I had kept it up longer, but on the other hand I was extraordinarily lucky to be there at the beginning. It's one of those glass half full half empty things.

The next I heard of Bitcoin was late 2010, when I was surprised to find that it was not only still going, bitcoins actually had monetary value. I dusted off my old wallet, and was relieved to discover that my bitcoins were still there. As the price climbed up to real money, I transferred the coins into an offline wallet, where hopefully they'll be worth something to my heirs.

Speaking of heirs, I got a surprise in 2009, when I was suddenly diagnosed with a fatal disease. I was in the best shape of my life at the start of that year, I'd lost a lot of weight and taken up distance running. I'd run several half marathons, and I was starting to train for a full marathon. I worked my way up to 20+ mile runs, and I thought I was all set. That's when everything went wrong.

My body began to fail. I slurred my speech, lost strength in my hands, and my legs were slow to recover. In August, 2009, I was given the diagnosis of ALS, also called Lou Gehrig's disease, after the famous baseball player who got it.

ALS is a disease that kills moter neurons, which carry signals from the brain to the muscles. It causes first weakness, then gradually increasing paralysis. It is usually fatal in 2 to 5 years. My symptoms were mild at first and I continued to work, but fatigue and voice problems forced me to retire in early 2011. Since then the disease has continued its inexorable progression.

Today, I am essentially paralyzed. I am fed through a tube, and my breathing is assisted through another tube. I operate the computer using a commercial eyetracker system. It also has a speech synthesizer, so this is my voice now. I spend all day in my power wheelchair. I worked up an interface using an arduino so that I can adjust my wheelchair's position using my eyes.

It has been an adjustment, but my life is not too bad. I can still read, listen to music, and watch TV and movies. I recently discovered that I can even write code. It's very slow, probably 50 times slower than I was before. But I still love programming and it gives me goals. Currently I'm working on something Mike Hearn suggested, using the security features of modern processors, designed to support "Trusted Computing", to harden Bitcoin wallets. It's almost ready to release. I just have to do the documentation.

And of course the price gyrations of bitcoins are entertaining to me. I have skin in the game. But I came by my bitcoins through luck, with little credit to me. I lived through the crash of 2011. So I've seen it before. Easy come, easy go.

That's my story. I'm pretty lucky overall. Even with the ALS, my life is very satisfying. But my life expectancy is limited. Those discussions about inheriting your bitcoins are of more than academic interest. My bitcoins are stored in our safe deposit box, and my son and daughter are tech savvy. I think they're safe enough. I'm comfortable with my legacy.

Hal Finney

72

Conclusion

SATOSHI NAKAMOTO brought together many existing mathematical and software concepts to create Bitcoin. Since then, Bitcoin has been an ongoing experiment, continuing to evolve and be updated on a regular basis. It has, so far, proven its utility and revolutionized the financial and monetary industry, particularly the electronic payment system, and is being accepted worldwide. Bitcoin, in itself, may or may not survive up to the year 2140 when all bitcoins will have been mined, but the idea of a distributed, peer-to-peer and decentralized limited-supply currency is here to stay.

The ability to transfer money digitally has been available only recently in human history but this is merely a mechanistic change in the handling of money, a new way to perform the same action. But gold and silver or any other non-inflatable entity cannot be directly transmitted electronically and thus require a conceptual delegate that might misrepresent their quantity if too many copies were to be made

(i.e., this delegate were to be inflated). The greater the quantity of any currency, the less valuable it becomes, and the less, in real goods and services, it is capable of purchasing.

Then, in late 2009, Satoshi introduced Bitcoin. The concept of a decentralized digital currency—open source and with an open accounting book—has been actualized. Interestingly, in contrast to gold and silver, which can only exist in the physical world, bitcoins can only exist in the electronic world.[3] Because of that, in essence one can argue that precious metals and Bitcoin complement each other very nicely.

The fact that Bitcoin is an open source software whose transactions must be confirmed by all members of the network and which operates with a public ledger makes it the polar opposite of a centrally controlled, closed currency system. Regardless of whether regulators are involved in a closed system or not, such systems are just as susceptible to corruptions and bribes to government leaders as any other government-controlled institution. As rare, precious metals, gold and silver are an excellent choice to use as money, but their inability to be transferred electronically requires some sort of intermediary, representative form, capable of being manipulated by a third party. Transporting a large sum of gold and silver is also cumbersome and expensive. Precious metals will, however, hold their value in major disruptions such as electric grid blackouts and would certainly become the currency of choice in a Mad Max scenario. For those fearful of such an eventuality, possessing a certain amount of gold and silver is appropriate. In any case, all fiat (i.e., government-decreed) currencies throughout history have always died, and you should not expect your country's currency to prove an exception to this rule.

