

Blockchain Lecture 2

Heung-No Lee

March 5th 2018

Class Schedule

Monday	Wednesday	
	4/11	
16	4/18 조원선 IncuBlock 대표 특강	
23 김형중 교수 특강	4/25 HW#2 (Midterm) due	
4/30	5/2 Midterm	
7	5/9	
14	16 UPBit 김형년 대표 특강	
21	23 HW#3 due	
28	30 박창기 GovernTech 대표 특강 TBD	
4	6 HW#4 due	
	13 Final Exam	

Bitcoin Summary

- Bitcoin is an electronic cash.
- This e-cash can be used to transfer the ownership chain of signatures.
- A fixed amount of bitcoins are created in each block.
- The created bitcoins for a block are given to the miner who has succeeded in finding the nonce value for the pertinent block.
- In fact, the miner has the right to produce a fixed amount of bitcoin and give it to one of his bitcoin address.
- Thus, each created bitcoin belongs to a bitcoin address.
- Ownership rights are transferred from payer bitcoin address to the payee bitcoin address.
- A transaction is valid only when it is attached with a digital sign.
- A valid digital sign shows the proof of ownership of the pertinent coin.
- Miners are the ones who verify the digital signs and make sure to see if the pertinent coin is not already spent.
- Miners put valid transactions to a block and find a good hash for the block.
- The miner who found a valid block summary is given the right to generate the bitcoin.
- Bitcoin is an electronic cash system which runs without the third party such as mint or bank.
- In Bitcoin, however, the third party is the network of miners who verify the validity of each transaction and scribing validated transactions into the blockchain.
- The miners are decentralized and autonomous. Anyone can join as a miner. They simply need to buy mining chips, connect to the open Bitcoin network, and become a miner (person). Anytime these miners can stop working as a miner any time.

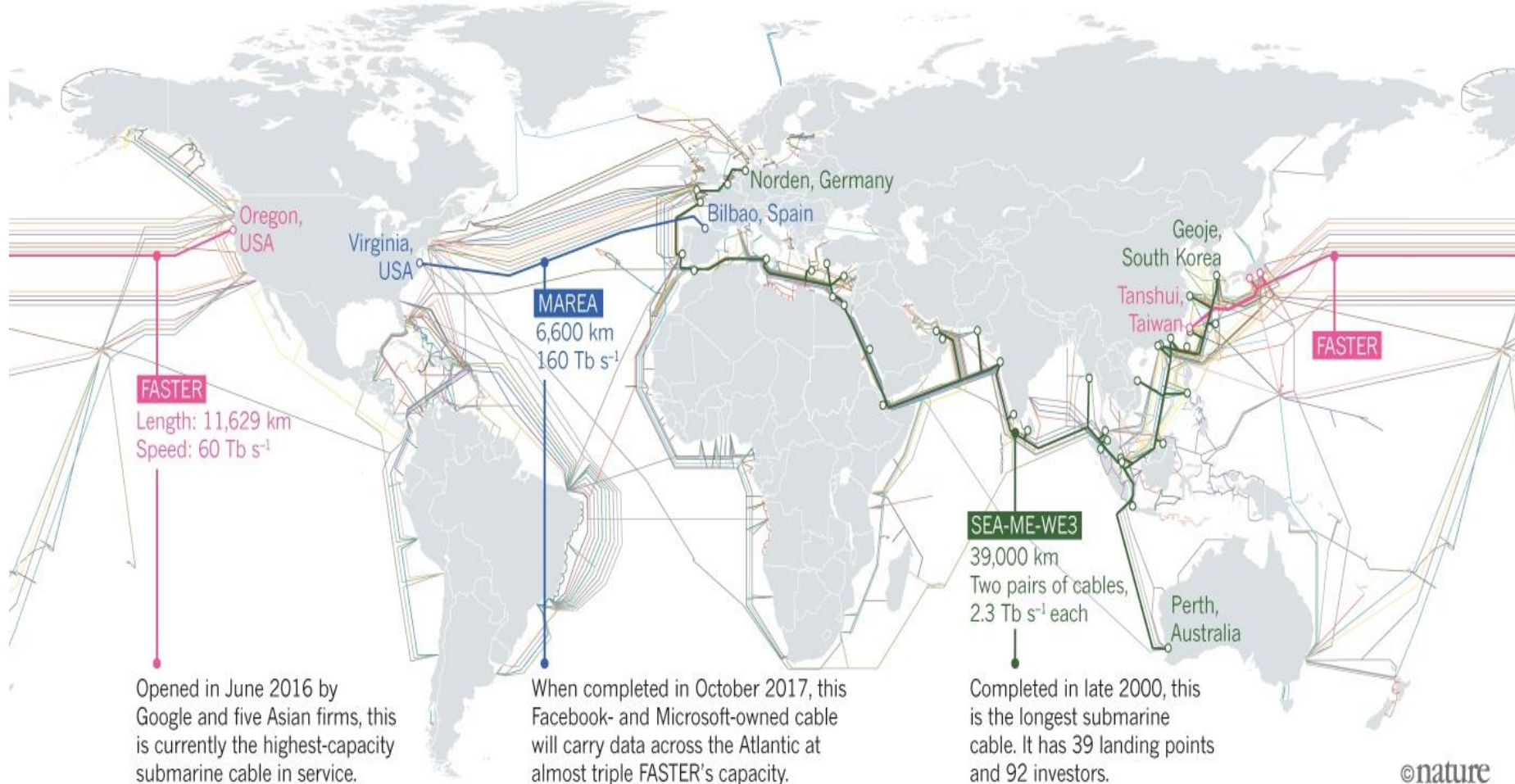
Fork occurs, why?

- Block validations are worked out by individual miners in a distributed way.
- Each miner independently works on a different block validation.
- There are internet delays. Thus, announcement of a mined block announcement may not reach to other miners in time.
- It might be under a double spending race attack. But this one is hidden until the attacker launches his attack by announcing it to the public.

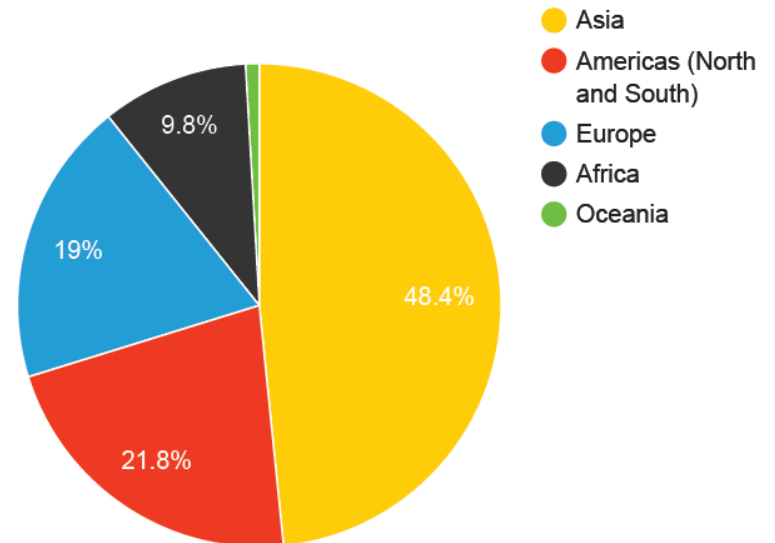
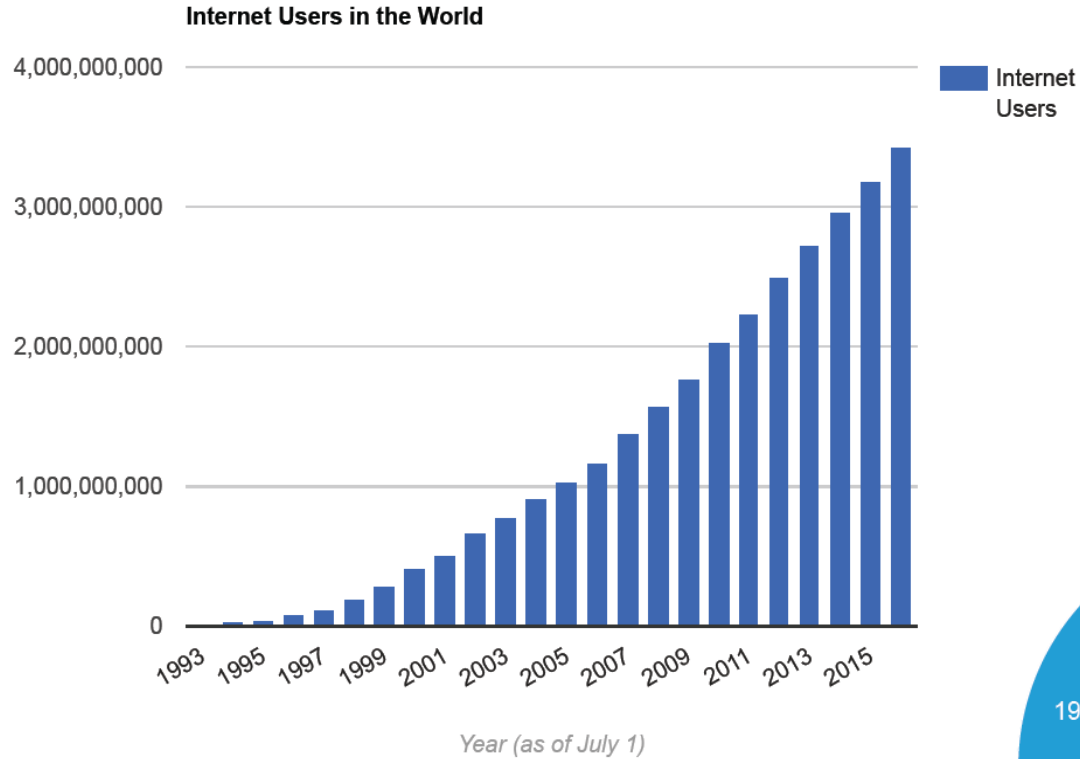
Internet Today

THE SUBMARINE WEB

Much of the world's Internet traffic passes under the oceans, through fibre-optic cables that can run along the sea bed for thousands of kilometres. Companies are constantly laying more and better cables.



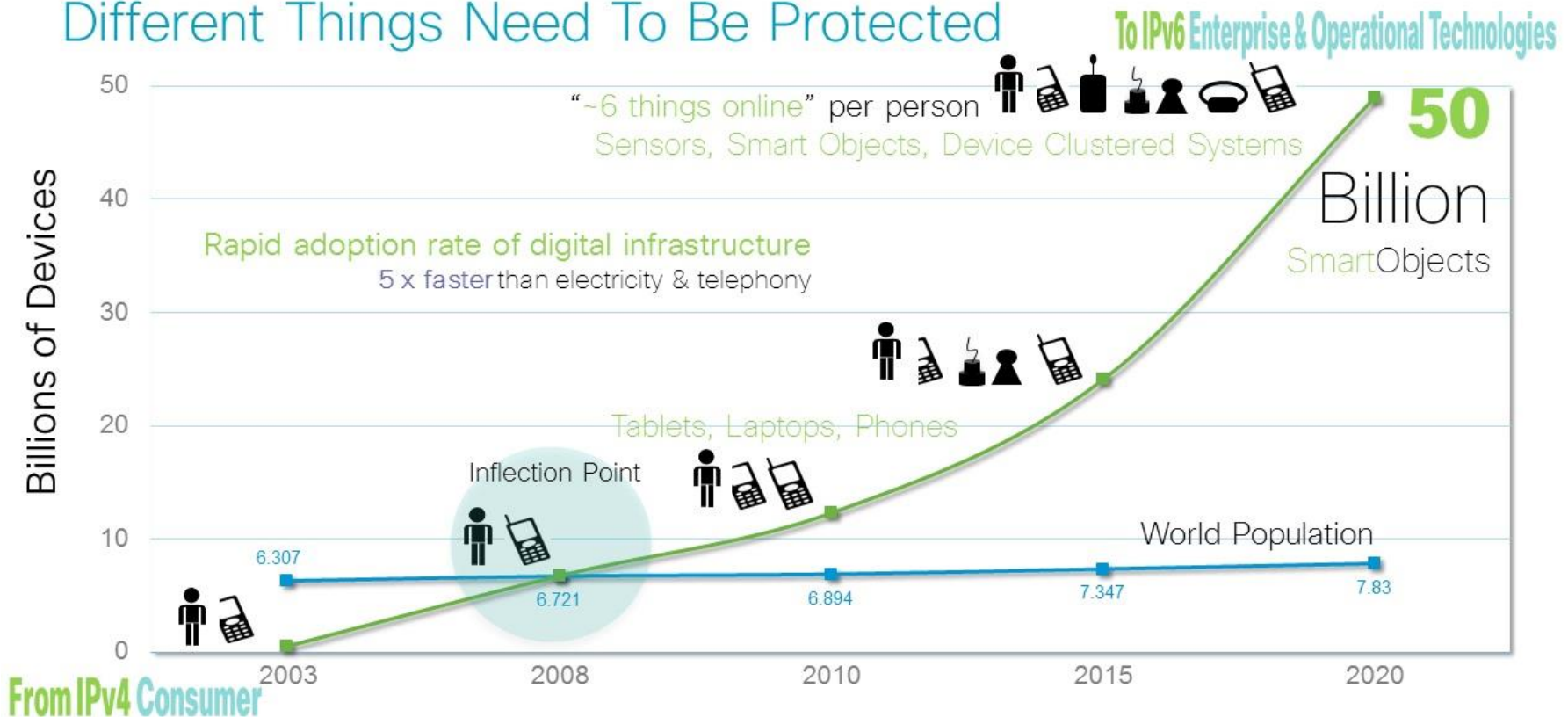
Internet Users (3.4B/7.4B, 46%)



Taken from google images

Internet of Things Devices

Different Things Need To Be Protected



Source: Cisco IBSG projections, UN Economic & Social Affairs <http://www.un.org/esa/population/publications/longrange2/WorldPop2300final.pdf>

Taken from google images

How to resolve a fork?