---

[3] So far, conveying bitcoins in the physical world implies some form of artifact like a paper wallet with the Bitcoin address and private key inscribed on it. Or alternatively, it requires some trust in a third party, for example the manufacturer of a physical coin with a hidden Bitcoin private key along with a visible Bitcoin address.

This book has presented all the most relevant conversations and discussions in which the creator of Bitcoin was involved. Whether a group or an individual, the person known as Satoshi Nakamoto expressed himself clearly and concisely and naturally understood very well Bitcoin's foundation. His various writings seem to indicate that he did not expect Bitcoin to take off as rapidly as it has. Satoshi assembled various existing concepts to create this formidable technology currently revolutionizing how a monetary system is conceptualized. He has opened a Pandora's Box, and many brilliant minds are working beyond Bitcoin to revolutionize other systems based on its precepts.

Whether Bitcoin represents money is subject to debate, but that it is a currency, a medium of exchange, is indisputable. Gold and silver are a store of value over the long term because of their limited supply and their usefulness. Bitcoin also has a limited supply—the 21 million bitcoins planned as of 2140—and has proven so far to be very useful as an easy form of payment over the Internet, its natural medium.

Satoshi covered many of the arguments we have seen debated over and over in the news media since Bitcoin's rise in popularity. Although we would have loved to hear him discuss them in person, this book gives us the ability to easily revisit the many opinions he shared during his "public life". Bitcoin's major impact has been to allow the population of the world to reconsider how a currency should function. It opens the door for humanity to a new monetary system, an electronic renaissance.

Thank you!

"Bitcoin: A Peer-to-Peer Electronic Cash System" by Satoshi Nakamoto

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

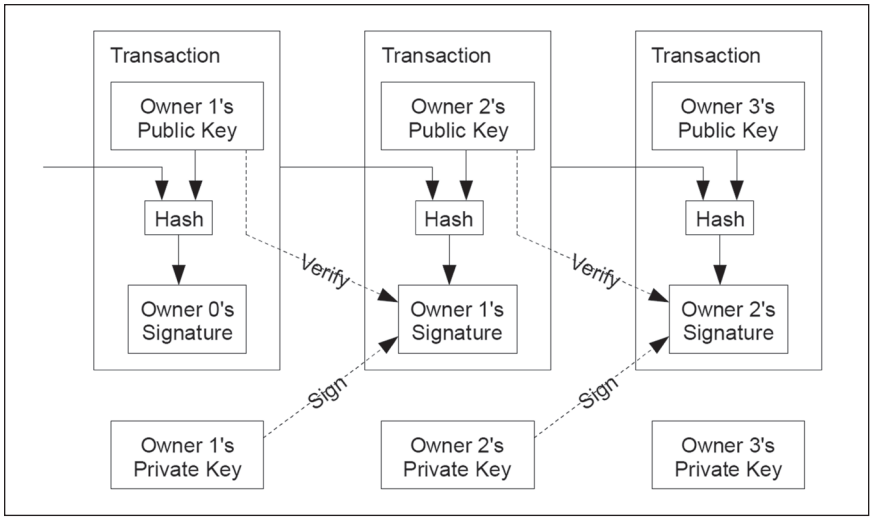
Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible,

since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

2. Transactions

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.



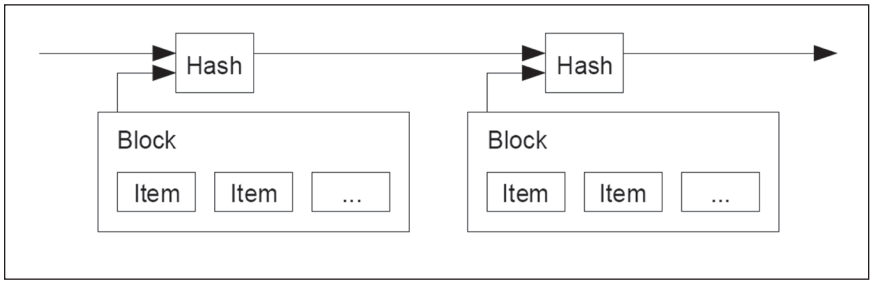
The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee

needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

3. Timestamp Server

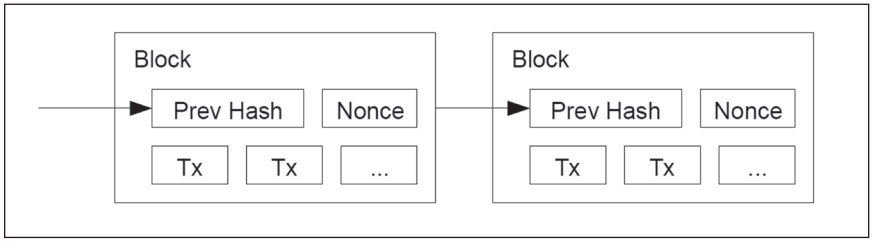
The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