- In Bitcoin, the longest chain wins.
- Longest chain has largest amount of proof-of-work done.
- More work is honored.
- Chain with more work done is safer from attacks.
- Thus, the miners work on extending the longer chain known to them.

What happens to TXs in the Deserted Chain?

- As miners do not work on the shorter chain, the shorter chains are deserted.
- Then, question arises
- 1) what happens to those transactions included in the forked branches of the deserted chain?
 - They have to go back to the pool of transactions.
- 2) What happens to the reward given to the miners in the deserted forked branches?
 - Each mining reward in a mined block is locked for 100 blocks. The coinbase transaction at the top of a particular block with a block height h cannot be spent until the block height of the chain reaches $h+100$.

Non-forkable blockchains

2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks

The Balance Attack or Why Forkable Blockchains Are Ill-Suited for Consortium

Christopher Natoli
School of IT
University of Sydney
Sydney, Australia
christopher.natoli@sydney.edu.au

Vincent Gramoli
School of IT
Data61-CSIRO and University of Sydney
Sydney, Australia
vincent.gramoli@sydney.edu.au

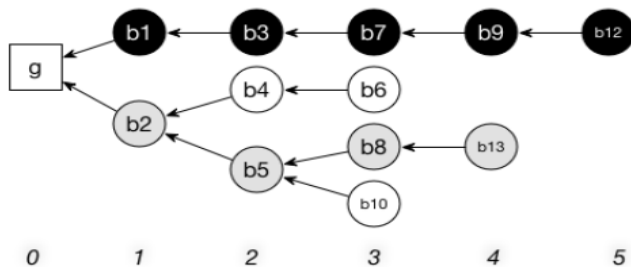


Figure 2: Nakamoto's consensus protocol at the heart of Bitcoin selects the main branch as the longest branch (in black) whereas the GHOST consensus protocol at the heart of Ethereum follows the heaviest subtree (in grey)

Abstract—Most blockchain systems are *forkable* in that they require participants to agree on a chain out of multiple possible branches of blocks. In this paper, we identify a new form of attack, called the Balance attack, against these forkable blockchain systems. The novelty of this attack consists of delaying network communications between multiple subgroups of nodes with balanced mining power. Our theoretical analysis captures the tradeoff between the network delay and the mining power of the attacker needed to double-spend in the GHOST protocol with high probability.

We quantify our analysis in the settings of the Ethereum testnet of the R3 consortium where we show that a single machine needs to delay messages for 20 minutes to double spend while a coalition with a third of the mining power would simply need 4 minutes to double spend with 94% of success. We experiment the attack in our private Ethereum chain before arguing for a non-forkable blockchain design to protect against Balance attacks.

Bitcoin vs. Ethereum Consensus Protocols

Nakamoto's consensus algorithm: The difficulty of the crypto-puzzles used in Bitcoin produces a block every 10 minutes in expectation. The advantage of this long period, is that it is relatively rare for the blockchain to fork because blocks are rarely mined during the time others are propagated to the rest of the nodes.

Algorithm 2 Nakamoto's consensus protocol at node p_i

```
6:  $m = 5$ , the number of blocks to be appended after the one containing
7:  $tx$ , for  $tx$  to be committed in Bitcoin

8:  $\text{get-main-branch}()_i$ : ▷ select the longest branch
9:  $b \leftarrow \text{genesis-block}(B_i)$  ▷ start from the blockchain root
10: while  $b.\text{next} \neq \perp$  do ▷ prune shortest branches
11:    $\text{block} \leftarrow \text{argmax}_{c \in \text{children}(b)} \{\text{depth}(c)\}$  ▷ deepest subtree
12:    $B \leftarrow B \cup \{\text{block}\}$  ▷ update vertices of main branch
13:    $P \leftarrow P \cup \{\langle \text{block}, b \rangle\}$  ▷ update edges of main branch
14:    $b \leftarrow \text{block}$  ▷ move to next block
15: return  $\langle B, P \rangle$  ▷ returning the Bitcoin main branch

16:  $\text{depth}(b)_i$ : ▷ depth of tree rooted in  $b$ 
17: if  $\text{children}(b) = \emptyset$  then return 1 ▷ stop at leaves
18: else return  $1 + \max_{c \in \text{children}(b)} \text{depth}(c)$  ▷ recurse at children
```

6) *The GHOST consensus algorithm:* As opposed to the Bitcoin protocol, Ethereum generates one block every 12–15 seconds. While it improves the throughput (transactions per second) it also favors transient forks as miners are more likely to propose new blocks without having heard about the latest mined blocks yet. To avoid wasting large mining efforts while resolving forks, Ethereum uses the GHOST (Greedy Heaviest Observed Subtree) consensus algorithm that accounts for the, so called *uncles*, blocks of discarded branches. In contrast with Nakamoto's protocol, the GHOST protocol iteratively *selects*, as the successor block, the root of the subtree that contains the largest number of nodes (cf. Algorithm 3).

Algorithm 3 The GHOST consensus protocol at node p_i

```
6:  $m = 11$ , the number of blocks to be appended after the one containing
7:  $tx$ , for  $tx$  to be committed in Ethereum (since Homestead v1.3.5)

8:  $\text{get-main-branch}()_i$ : ▷ select the branch with the most nodes
9:  $b \leftarrow \text{genesis-block}(B_i)$  ▷ start from the blockchain root
10: while  $b.\text{next} \neq \perp$  do ▷ prune lightest branches
11:    $\text{block} \leftarrow \text{argmax}_{c \in \text{children}(b)} \{\text{num-desc}(c)\}$  ▷ heaviest tree
12:    $B \leftarrow B \cup \{\text{block}\}$  ▷ update vertices of main branch
13:    $P \leftarrow P \cup \{\langle \text{block}, b \rangle\}$  ▷ update edges of main branch
14:    $b \leftarrow \text{block}$  ▷ move to next block
15: return  $\langle B, P \rangle$ . ▷ returning the Ethereum main branch

16:  $\text{num-desc}(b)_i$ : ▷ number of nodes in tree rooted in  $b$ 
17: if  $\text{children}(b) = \emptyset$  then return 1 ▷ stop at leaves
18: else return  $1 + \sum_{c \in \text{children}(b)} \text{num-desc}(c)$  ▷ recurse at children
```

Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies

Florian Tschorsch and Björn Scheuermann

Abstract—Besides attracting a billion dollar economy, Bitcoin revolutionized the field of digital currencies and influenced many adjacent areas. This also induced significant scientific interest. In this survey, we unroll and structure the manifold results and research directions. We start by introducing the Bitcoin protocol and its building blocks. From there we continue to explore the design space by discussing existing contributions and results. In the process, we deduce the fundamental structures and insights at the core of the Bitcoin protocol and its applications. As we show and discuss, many key ideas are likewise applicable in various other fields, so that their impact reaches far beyond Bitcoin itself.

Index Terms—Altcoins, Bitcoin, blockchain, cryptocurrencies, digital currencies, distributed consensus, survey, tutorial.

attempt. Accomplishing the same in a distributed setting is far from trivial. The distribution of information and the problem of mutual agreement on a consistent state is a challenge, especially in the presence of selfish and/or malicious participants. It boils down to the Byzantine Generals problem [11]. This insight [12] pushed the idea to employ quorum systems [13]. Quorum systems, as described in [14], accept the possibility of faulty information and the existence of malicious entities in a distributed environment. They introduce the concept of voting. As long as the majority of any chosen subset of peers (quorum) is honest, the correct ledger state can be obtained by election. However, the approach is vulnerable to the Sybil attack

- This paper is 39 page long with 245 references.
- It is a good idea to read this paper to understand the status of Bitcoin network.
- It is difficult to follow user group posts and program structure of bitcoin source code.
- This is much easier to follow.
- In this lecture, I will follow the materials discussed in this paper.
- Given that this paper was published in 2016, some materials could have become outdated.

Gambler's Ruin Problem

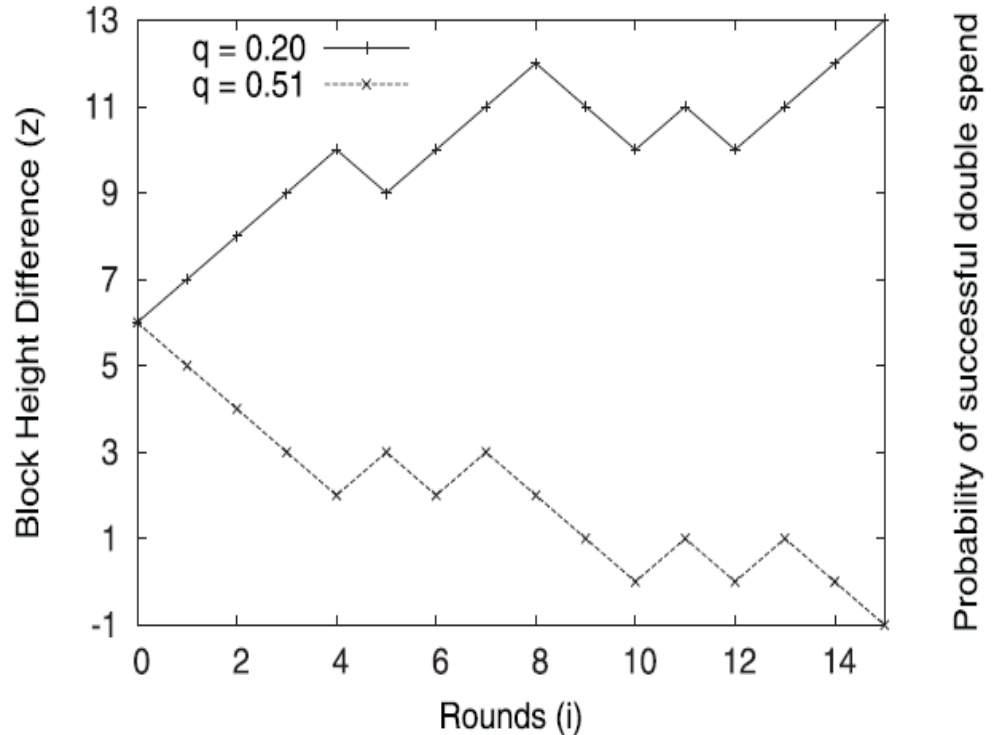
In order to assess the success conditions for this type of double spend attacks, let us take a look how to model the race between the benign and the fraudulent block chain. As in [16], [58], we describe it as a binomial random walk. Assume that the hash rates and therefore the difficulty remain constant. Further assume the probability that an honest node finds the next block is p and the probability that an attacker finds the next block is $q = 1 - p$. We denote the difference in heights between the fraudulent and the benign block chain by z . Whenever a block is found, z changes by either $+1$ for a benign block or by -1 for a fraudulent block. Thus z is given by

$$z_{i+1} = \begin{cases} z_i + 1 & \text{with probability } p \\ z_i - 1 & \text{with probability } q. \end{cases}$$

We are interested in the question whether z will ever become -1 , which implies that the attacker surpassed the benign block chain and successfully mounted a double spend. Analogously to the results presented in [58], Figure 5a exemplarily shows the two possible outcomes in a random walk simulation. We

in a single iteration, it can be shown that the probability q_z of experiencing gambler's ruin having started with z credits yields

$$q_z = \begin{cases} 1 & \text{if } z < 0 \text{ or } q > p \\ (q/p)^z & \text{if } z \geq 0 \text{ and } q \leq p. \end{cases}$$



(a) Random walk simulation ($z_0 = 6$).

Fig. 5. Hash rate-based double spending analysis based on the results from [16], [58].

Still, the same on this part!

Race Attack Success Probability

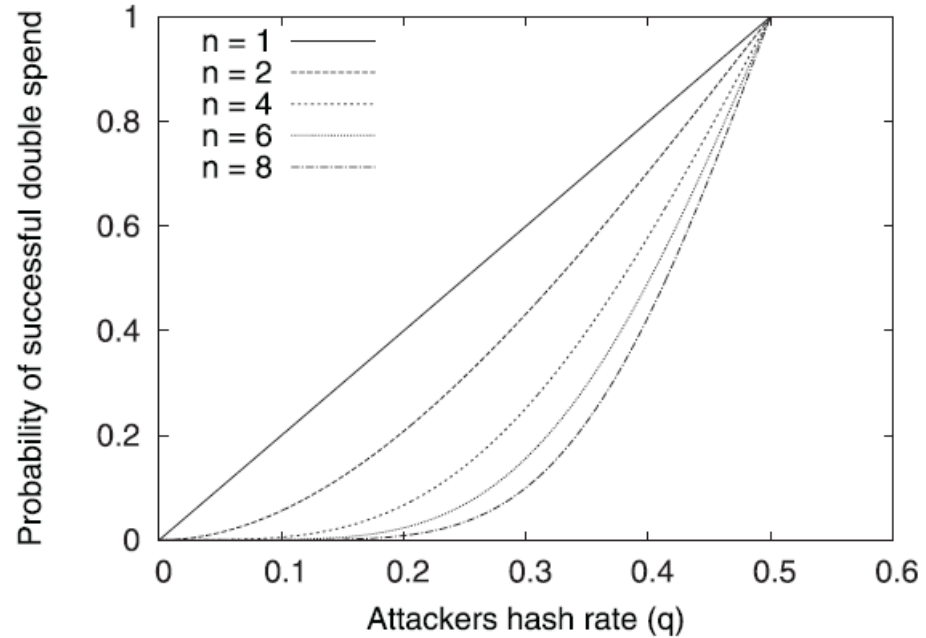
A little bit of improve on Poisson

the number of which we denote by m . The original Bitcoin paper [16] assumes that m follows a Poisson distribution. More accurately, [58] models the probability of m as a negative binomial variable $P(m)$. Furthermore it is there assumed that the attacker pre-mined a block before initiating the attack, hence $z = n - m - 1$. It follows that the probability of a successful double spend equals

$$r = \sum_{m=0}^{\infty} P(m) \cdot q_z$$

$$= \begin{cases} 1 & \text{if } q \geq p \\ 1 - \sum_{m=0}^n \binom{m+n-1}{m} (p^n q^m - p^m q^n) & \text{if } q < p. \end{cases} \quad (1)$$

We visualized the results of equation (1) for various numbers of confirmations n in Figure 5b. Clearly, the higher the num-








(b) Probability of successful double spending.

[58] M. Rosenfeld, "Analysis of hashrate-based double spending," Tech. Rep., 2012 [Online]. Available: <https://bitcoil.co.il/Doublespend.pdf>

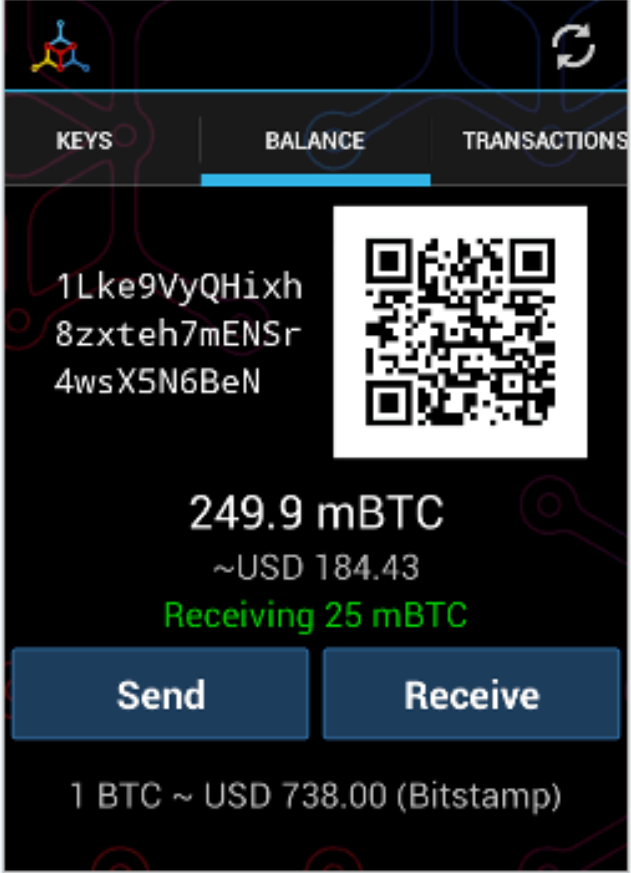
Mycelium Wallet

Mycelium

[Install](#) [Source code](#)

-  **Control over your money** ?
-  **Centralized validation** ?
-  **Basic transparency** ?
-  **Secure environment** ?
-  **Basic privacy** ?

Mycelium Bitcoin Wallet for Android is designed for security, speed, and ease of use. It has unique features to manage your keys and for cold storage and offers compatibility with Trezor and others.



KEYS | **BALANCE** | TRANSACTIONS

1Lke9VyQHixh
8zxteh7mENSr
4wsX5N6BeN

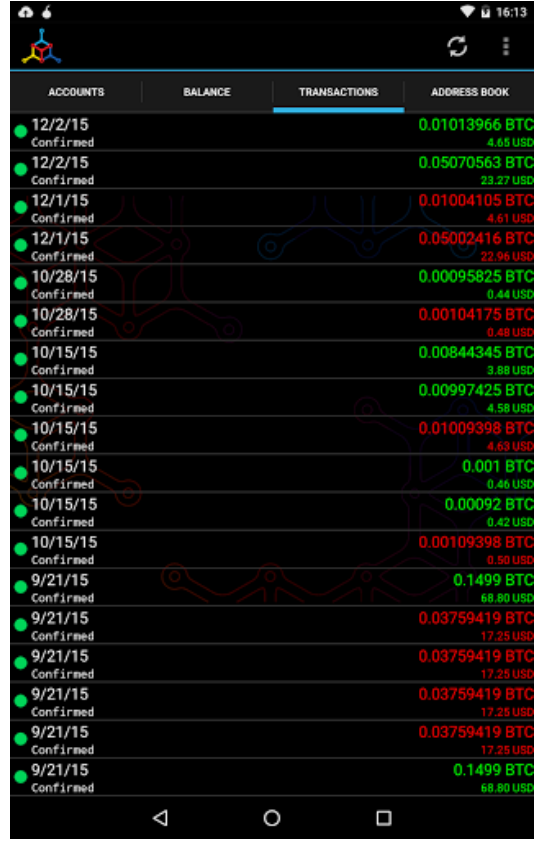
249.9 mBTC
~USD 184.43
Receiving 25 mBTC

Send **Receive**

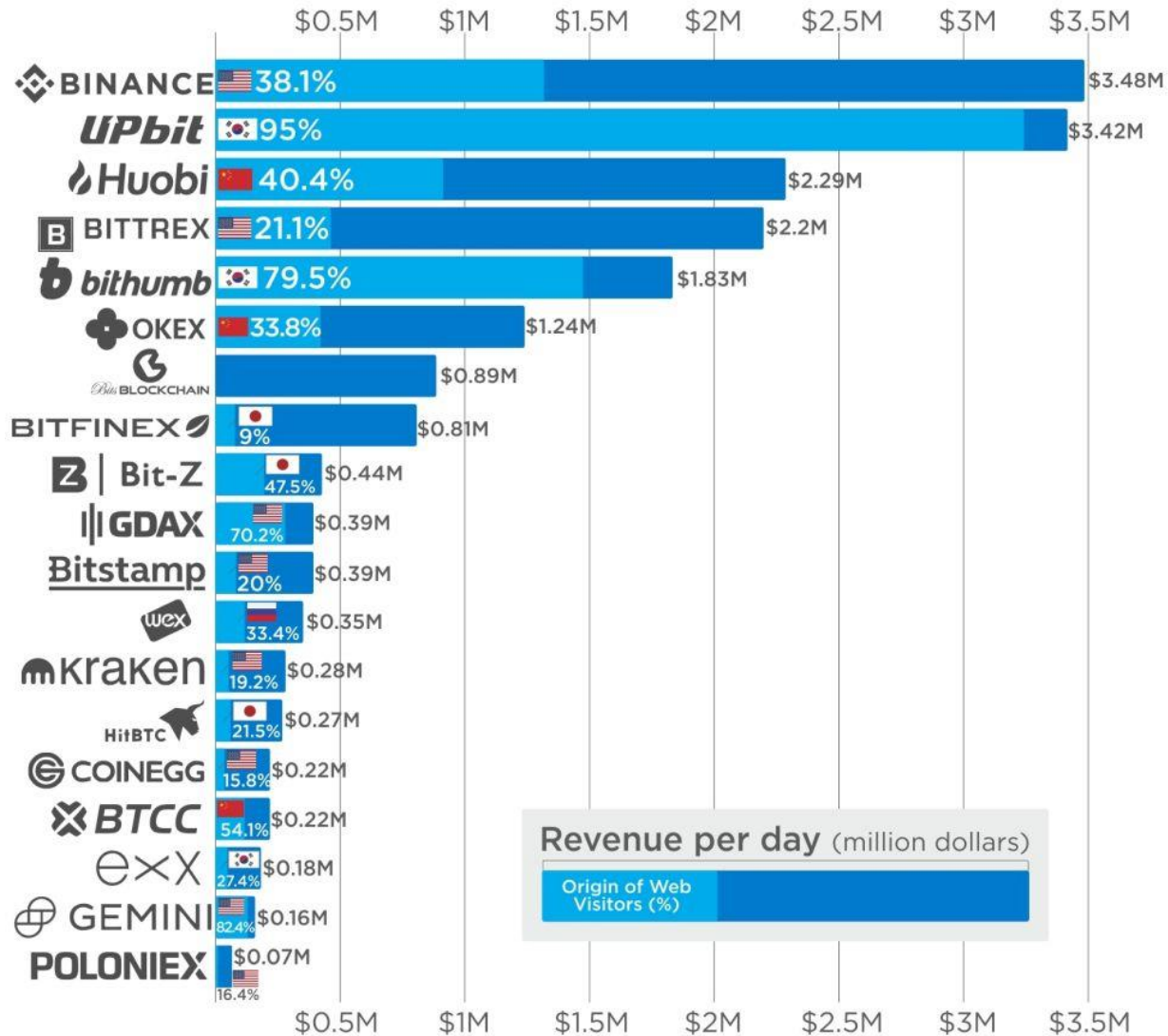
1 BTC ~ USD 738.00 (Bitstamp)

Wallet

- Stores one's private and public key pairs
- Bitcoin addresses are derived from hashing a public key using SHA256 and RIPEMD-160, prepending a version number and appending a checksum for error detection.
- Addresses are base58-encoded to eliminate ambiguous characters.
- The purpose is to make it short and hide the public key.
- There was comparison based attack on signatures.



Crypto Exchanges' Trading Revenue Per Day



* Daily revenue estimated with CoinMarketCap reported 24Hr volume and fees listed on exchanges' websites.
 ** Percent of visitors estimated by Alexa.com. It does not necessarily represents the % of revenue but only the % of web visitors.
Article & Sources:
<https://howmuch.net/articles/crypto-exchanges-revenue>
<https://www.bloomberg.com>
<https://www.alex.com>

Coin Exchange

- What is a coin exchange.
- It provides service for good exchanges just like humanity did in ancient times.
- The assets for exchanges were rice, shells, and fish in ancient times.
- For cryptographic coins exchanges, the assets are the cryptocurrencies.
- Exchanges let users buy and sell coins for fiat money or altcoins.
- Exchanges many hold a significant amount of coins.
- They act as wallet providers.
- Users put a deposit who wish to trade.
- An exchange works because there are sellers and buyers.

Coin Exchange, how it works?

How cryptocurrency exchanges work

There are numerous exchanges and they operate in individual ways but they typically offer these facilities.

ENTERING THE MARKET



Customers wanting to trade cryptocurrencies set up an account where they deposit traditional currency – such as dollars or euros – or some cryptocurrency.

WORKING WITH EXCHANGES



When a customer places an order, the exchange finds a suitable buyer or seller and then either credits or debits the trader's account.



Some use "hot" online wallets, which may be vulnerable to hacking; others use "cold" wallets, which are offline and therefore less vulnerable.



Exchanges record the transactions on their ledgers, and a trader's funds are held by the exchange until the trader withdraws them.

MAKING MONEY



Exchanges charge traders fees (normally a percentage of each transaction) and offer financing services to their clients. Prices vary between exchanges so some traders place orders in various exchanges hoping to profit from the different prices on offer.



Ordinary customers can trade directly on exchanges without using professional brokers. They may find themselves trading directly against professionals, such as cryptocurrency hedge funds.

Source: Reuters

Coin Exchange

- Buyers place buy-orders and sellers place sell-orders to the exchange.
- Buy-orders in its basic form comes with a maximum buying price per unit. The buyer is willing to purchase the coins as long as he/she can buy them at a price under the maximum buying price per unit.
- Sell-orders, similarly, comes with a minimum selling price per unit. A seller must have a certain amount of coins and is willing to sell them at a price equal or higher than the minimum selling price per unit.
- This business needs an online exchange portal where it connects the cloud storages like wallets, public ledger, traders and people at one place.
- Average exchange rate for UPbit is about 1% of exchanged fund. Per day, its about 3.5 billion KRW.
<http://www.yonhapnews.co.kr/bulletin/2018/01/02/0200000000AKR20180102043300008.HTML>
- Reference: <https://bitdeal.net/blog/post/bitcoin-exchange-business-plan-and-revenue-model>

On line wallets

- An online wallet is a wallet stored in the cloud, and you access it using a web interface on your computer or using an app on your smartphone.
- Some online wallet services were popular in 2015, Coinbase and blockchain.info.
- Your keys are stored in the site. At least it will have the ability to access your keys.
- Those keys ideally are stored under a password that only you know.
- It's convenience such as accessibility anywhere and anytime.
- But you simply have to trust them for keeping them in safety and not doing any unlawful transaction using your coins.
- For security, coins you should store them yourself.