5. Network

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

6. Incentive

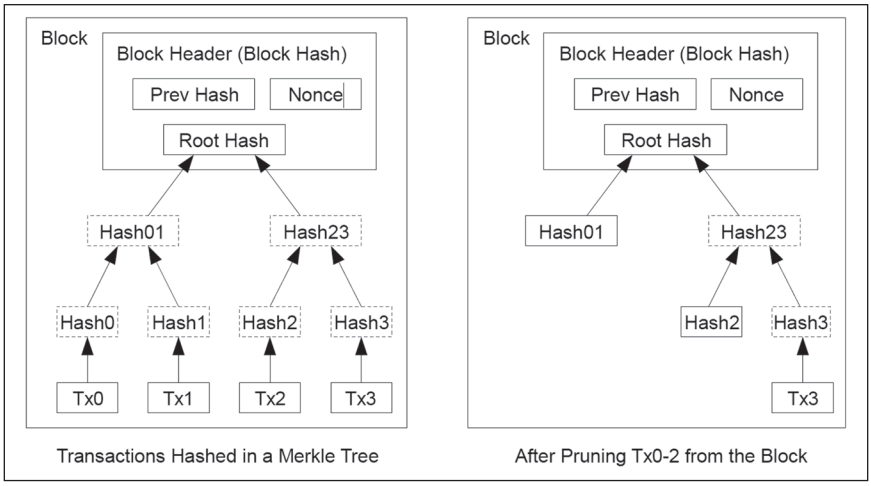
By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

7. Reclaiming Disk Space

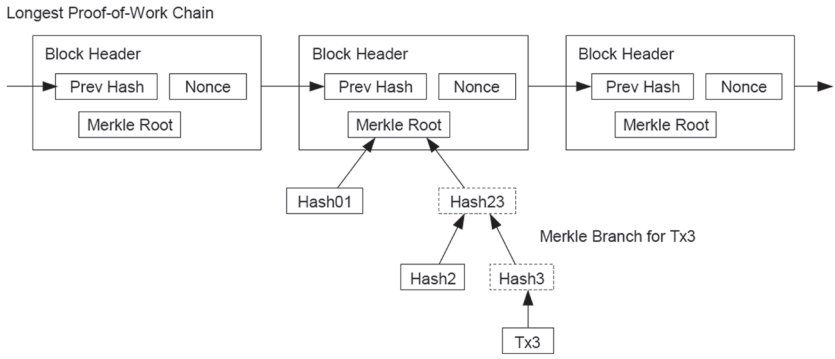
Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash, transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.



A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory.

8. Simplified Payment Verification

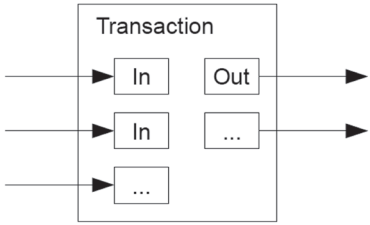
It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

9. Combining and Splitting Value

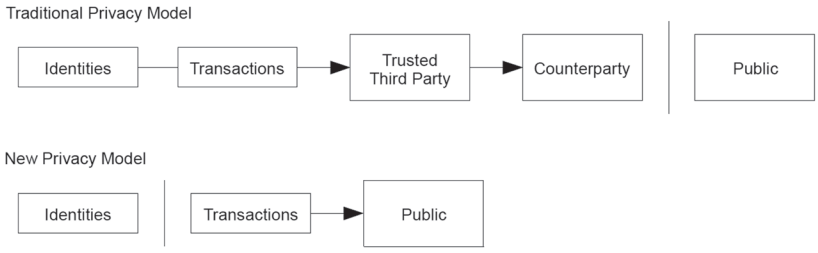
Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

10. Privacy

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

11. Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Given our assumption that p > q, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution…

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Converting to C code…

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
     double p = 1.0 - q;
     double lambda = z * (q / p);
     double sum = 1.0;
     int i, k;
     for (k = 0; k <= z; k++)
     {
       double poisson = exp(-lambda);
       for (i = 1; i <= k; i++)
         poisson *= lambda / i;
       sum -= poisson * (1 - pow(q / p, z - k));
}
return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1
z=0 P=1.0000000
z=1 P=0.2045873
z=2 P=0.0509779
z=3 P=0.0131722
z=4 P=0.0034552
z=5 P=0.0009137
```

```
z=6 P=0.0002428
z=7 P=0.0000647
z=8 P=0.0000173
z=9 P=0.0000046
z=10 P=0.0000012
q=0.3
z=0 P=1.0000000
z=5 P=0.1773523
z=10 P=0.0416605
z=15 P=0.0101008
z=20 P=0.0024804
z=25 P=0.0006132
z=30 P=0.0001522
z=35 P=0.0000379
z=40 P=0.0000095
z=45 P=0.0000024
z=50 P=0.0000006
```

Solving for P less than 0.1%...