Example TX at CryptoXchange

- Reference for further reading: page 112 ~ 114, Princeton_bitcoin_book.pdf.
- Suppose Alice holds 5000 dollars and three bitcoins in her account at CXchange .
- Bob holds 2000 dollars and four bitcoins in his account at CXchange .
- Alice put an order to buy 2 bitcoins at 500 dollars each. Fee is 10 dollars, 1% of TX money.
- Bob put an order to sell 2 bitcoins at a price above or equal to 500 dollars each. Fee is 10 dollars, 1% of TX money.
- CXchange matches up Alice and Bob and completes the transaction.
- After the match up transaction is completed, their account balances are changed to
- Alice has five bitcoins and 3990 dollars in her account.
- Bob has two bitcoins and 2990 dollars in his account

- Note here that no transactions have actually happened on the Bitcoin blockchain.
- All CXchange has done is changing the numbers in each account.
- It did not have to go through the blockchain to complete these exchange transactions. All it had done is to find the matchups.
- This practice of business is probably o.k. as long as the account holders at CXchange are satisfied. The exchange can have them satisfied as long as it retains the ability to give the money back in the account when it was asked to.

Kimchi premium, why?

- It is defined as the gap in bitcoin price in Korean exchanges compared to foreign exchanges.
- In December 2017, the demand for buying bitcoin in South Korea was at the peak.
- South Koreans has to pay a higher price for bitcoins than traders in other countries.
- This phenomenon was called the "kimchi premium." Kimchi is a Korean traditional side dish, fermented cabbage.
- The price difference was more than 40%, Korean bitcoin price was higher than the price in the United States.
- Investors can take the advantage of arbitrage.
- South Korean traders would first have to exchange the Korean won for a the U.S. dollar, to purchase a bitcoin on a US cryptocurrency exchange.
- Foreign investors would simply have to purchase bitcoins abroad and sell them on a South Korean exchange. The draw of profit should have eliminated this gap arbitrage, but [capital controls](#), financial regulations, and [anti-money laundering](#) laws make the process difficult.
- South Koreans and South Korean firms are limited to the amount of money they can move out of the country each year, and the transfer must be approved by regulators. Regulators are likely to block the transfer for fear that it is really being made to launder money.
- Even if regulators approved of the transfer, it may take so much time that the arbitrage opportunity is no longer available. Capital controls also limit the inflow of cryptocurrencies by foreign investors. This has created a scenario in which digital currencies can only be traded in South Koreans by South Koreans.

Read more: [Kimchi Premium Definition | Investopedia](#) <https://www.investopedia.com/terms/k/kimchi-premium.asp#ixzz5C5OJLbhF>
Follow us: [Investopedia on Facebook](#)

'주식 공매도 금지' 청와대 국민청원 20만 돌파

- 삼성증권 우리사주 배당 사고와 관련해 삼성증권을 규제하고 **공매도**(없는 주식을 빌려 파는 것)를 금지해 달라는 청와대 국민청원 참여자가 20만 명을 넘어섰다.
- 청원 게시자는 지난 6일 '삼성증권 시스템 규제와 공매도 금지'라는 제목의 청원에서 "삼성증권의 발행 한도는 1억2천만 주인데 우리사주 1주당 1천 주씩 총 28억 주가 배당됐고 500만 주가 유통됐다"며 "이는 없는 주식을 배당하고, 그 없는 주식이 유통될 수 있다는 이야기로 주식을 빌리지 않고도 공매도 할 수 있다는 이야기가 된다. 서민만 당하는 공매도를 꼭 폐지하고 이를 계기로 증권사의 대대적인 조사를 바란다"고 적었다. 이 청원은 10일 오후 2시께까지 20만5천여명이 참여했다. 삼성증권은 지난 6일 직원들이 보유한 우리사주 283만1620만 주를 대상으로 1주당 1천 원씩 배당금을 주기로 했으나, 직원의 **입력실수로 1주당 1천 주를 배당**하는 사고를 냈다. 삼성증권 직원들은 이때 28억3천만 주 가량을 배당받았고, 이들 가운데 16명은 500만 주 이상 매도해 6일 삼성증권 주가가 장중 11.68% 포인트 급락한 바 있다.
- 성연철 기자 sychee@hani.co.kr
- 원문보기: <http://www.hani.co.kr/arti/politics/bluehouse/839923.html#csidxb130b1b996922a7839eb7fd442fa0f0>

공매도란

- Short Selling (공매도): 없는 것은 판다는 의미. 개인 혹은 단체가 주식, 채권 등을 보유하지 않은 상태에서 매도하는 행위를 말한다. 하락장에서 쓰는 투자 수단.
- **What is a 'Short (or Short Position)'**
- A short, or short position, is a directional trading or investment strategy where the investor sells shares of borrowed stock in the open market. The expectation of the investor is that the price of the stock will decrease over time, at which point he will purchase the shares in the open market and return the shares to the [broker](#) which he borrowed them from.
- **What is a 'Long (or Long Position)'**
- A long (or long position) is the buying of a security such as a stock, commodity or currency with the expectation that the asset will rise in value. In the context of options, long is the buying of an [options contract](#). An investor that expects an asset's price to fall will go long on a [put option](#), and an investor that hopes to benefit from an upward price movement will be long a [call option](#).

Short and Long Positions

- <https://steemit.com/coin/@wjdtk915/6fqake>
- Steemit에 나온 공매수, 공매도 설명.



Why is Short Selling Legal? A Brief History

By [Adam Hayes, CFA](#) | Updated August 9, 2016 — 11:19 AM EDT

Short Selling Becomes Legitimate

- The SEC adopted Rule 10a-1 in 1937, [the uptick rule](#), which stated market participants could legally sell short shares of stock only if it occurred on a price uptick from the previous sale.
- Despite its new legal status and the apparent benefits of short selling, many policymakers, regulators – and the public – remained suspicious of the practice. Being able to profit from the losses of others in a bear market just seemed unfair and unethical to many people.
- The SEC eventually eliminated the uptick rule in 2007 following a years-long study which concluded that the regulation did little to curb abusive behavior and had the potential to limit market liquidity.

The "Naked" Short Sale is banned by SEC in 2009, as a means to driving price down.

- The seller must "locate" shares to sell to avoid "selling shares that have not been affirmatively determined to exist." In the U.S., broker-dealers are required to have reasonable grounds to believe that shares can be borrowed so they can be delivered on time before allowing such a short sale. Executing a naked short runs the risk that they will not be able to deliver those shares to whomever the receiving party in the short sale. Another prohibited activity is to sell short and then fail to deliver shares at the time of settlement with the intent of driving down an asset's price. (For more, see: [The Truth About Naked Short Selling](#).)

Read more: [Why is Short Selling Legal? A Brief History Investopedia](#) <https://www.investopedia.com/articles/investing/110614/why-short-selling-legal-brief-history.asp#ixzz5CcvmxZv1>

ISSUES

Reforming Wall Street

Wall Street cannot continue to be an island unto itself, gambling trillions in risky financial decisions while expecting the public to bail it out.

It is time to break up the largest financial institutions in the country.

The six largest financial institutions in this country today hold assets equal to about 60% of the nation's gross domestic product. These six banks issue more than two thirds of all credit cards and over 35% of all mortgages. **They control 95% of all derivatives and hold more than 40% of all bank deposits in the United States.** We must break up too-big-to-fail financial institutions. Those institutions received a **\$700 billion bailout** from the US taxpayer, and more than **\$16 trillion in virtually zero interest loans** from the Federal Reserve. Despite that, financial institutions made over \$152 billion in profit in 2014 – the most profitable year on record, and three of the four largest financial institutions are 80% bigger today than they were before we bailed them out. Our banking system must be part of the productive, job-creating economy. The Federal Reserve, a government entity which serves as the engine of the banking industry, must eliminate its internal conflicts of interest, provide stricter oversight, and **insist that the banks serve the economy in a way that works for everyone, not just a few.**



The Evolution of Trust

***Scientific American* 318, 38 - 41 (2018)**
Published online: 19 December 2017
| doi:10.1038/scientificamerican0118-38

Natalie Smolenski

- Banks and governments have in many ways failed to broker trust for the global economy, especially in the past few decades. Ordinary people have grown wary of centralized power and are seeking alternatives.
- Bitcoin—and blockchain technology in general—allows the brokering of trust to be shifted toward machines and away from human intermediaries such as bankers. This technology could design exploitation out of the system instead of punishing it later.
- Blockchains lend themselves both to human emancipation and to an unprecedented degree of surveillance and control. How they end up being used depends on how the software handles digital identity.

How much does electricity cost?

Average national electricity prices in US cents/kWh (2011)

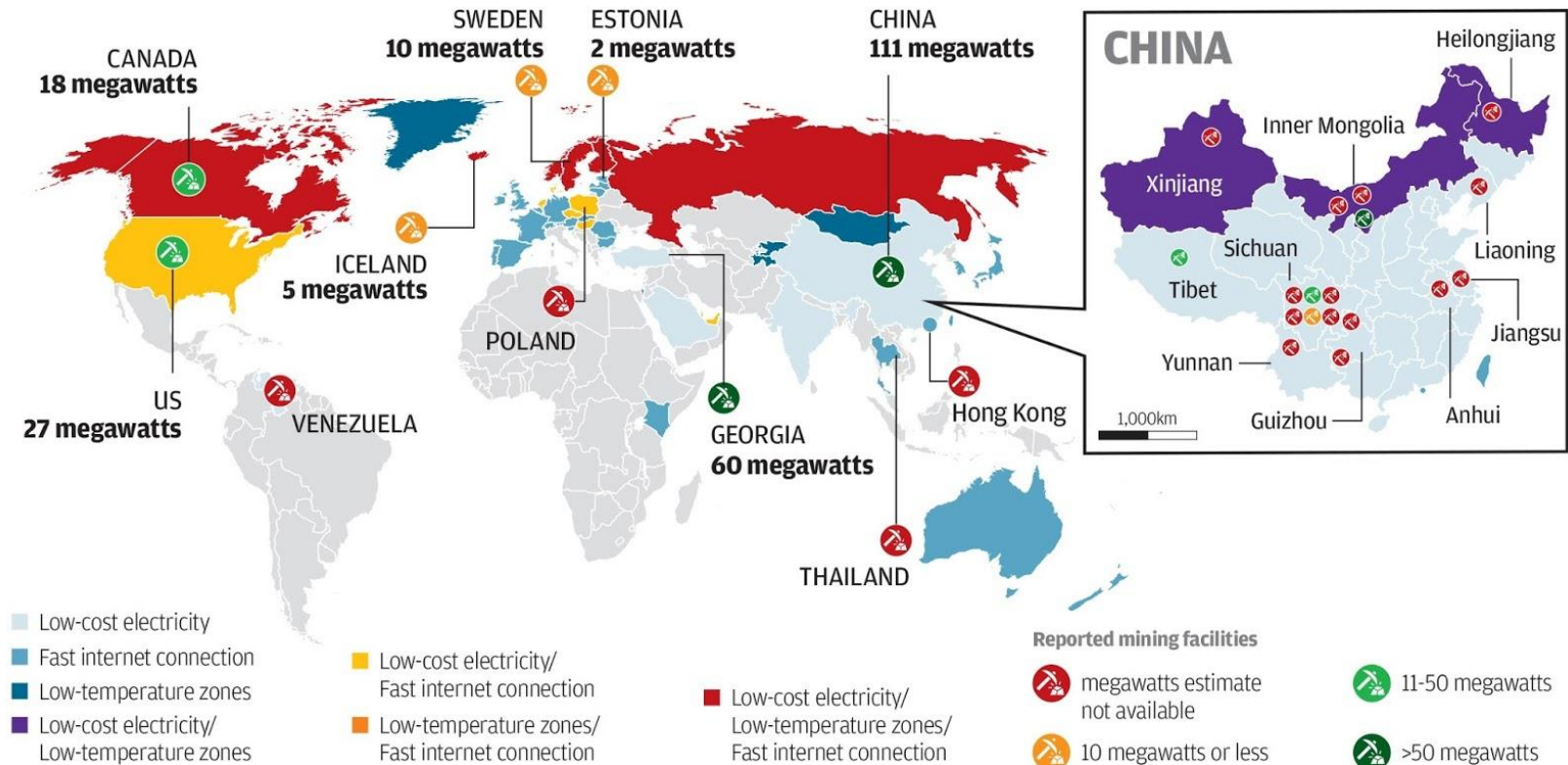


Data: average prices from 2011 converted at mean exchange rate for that year

Sources: IEA, EIA, national electricity boards, OANDA shrinkthatfootprint.com

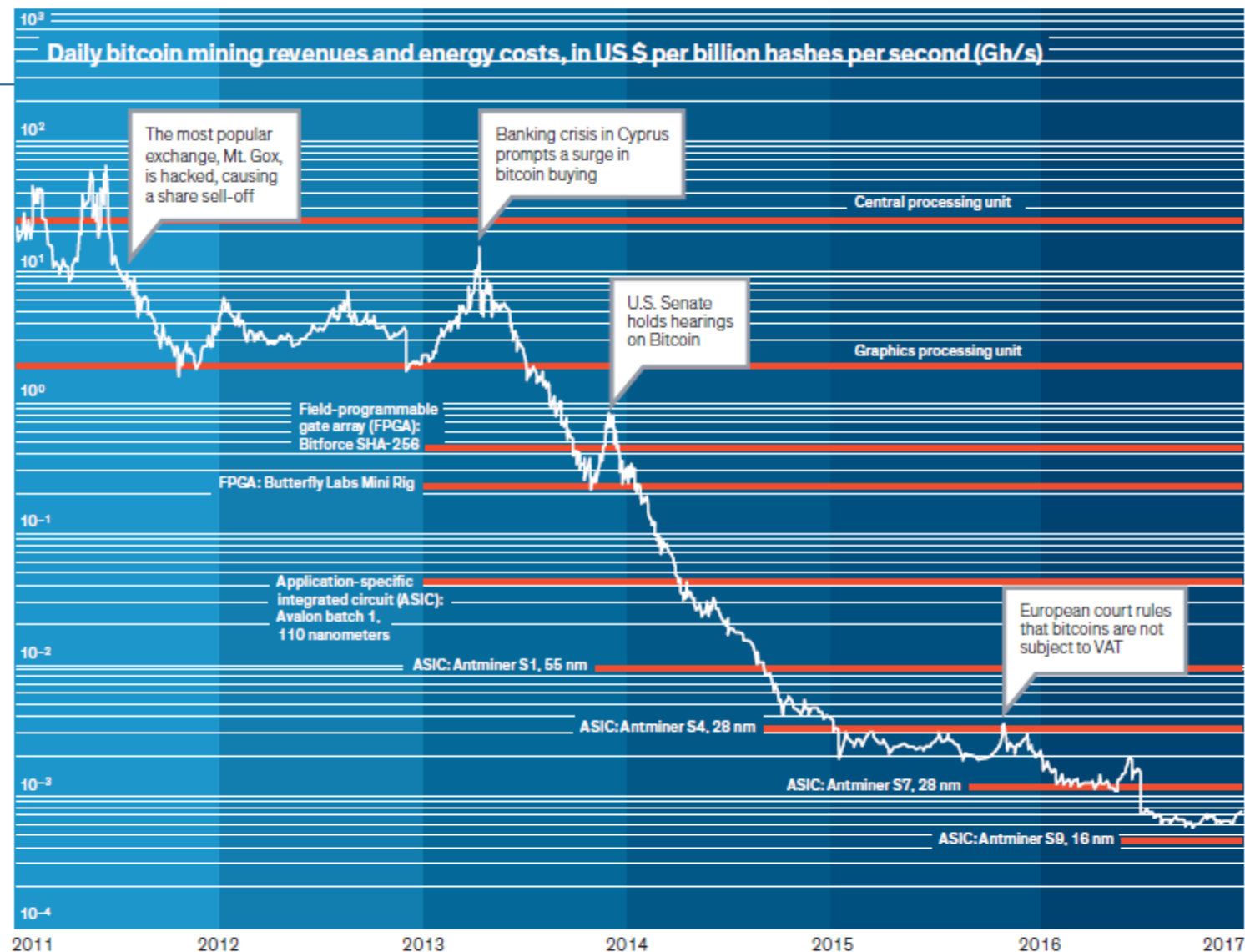
PoW, Monopolized?

Global cryptocurrency mining sites



Source: University of Cambridge

SCMP



Sisyphian Slide: Daily revenues for mining bitcoins [white], in US dollars per unit of computational power, are generally somewhat higher than the daily energy costs [red] of running the computers.

Proof of Work, any alternative?

- Proof-of-work has been monopolized today.
- Handful of mining sites are dominating the bitcoin mining.
- The trust has been degraded.
- No more one cpu one vote.
- Miners are by nature rational profit seekers.
- They use ASICs now.

B. Proof of Work—the Monopoly Problem

Proof of work is a key component of Bitcoin. Inherently, any task suitable as a basis for proof-of-work schemes needs to be difficult to solve, but trivial to verify. It often boils down to a

Karma implements a distributed currency by maintaining a so-called bank set. Karma also considered the effect of inflation and deflation and proposed means to adjust money creation. Probably Bit Gold is the most advanced approach of all precursors, as it chains the proof of work, uses the last entry to create the next challenge and adjusts the difficulty. However, it also relies on a quorum of hosts/addresses rather than a quorum of computing power. It is thus vulnerable to Sybil attacks. Finally, it was the Bitcoin protocol to combine Sybil resistance and coin minting by a sophisticated proof-of-work scheme.

Originally, the paradigm of proof of work is “one-CPU-one-vote” [16]. Bitcoin uses a CPU-bound function (i.e., SHA-256 [30]) as the basis for its proof-of-work scheme. Miners are by nature rational profit seekers. Their mining costs consist of expenses for mining hardware and ongoing energy cost. They strive to reach the break-even point as quickly as possible, to make as much profit as possible. The first miners used computers with ordinary CPUs to solve the proof of work. Even

Mining operations are highly parallelizable. Some graphics processors (GPUs) are therefore able to compute the repeated hash operations much faster and much more energy efficient than any CPU. GPU mining quickly replaced CPU mining.

(ASICs). FPGAs and especially ASICs significantly increased the speed and efficiency of mining. Since then, only ASICs (if at all) are economically viable for bitcoin mining. They achieve hash rates in the order of one terahash per second. In [207],

Pre-cursors to Bitcoin

- PoW is a gold, or a coin.
- Hashcash (92')
 - a proof-of-work system used to limit email spam
- RPOW (03') is a centralized currency.
 - Centralized approach: a server issues a coin in return for a PoW.
 - Coins are reusable and transferrable. The server checks the validity.
- B-money (98') is a decentralized currency.
 - Uses broadcast channel and a set of servers.
- Karma (03') is a distributed currency
 - Using a bank set
 - Coin creation is adjusted considering inflation and deflation.
- BitGold (05')
 - Most advanced of all precursors
 - Suggested to chain the proof-of-work (uses the last entry to create new puzzle and adjust difficulty)
 - But relied on IP addresses and thus vulnerable to Sybil attack.
- Bitcoin uses a CPU-bound function SHA256 for PoW.

Items to consider for new PoW

- One way is to diversify the puzzles and change the puzzle over time.
- Considerations for new puzzles
 - A puzzle should be difficult to solve but very easy to check.
 - The puzzle should be resistant to attacks.
 - Solution to the puzzle for a block should not be reusable.
 - Puzzle difficulty should be adjustable.
 - Anyone with a cpu wishes to participate should be able to join.
 - Better if the puzzle is not parallelizable.

Fundamental Requirements

- PoW is good but what would occur when the mining rewards go to zero.
 - Would transaction fees be good enough to keep the miners in the bitcoin network?
 - Nash Equilibrium and Tragedy of Commons suggests there is a significant risk for security with the current PoW and Mint system.
- First, The block generation must be somehow “expensive” and Individual miners shall not be able to gain an over proportionally high ability to mint coins.
- Second, consensus must eventually be reached; there must be a common rule to resolve forks and to determine the main block chain.
- Third, it must be forgery proof.
- Reference: See page 2114, IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 18, NO. 3, THIRD QUARTER 2016.

Proof of Stake

It turns out that *coin age* is a viable alternative to proof of work. Coin age is defined as the currency amount times the holding period [227]. For example, if Alice transfers two coins to Bob and Bob held the coins for 90 days, the coin age is 180 coin-days. When Bob spends the two coins, the coin age he accumulated is destroyed.

The idea to use the coin age to define the reward is known as *proof of stake* [227] (PoS). It is, for example, implemented in *Peercoin* (PPC, peercoin.net). Mining a proof-of-stake block requires to construct a so-called *coinstake block* (named after Bitcoin's coinbase transaction). In a coinstake transaction, owners send coins in their possession to themselves and add a predefined percentage as their reward. Analogue to proof of work, a hash value below or equal to a target value is required to successfully mint a block. In contrast to proof of work (and Bitcoin), the difficulty is individually determined: it is inversely proportional to the coin age. Because the hash is—except for a timestamp—calculated on static data, there is no way for miners to use their computational power to solve the puzzle faster than others. In particular, there is no nonce which can be modified. Instead, every second the timestamp changes and miners have a new chance of finding the solution. If they find a solution, they broadcast the block including the coinstake transaction. The coinstake transaction assigns the reward to the miner, but also resets the coin age. Of course, new coin age can subsequently be accumulated again, slowly increasing the chances of solving the puzzle next time.

value into the proof of work, proof of stake eliminates the high energy consumption altogether. It shifts from a highly competitive tournament to a raffle-like scheme, with repeatedly occurring new chances for all participants. In addition, miners destroy the coin age by claiming the reward; they do not keep it for the next round. This gives others the chance to “win the raffle”, too.

These properties mitigate the risk of monopoly in the tragedy of the commons problem. Note that the voting power is more equally distributed. Thus, “rich gets richer” is complemented by “poor gets richer” [228], meaning every participant can provide a proof of stake, thus help to secure the block chain and in return get a reward in proportion to their holding.

Besides, exploiting a proof of stake-based currency seems much more expensive. In contrast to proof of work, where the longest block chain survives, proof of stake declares the block chain with the highest total sum of destroyed coin age as the main chain. In order to perform an attack similar to the 51%-attack, an attacker must hold a huge amount of coins, which even when destroying the coin age suffices to gain more than half of the odds. It is assumed that the cost for gaining the majority of computing power in a proof-of-work scheme (e.g., for hardware) is smaller than the cost for buying enough coins in a proof-of-stake setting. However, there are also objections to this claim [229], suggesting that the attack would be anticipated and thus coins would be ditched, which reduces the attacker's costs. Nevertheless an attacker holding lots of coins would suffer severely from ruining the currency, which probably reduces the incentive to do so in the first place.

Proof of Activity

The major weakness, though, is that **coin age accumulates even when the node is not connected to the network**. It suffices when nodes come online occasionally and wait for their reward, only to go offline again afterwards. This behavior will result in a more bursty reward distribution than in the case where nodes remain online all the time, but stakeholders most likely will accept that. The lack of a sufficient number of online nodes, though, facilitates attacks.

In [234] the author follows the idea that a **higher activity produces a healthier economy**. He identifies the problem that coin age is a linear function of time. In practice, Peercoin implements an upper and a lower bound of coin age to mitigate some of the mentioned problems. However, changing the increment function to an exponential decay function, for example, would have a profound impact. In such a setting, the increment rate of the coin age decreases with time and asymptotically converges to zero. Parameterizing the decay constant allows for a deliberate specification of the function's half-life time. This changes the incentive: a fresh coin accumulates coin age much faster, up to a fixed value. The intention is to reduce the resistance to trade coins and to encourage users to stay online. It is conceivable to employ other functions, such as non-monotonic and/or periodic functions. That would imply to punish hoarding coins even more, or to reflect a seasonal pattern. The basic idea is termed **proof of stake velocity** and is implemented in **Reddcoin** (RDD, reddcoin.com).

The approach by [226] is to **directly reward active peers for their contribution**. The idea is to raffle a fraction of the proof-of-work block reward among all active nodes, while their stake determines the amount of raffle tickets, i.e., their chances of winning. **It is thus a combination of proof of work and proof of**

stake. In detail, miners mine “empty” blocks only. If they solve the proof-of-work puzzle, they broadcast it in the network as before. Everybody receiving the block derives N deterministic pseudorandom ticket numbers from it. The first $N - 1$ most lucky stakeholders sign the block with their respective private key and broadcast the signature. If the N -th most lucky stakeholder sees the block, she creates a wrapper, includes the block, all transactions, the $N - 1$ signatures, adds her own signature and broadcasts the wrapped block. Others will consider it as a legitimate extension of the block chain if the block and the lucky stakeholders are valid. Finally the transaction fees are shared by stakeholders and the miner.

In order to determine the N lucky stakeholders, the pseudorandom number is interpreted as an index in the list of all so far minted satoshis. Finding the user in possession of the satoshi is done by a procedure called “follow-the-satoshi”. Everybody can verify it by inspecting the block chain and following the satoshi from the coinbase transaction up to the address currently holding it. Please note that this is much like proof of stake, with the difference that the coin age is irrelevant. Alice holding two coins has twice as high a chance to be picked as Bob holding one coin. The decision to share the transaction fees as reward among the stakeholders and not, for example, the entire block reward is rooted in the observation which we pointed out earlier: a high reward for the stake incentivizes undesirable coin hoarding. The small fees are considered a nice bonus, but not an incentive for hoarding.

Proof of Publication

- Recall the timestamp server of the bitcoin white paper!
- Bitcoin provides a secure distributed timestamping service, with an accuracy of about 10 min.

E. *Proof of Publication—Provable Commitments*

After looking at various proof-of-X schemes, let us close the circle and come to another achievement of computing history which apparently influenced the Bitcoin design, namely **secure timestamping**. A timestamping service provides **timestamps for digital documents, which securely keep track of the creation and modification time of the document**. There are many timestamping schemes, such as PKI-based centralized services, where documents and timestamps are hashed and secured by the private key of the timestamping server. However, the server can easily backdate documents by hashing and signing a previous timestamp. Thus, the approach comes with the premise of trusting the timestamping server.