```
P < 0.001
q=0.10 z=5
q=0.15 z=8
q=0.20 z=11
q=0.25 z=15
q=0.30 z=24
q=0.35 z=41
q=0.40 z=89
q=0.45 z=340
```

12. Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a

public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

References

1. W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.
2. H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.
3. S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99–111, 1991.
4. D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.
5. S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28–35, April 1997.
6. A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

7. R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122–133, April 1980.
8. W. Feller, "An introduction to probability theory and its applications," 1957.

Terms & Definition

Asymmetric Encryption—A type of encryption involving two keys, a private key and a public key. Text that is encrypted with the private key must be decrypted with the public key and vice versa. The public key is easily derived from the private key but the reverse is nearly impossible.

Bitcoin Address—A long series of digits to which the block chain will associate bitcoins. It is the hash output of the public key. Any bitcoins it contains can only be transferred to another Bitcoin address by the person who owns its corresponding private key.

Block—A chunk of data that contains several Bitcoin transactions that miners work on creating.

Block Chain—The Bitcoin block chain is shared via a peer-to-peer network between all miners and interested nodes (computers). It contains all the blocks since the creation of Bitcoin on January 3rd, 2009.

BTC—The acronym representing Bitcoin currency.

Cryptography—The study of techniques by which communications are secured.

Cryptographic Hash—An algorithm that creates a fixed-length series of numbers from an input having an arbitrary length. The algorithm's output can be defined as the equivalent of the "fingerprint" of a document.

Distributed File Sharing—A system in which files are shared across multiple computers which act as both consumers and providers of information.

Encryption—The process of encoding messages or information in such a way that only authorized parties can read or access them.

Elliptic Curve Cryptography—A public-key cryptography based on the algebraic structure of elliptic curves over a finite number of elements (finite fields). Elliptic curves are also used in several integer factorization algorithms that have applications in cryptography.

Elliptic Curve Digital Signature Algorithm (ECDSA) - In cryptography, the Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA), which employs elliptic curve cryptography.

Genesis Block—The very first block of the block chain.

Hash, Hash Function—Hash is the fixed-length output of a cryptographic algorithm, or hash function. Hash is a document's "fingerprint", where the document, which may be of any length, is the text being encoded by the hash function.

Hexadecimal Number System—Whereas the decimal numbering system is based on 10, the hexadecimal system is based is 16, and the system employs the symbols 0 through 9 to represent the numbers 0 through 9 and the symbols A, B, C, D, E, and F (either in lower case or upper case) to represent the numbers 10 through 15. Hexadecimal numbers are prefixed by 0x, and so decimal 16 is 0x10 in hexadecimal, decimal 17 is 0x11, and so on.

Ledger—In accounting, this is the principal book or computer file for recording and totaling monetary transactions by account. It includes a beginning balance, debits, credits, and an ending balance.

Message Digest—The output of a cryptographic hash function.

Miners—Initially called nodes, these are devices with specialized hardware that compete in creating the next block and so collecting the rewards associated with it. Rewards are composed of new

bitcoins the protocol allows miners to create along with the sum of all transaction fees contained in that block.

Nonce—A number within a block that a miner increments until he or another miner obtains the message digest with the characteristics required by the Bitcoin protocol to constitute "winning" that block.

Open Source Software—Software code (blue print) that is shared and available for anyone to see, inspect, and modify so as to be able to reproduce the programs themselves.

Peer-to-Peer network—A decentralized and distributed network architecture where individual nodes (computers) in the network act as both suppliers and consumers of resources. This is in contrast with a centralized client-server model where clients request resources from the server.

Proof-of-Work—This is like a "contest" in which each miner competes. The first miner to obtain a "nonce" that generates the message digest with the characteristics defined by the Bitcoin protocol as a "win".

Protocol—An established procedure that miners and clients must follow. It is dictated by the Bitcoin open source software which all miners run.

SHA256—One type of cryptographic hash algorithm. It is currently used by the Bitcoin software.

Satoshi—The smallest denomination of Bitcoin. It is equivalent to 10^{-8} bitcoin, and so there are 100,000,000 satoshis in 1 bitcoin.

Transaction fee—This is the cost that senders of bitcoins pay miners to include their transactions in the next block chain.

Wallet—The software that contains a list of Bitcoin addresses and their corresponding private keys.

Index