The authors of [40] come to a similar conclusion and propose a carbon dating commitment protocol based on Bitcoin, namely **CommitCoin**. The idea is, in a more general context, also known as *proof of publication* and comes in various manifestations. In the case of CommitCoin, a Bitcoin address is generated which encodes the information of a respective document. Other use cases include coin tosses [238], lotteries [239], or decentralized poker [240]. This works without a central entity and without the need to trust each other, that is, secure multi-party computations (MPC). MPC enables such use cases, however, comes with the caveat that it cannot enforce payouts or compensation. Cryptocurrencies are, therefore, a natural choice for combining MPC with money: players bet coins by issuing elaborate transactions, i.e., commitments, while still not trusting each other or any kind of third party. Such protocols are not limited to gambling and generalizations exist [183], [241].

TABLE IV
SUMMARY OF ALTCOINS AND EXTENSIONS

	Approach	Distinct Feature (incl. References)	Sec.
Precursor	B-Money	Mining reward proportional to proof of work difficulty; requires a broadcast channel [7]	II-B, V-D, V-E
	Bit Gold	Chained proof of work [10]; Byzantine-resilient quorum [13]	III-B, V-D, V-E
	Karma	Distributed currency maintained by a bank set [8]	V-E
	RPOW	Centralized (reusable) proof of work exchange/ bank [9]	V-E
Altcoins	Bitshares (BTS)	Delegated proof of stake [231]	V-F
	Bytecoin (BCN)	Implements CryptoNote [190], which aims for unlinkable and untraceable transactions	V-C, V-E
	Counterparty (XCP)	Colored coin; used proof of burn	V-H, V-H
	Cryptonite (XCN)	Implements the mini block chain scheme [127]	IV-D
	Dash (DASH)	Formerly known as Darkcoin; implements native CoinJoin-like transactions [178]	V-C
	Dogecoin (DOGE)	Block payload holds TXIDs only; fast block generation	IV-D, V-E
	Litecoin (LTC)	Uses scrypt [214] to foster distributed power among miners	V-E
	Mastercoin (MSC)	Colored coin; exodus address	V-H
	Nextcoin (NXT)	Entirely proof of stake based	V-F
	Peercoin (PPC)	Identified coin age as alternative measure; proof of stake [227]	V-F
	Primecoin (XPM)	Proof of work with intrinsic value i. e. prime chains [218]	V-E
	Reddcoin (RDD)	Proof of stake velocity [234]	V-E
	RSCoin	Centrally controlled money supply with distributed verification [126]	IV-D
	Ripple (XRP)	Implements a novel Byzantine agreement protocol [200]	V-D
Zerocash	Full-fledged altcoin, carrying on the ideas of Zerocoin [189]	V-C	
Altchains	Bitmessage	Secure messaging service [145]	IV-G
	Ethereum (Ether)	Turing complete smart contract processing [44], [45]	II-E
	Namecoin (NMC)	Key-value storage; realizes decentralized domain name coordination [143]	IV-G
	Permcoin	Decentralized file storage; proposes proof of retrievability [100]	V-E
Protocols / Extensions	CoinJoin	Uses multi-signature transactions to enhance privacy [160]	V-C
	CoinShuffle	Decentralized protocol to coordinate CoinJoin transactions [180]	V-C
	CoinSwap	Enables P2P-based trustless mixing [41]	V-C
	CommitCoin	Secure timestamping protocol [40]	V-H
	Mini block chain	Identifies individual block chain components [127]	IV-D
	Mixcoin	Mixing with accountability [174]	V-C
	Zerocoin	Unlinkable and untraceable transactions by employing zero knowledge proofs [187]	V-C

Difficulty

- The lower the target is, the more difficult the puzzle is.
- Recall our lecture note set 1. If target is a hash with 10 starting hexadecimal zeros, it would take on the average 2^{40} trials to find a good hash.
- Among the network participants, your chance of winning a mining game is your hash power, i.e., the hash rate percentage of your mining network.
- The target value is adjusted every 2016 blocks so that one block is mined every 10 minutes. On the average, it takes 2 weeks to mine 2016 blocks.
- The new target is thus given by
$$T_{new} = T_{old} \frac{\text{actual time taken to mine 2016 blocks}}{2016 \cdot 10 \text{ min}}$$
- In Bitcoin, the difficulty is used to indicate how difficult it is to find a hash below a given target.
- Difficulty is defined as the ratio of the maximum allowed target to the current target.
- The maximum allowed target value is 2^{224} at which difficulty is 1.
 - $256 - 224 = 32$ bits or $32/4 = 8$ hexadecimals.
 - Target with 32 leading zero bits is the minimum difficulty!

Difficulty today = $3.5e12$

- Block #516447
- **BlockHash** 0000 0000 0000 0000 0006 082b 0352 8b8d d578 fa70
b5f5d1b1af62930a139a89b6
- **Difficulty** 3,511,060,552,899.7197 = $3.5e12$
 - Notice from difficulty that target = $2^{\text{floor}(\log_2(2^{224}/\text{diff}))} = 2^{182}$.
 - Thus, the target hash has $256 - 182 = 74$ leading zero bits.
 - Thus, the probability of successful mining per hash is 2^{-74} .
 - On the average, it would take $2^{74} \sim 1.8e22$ hashes to mine a single block.
- The hash rate of the bitcoin network should be
$$1.88e22/600 = 3.14 \text{ e}19 \text{ [hashes/sec]} = 31.4 \text{ [exa H/sec]}.$$
- Today's hash rate at blockchain.info is 30 exaH/sec.
<https://blockchain.info/ko/charts/hash-rate>

An Efficient Elliptic Curve Digital Signature Algorithm (ECDSA)

Shweta Lamba

M. Tech Scholar, Computer Science Department
Technological Institute of Textile & Sciences
Bhiwani, India
shweta.c.lamba@gmail.com

Monika Sharma

Assistant Professor, IT Department,
Technological Institute of Textile & Sciences
Bhiwani, India
monikasharma1510@gmail.com

Abstract—In recent years, Elliptic Curve Cryptography (ECC) has attracted the attention of researchers and product developers because of its robust mathematical structure and highest security in comparison to other existing algorithms like RSA (Rivest Adleman and Shameer Public key Algorithm). Elliptic Curve Digital signature represents one of the most widely used security technologies for ensuring un-forge-ability and non-repudiation of digital data. Its performance heavily depends on an operation called point multiplication. Furthermore, root cause of security breakdown of ECDSA is that it shares three points of the elliptic curve publically which makes it feasible for an adversary to gauge the private key of the signer. In this paper we proposed a new ECDSA which involves not as much of point multiplication operations as in existing ECDSA and shares only two curve points with everyone. The proposed method also reduces the point addition and point doubling operations. It is found to be more secure in contrast to existing ECDSA.

Keywords—ECDSA, ECC, Point multiplication, Point Addition, Signature generation, Signature verification.

information held by an entity into a tag called signature. Digital signature represents one of the most widely used security technologies for ensuring un-forge-ability and non-repudiation of digital data.

The Elliptic Curve Digital Signature Algorithm is the Elliptic Curve analogue to the more widely used Digital Signature Algorithm (DSA). It is the application of ECC to digital signature generation and verification. Its security is based on the elliptic curve discrete logarithm problem [2] (ECDLP).

ECDSA was first anticipated in 1992 by Scott Vanstone in response to NIST's request for public comments on their first proposal for DSS. It was accepted in 1998 as an ISO standard (ISO 14888-3), accepted in 1999 as an ANSI standard (ANSI X9.62), and accepted in 2000 as an IEEE standard (IEEE 1363-2000) and a FIPS standard (FIPS 186-2). It is now in many standards or recommendations, such as IEEE standard (IEEE 1363-2000) and FIPS standards (FIPS 186-3).

Elliptic curve digital signature algorithm

- Taken from the paper, these sentences are almost generic for signatures and authentication that we have done with RSA.
- The steps involved in ECDSA are formation of key-pair, signature-generation and signature-verification.
- The digital signature is typically created using the hash function.
- The transmitter sends the encrypted data along with signature to the receiver.
- The receiver in possession of sender's public key and domain parameters can authenticate the signature.
- The prime q of the finite field Fq , the equation of the elliptic curve E , the point G on the curve and its order n , are the public domain parameters.
- Furthermore, a randomly selected integer d from the interval $[1, n-1]$ forms a private key.
- Multiplying G by the private key k , which is called scalar multiplication, will generate the corresponding public key Q .
- The pair (K, k) forms the ECC public-private key pair with K the public key and k is the private key.
- The generating point G , the curve parameters 'a' and 'b', together with few more constants constitute the domain parameters of ECC.

Elliptic Curve Digital Signature Algorithm

Alice sends a signed message to Bob (Curve equation, G , n)

Public domain

1. Use a designated hash function $H(*)$
2. A curve : $y^2 = x^3 + 7$ over F_q , $q \sim$ prime.
3. $G = (x, y)$, a point on the curve
4. n the multiplicative order of G

KeyGenerate

Out: k (private key), K (public key)

1. Select an integer k in $[0, n-1]$.
2. Compute $K = kG$.
3. K and $G \sim$ points on the curve.
4. The key-pair is (k, K) .

Results: Alice's pair (k_A, K_A) and Bob's pair (k_B, K_B)

SignGenerate

In: m the message, Alice's private key k_A

Out: Alice' signature (r, s)

1. Calculate the message hash $e=H(m)$
2. Let z be the L_n leftmost bits of e where L_n is the bit length of the group order n .
3. Select an integer d from $[1, n-1]$.
4. Calculate the curve point $(x_1, y_1)=dG$.
5. Calculate $r=x_1 \bmod n$. If $r=0$, go to step 3.
6. Calculate $s=k^{-1}(z+rk_A) \bmod n$. If $s=0$, go to step 3.
7. The signature is the pair (r, s) .

https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm

Elliptic Curve Digital Signature Algorithm

Alice sends a signed message to Bob (Curve equation, G , n)

isSignatureValid

In: m the message,
Alice' signature (r, s) , and K_A

Out: Valid or invalid

1. Verify if K_A is a valid curve point as follows:
 1. Check to see if K_A is not equal to the identity element O
 2. Check to see if K_A lies on the curve
 3. Check that $n \times K_A = O$
2. Verify that r and s are integers in $[1, n-1]$. If not, the signature is invalid.
3. Calculate $w = s^{-1} \bmod n$.
4. Calculate $u_1 = z w \bmod n$ and $u_2 = r * w \bmod n$.
5. Calculate the curve point $(x_1, y_1) = u_1 * G + u_2 * Q_A$. If $x_1, y_1 = O$, then the signature is invalid.
6. The signature is valid if $r \equiv x_1 \bmod n$, invalid otherwise.

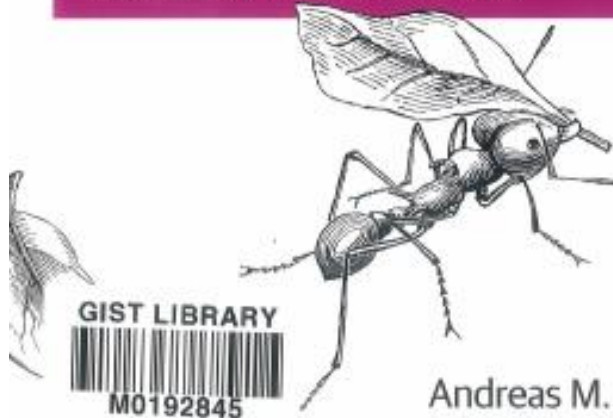
Reference https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm

O'REILLY

2nd Edition

Mastering Bitcoin

PROGRAMMING THE OPEN BLOCKCHAIN



GIST LIBRARY



M0192845

Andreas M. Antonopoulos

Private and Public Keys

A bitcoin wallet contains a collection of key pairs, each consisting of a private key and a public key. The private key (k) is a number, usually picked at random. From the private key, we use elliptic curve multiplication, a one-way cryptographic function, to generate a public key (K). From the public key (K), we use a one-way cryptographic hash function to generate a bitcoin address (A). In this section we will start with generating the private key, look at the elliptic curve math that is used to turn that into a public key, and finally, generate a bitcoin address from the public key. The relationship between private key, public key and bitcoin address is shown below:

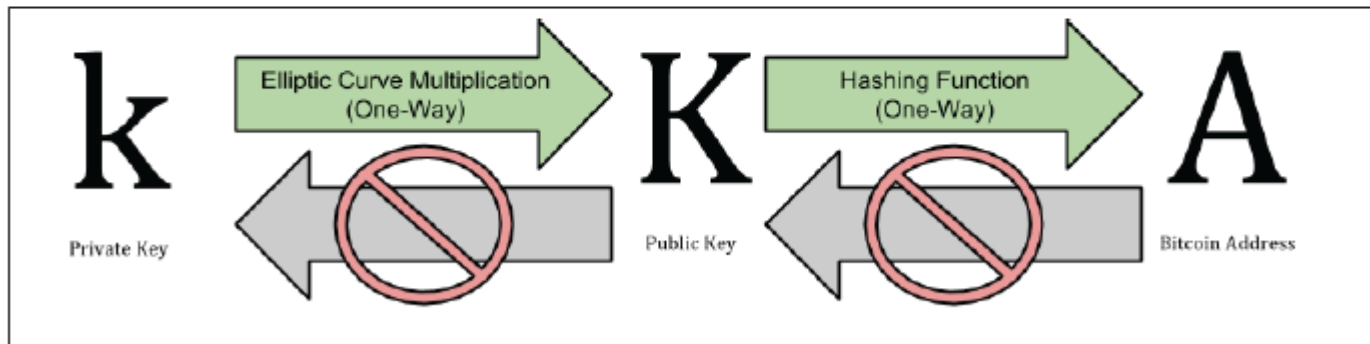


Figure 4-1. Private Key, Public Key and Bitcoin Address

Private key, public key, address

(256 bits shown as 64 hexadecimal digits, each 4 bits):

$k =$ 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD

$G = (x, y) =$ (55066263022277343669578718895168534326250603453777594175500187360389116729240,
32670510020758816978083085130507043184471273380659243275938904335757337482424)

Multiply the private key k with the generator point G to find the public key K .

$K = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD * G$

Public Key K defined as a point $K = (x, y)$.

$K = (x, y)$

where,

$x = F028892BAD...DC341A$

$y = 07CF33DA18...505BDB$

Bitcoin Address

From the public key Bitcoin addresses are generated by going through SHA256 and RIPEMD160

A modified Base 58 [binary-to-text encoding](#) known as **Base58Check** is used for encoding [Bitcoin addresses](#).

More generically, Base58Check encoding is used for encoding byte arrays in Bitcoin into human-typable strings.

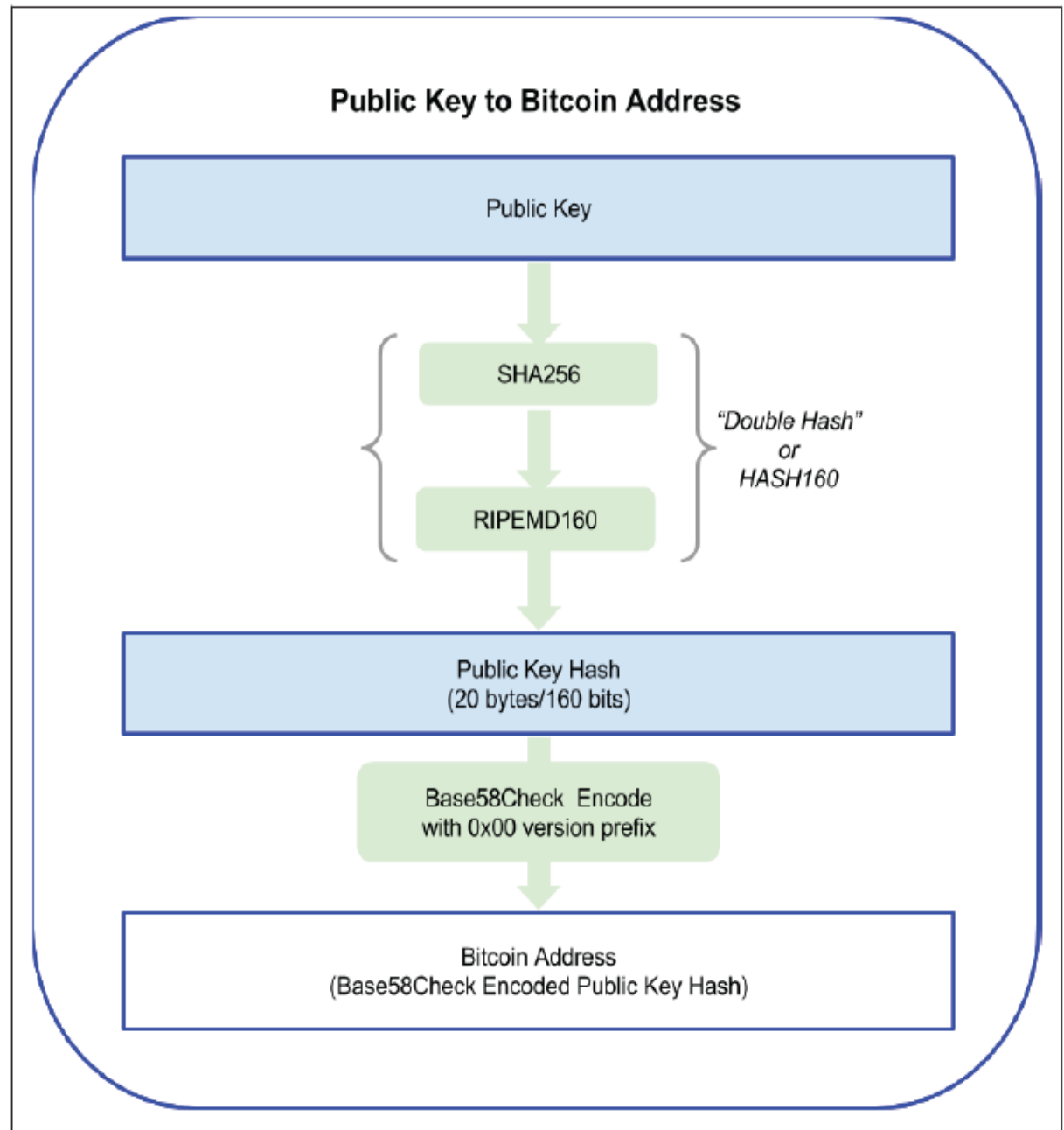


Figure 4-5. Public Key to Bitcoin Address: Conversion of a public key into a bitcoin address

Base58 symbol chart

Value	Character	Value	Character	Value	Character	Value	Character
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Base58Check Version Prefix

Table 4-1. Base58Check Version Prefix and Encoded Result Examples

Type	Version prefix (hex)	Base-58 result prefix
Bitcoin Address	0x00	1
Pay-to-Script-Hash Address	0x05	3
Bitcoin Testnet Address	0x6F	m or n
Private Key WIF	0x80	5, K or L
BIP38 Encrypted Private Key	0x0142	6P
BIP32 Extended Public Key	0x0488B21E	xpub

A bitcoin address from a public key

$K = (x, y)$ a public key

x coordinate=

7a633d546e723c3f41794549272f63617057382a227b6d393b35303d38

y coordinate=

44437a7439746e35565d3a27713c706423557e78444f4e767a22515724



These numbers are shown in Hexadecimal format, or 256 binary digits shown as 64 hexadecimal digits. If the number was shown in decimal format it would be 10^{77} figures long.

If you take these two coordinates and concatenate them i.e. join them end to end to make a 128 characters long string in Hexadecimal format, and then hash them whilst adding to the front a 1 (to indicate an address on the main network, if the address was for the testnet it would start with an m or an n).

Public_K=G Private_K=(x,y)

Address=(Network Version) & Ripemd160(sha256(x&y) & checksum

There is also the checksum to add which is essentially a hash of the address of the hash of the address – this is to check that the address is what it is – to stop typos et al.

Checksum=First four bytes of sha256(sha256((Network Version)&Ripemd160(sha256(x&y)))

A bitcoin address from a public key

The last step is to change the coding structure into a more readable format or Base58 in the case of Bitcoin. Base 58 is similar to base 64 but with a few characters removed. Base64 uses A-Z, a-z, 0-9, + and /.

Base 58 uses the same symbols but removes +, /, 0, O, l and I. All the symbols that could be confused for each other are removed making the format readable. The end result is a Bitcoin address of between 27 and 34 characters long! Such as below.

```
1BitBE9zZDwTGhXjwPSapWtViWJf2NJYyt
```

Questions still remained

- What should be the message m in bitcoin transaction?

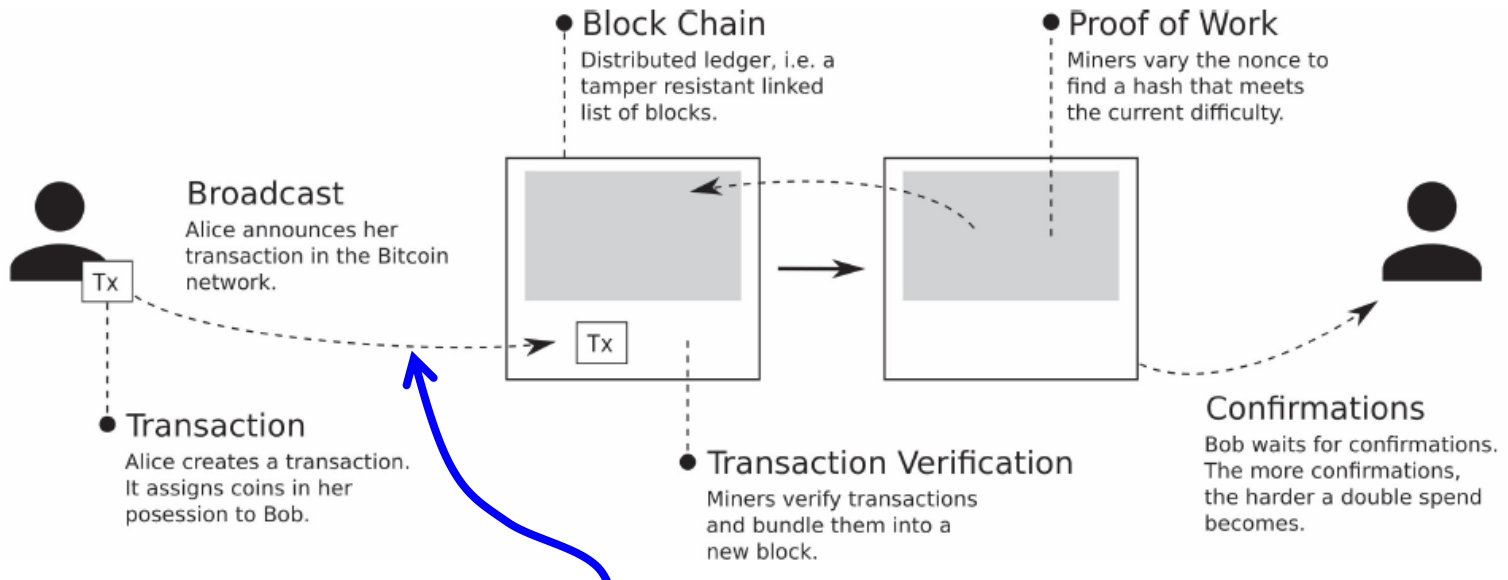


Fig. 4. Bitcoin's building blocks explained.

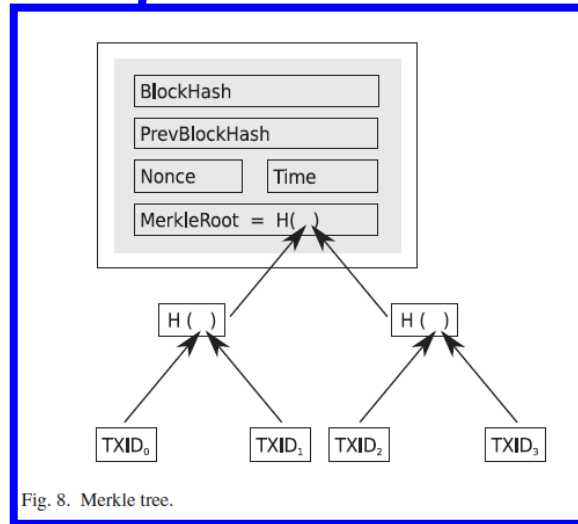


Fig. 8. Merkle tree.

Bitcoin Transactions

Create 25 coins and credit to Alice	ASSERTED BY MINERS
Transfer 17 coins from Alice to Bob	SIGNED(Alice)
Transfer 8 coins from Bob to Carol	SIGNED(Bob)
Transfer 5 coins from Carol to Alice	SIGNED(Carol)
Transfer 15 coins from Alice to David	SIGNED(Alice)

Figure 3.1 an account-based ledger

1	Inputs: \emptyset Outputs: 25.0→Alice	
2	Inputs: 1[0] Outputs: 17.0→Bob, 8.0→Alice	SIGNED(Alice)
3	Inputs: 2[0] Outputs: 8.0→Carol, 9.0→Bob	SIGNED(Bob)
4	Inputs: 2[1] Outputs: 6.0→David, 2.0→Alice	SIGNED(Alice)

Figure 3.2 a transaction-based ledger, which is very close to Bitcoin

Code for bitcoin transaction

```
    {  
      "hash": "5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",  
      "ver": 1,  
      "vin_sz": 2,  
      "vout_sz": 1,  
      "lock_time": 0,  
      "size": 404,  
      "in": [  
        {  
          "prev_out": {  
            "hash": "3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",  
            "n": 0  
          },  
          "scriptSig": "30440..."  
        },  
        {  
          "prev_out": {  
            "hash": "7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",  
            "n": 0  
          },  
          "scriptSig": "3f3a4ce81...."  
        }  
      ],  
      "out": [  
        {  
          "value": "10.12287097",  
          "scriptPubKey": "OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"  
        }  
      ]  
    }
```

metadata

input(s)

output(s)

Figure 3.3 An actual Bitcoin transaction.

As you can see in Figure 3.3, there are three parts to a transaction: some metadata, a series of inputs, and a series of outputs.

- **Metadata.** There's some housekeeping information — the size of the transaction, the number of inputs, and the number of outputs. There's the hash of the entire transaction which serves as a unique ID for the transaction. That's what allows us to use hash pointers to reference transactions. Finally there's a "lock_time" field, which we'll come back to later.
- **Inputs.** The transaction inputs form an array, and each input has the same form. An input specifies a previous transaction, so it contains a hash of that transaction, which acts as a hash pointer to it. The input also contains the index of the previous transaction's outputs that's being claimed. And then there's a signature. Remember that we have to sign to show that we actually have the ability to claim those previous transaction outputs.
- **Outputs.** The outputs are again an array. Each output has just two fields. They each have a value, and the sum of all the output values has to be less than or equal to the sum of all the input values. If the sum of the output values is less than the sum of the input values, the difference is a transaction fee to the miner who publishes this transaction.

And then there's a funny line that looks like what we want to be the recipient address. Each output is supposed to go to a specific public key, and indeed there is something in that field that looks like it's the hash of a public key. But there's also some other stuff that looks like a set of commands. Indeed, this field is a script, and we'll discuss this presently.

UTXO

- Unspent transaction outputs
 - cf) STXO
- Use `listunspent` to get all the unspent outputs belongs to an address.
- Use this result to create a new transaction.

Creating, signing and submitting transactions based on unspent outputs

Commands: `listunspent`, `gettxout`, `createrawtransaction`, `decoderawtransaction`, `signrawtransaction`, `sendrawtransaction`

Bitcoin's transactions are based on the concept of spending "outputs", which are the result of previous transactions, to create a transaction chain that transfers ownership from address to address. Our wallet has now received a transaction that assigned one such output to our address. Once this is confirmed, we can now spend that output.

First, we use the `listunspent` command to show all the unspent **confirmed** outputs in our wallet:

```
$ bitcoin-cli listunspent
[
  {
    "txid" : "9ca8f969bd3ef5ec2a8685660fdbf7a8bd365524c2e1fc66c309acbae2c14ae3",
    "vout" : 0,
    "address" : "1hvzSofGwT8cjb8JU7nBsCSfEVQX5u9CL",
    "account" : "",
    "scriptPubKey" : "76a91407bdb518fa2e6089fd810235cf1100c9c13d1fd288ac",
    "amount" : 0.05000000,
    "confirmations" : 7
  }
]
```

We see that the transaction `9ca8f9...` created an output (with `vout` index 0) assigned to the address `1hvzSo...` for the amount of 50 millibits, which at this point has received 7 confirmations. Transactions use previously created outputs as their inputs by referring to them by the previous `txid` and `vout` index. We will now create a transaction that will spend the 0th `vout` of the `txid` `9ca8f9...` as its input and assign it to a new output that sends value to a new address.

Closer look at txid 9ca8..., vout 0

- Use `gettxout` to take a closer look
- Shows the details of that transaction
- blockhash, confirmations, value 0.05, given to 1hvz... address, signature.

First, let's look at the specific output in more detail. We use the `gettxout` to get the details of this unspent output above. Transaction outputs are always referenced by txid and vout, and these are the parameters we pass to `gettxout`:

```
$ bitcoin-cli gettxout 9ca8f969bd3ef5ec2a8685660fdbf7a8bd365524c2e1fc66c309acbae2c14ae3 0
{
  "bestblock" : "0000000000000001405ce69bd4ceebcdfdb537749cebe89d371eb37e13899fd9",
  "confirmations" : 7,
  "value" : 0.05000000,
  "scriptPubKey" : {
    "asm" : "OP_DUP OP_HASH160 07bdb518fa2e6089fd810235cf1100c9c13d1fd2\

    OP_EQUALVERIFY OP_CHECKSIG",
    "hex" : "76a91407bdb518fa2e6089fd810235cf1100c9c13d1fd288ac",
    "reqSigs" : 1,
    "type" : "pubkeyhash",
    "addresses" : [
      "1hvzSofGwT8cjb8JU7nBsCSfEVQX5u9CL"
    ]
  },
  "version" : 1,
  "coinbase" : false
}
```

What we see above is the output that assigned 50 millibits to our address 1hvz.... To spend this output we will create a new transaction. First, let's make an address to which we will send the money:

```
$ bitcoin-cli getnewaddress
1LnFTndy3qzXGN19Jwscj1T8LR3MVe3JDb
```

Scrap up all UTXOs to make a transaction
 It has TXID, value at each output.
 Later transactions point to TXID and output no.

Transaction as Double-Entry Bookkeeping			
Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC
	<i>Inputs</i>		<i>0.55 BTC</i>
	- <i>Outputs</i>		<i>0.50 BTC</i>
	<i>Difference</i>		<i>0.05 BTC (implied transaction fee)</i>

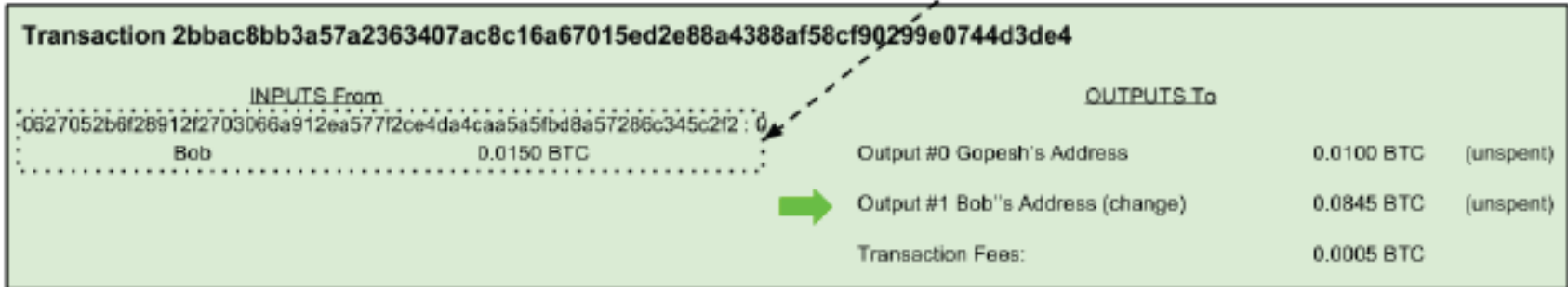
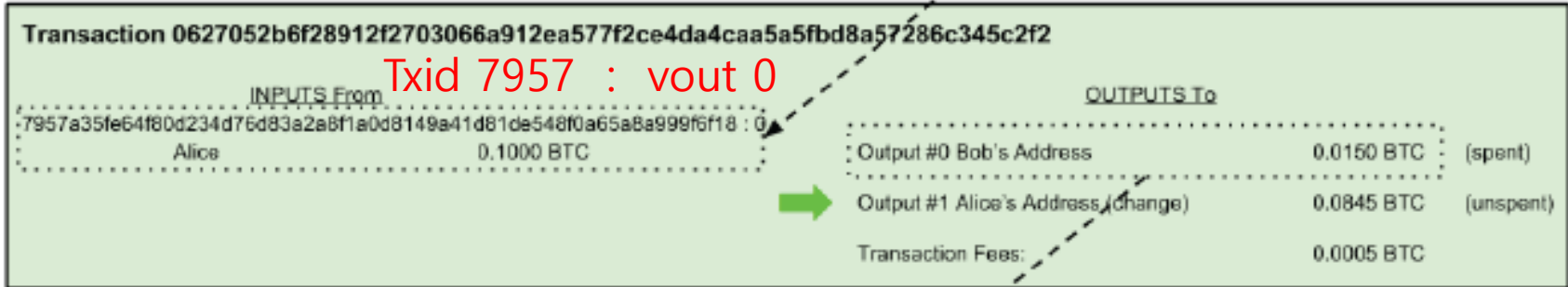
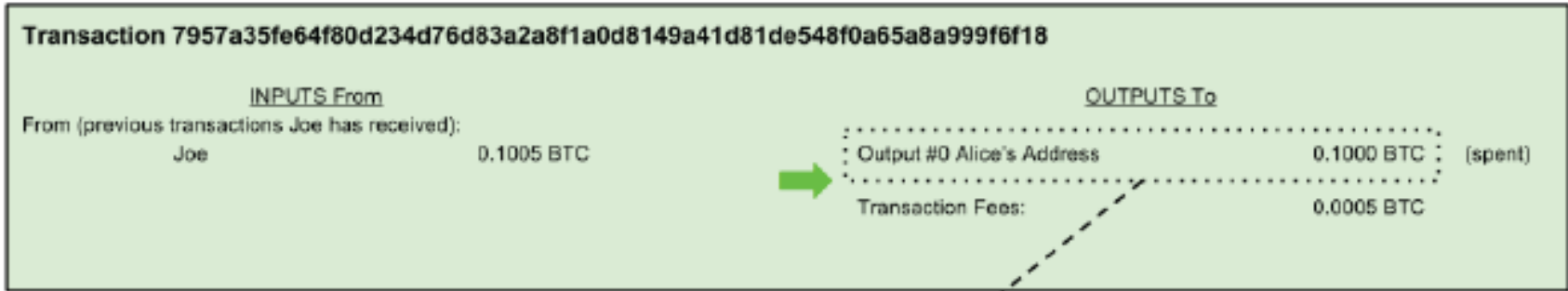
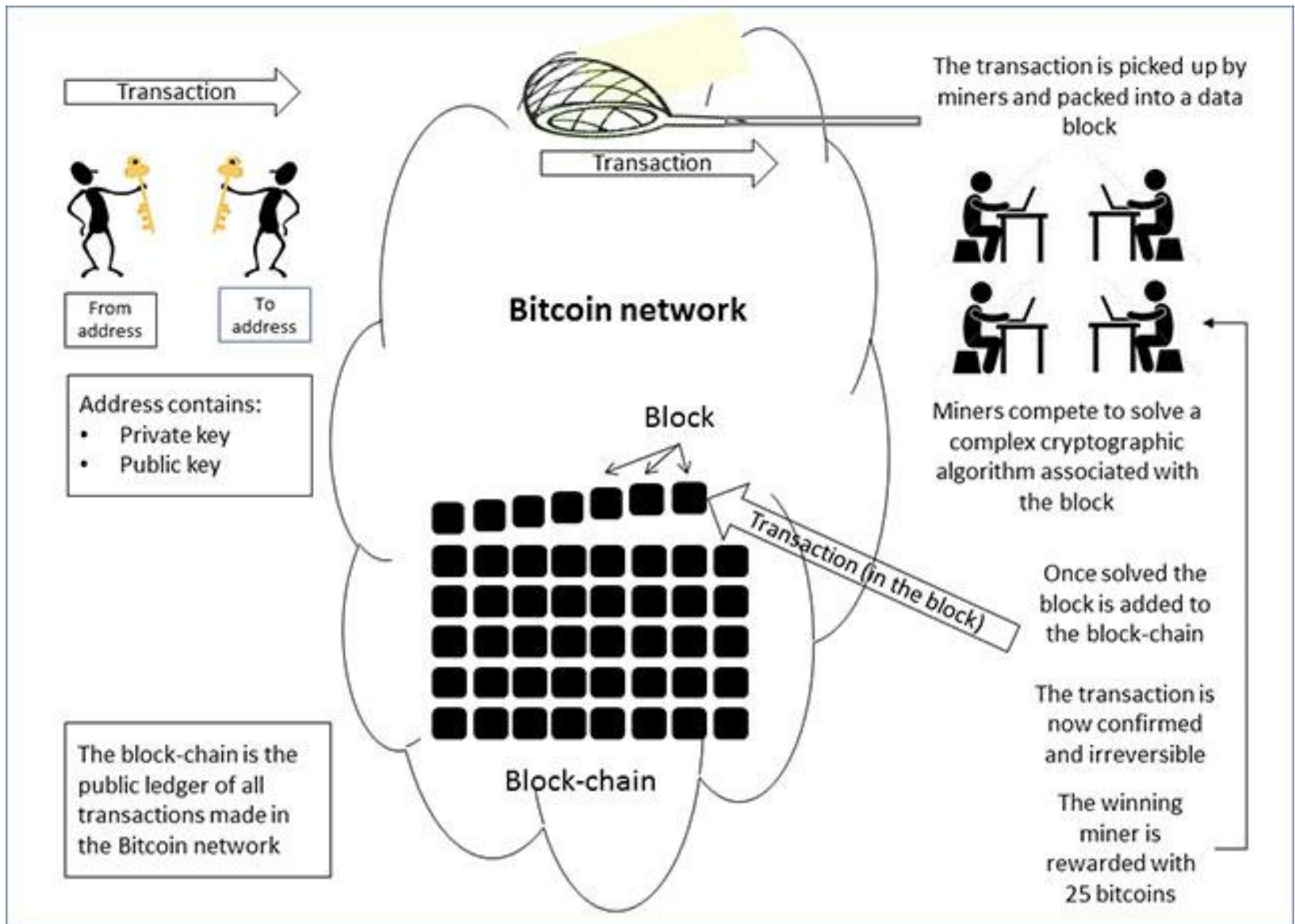


Figure 2-4. A chain of transactions, where the output of one transaction is the input of the next transaction

Bitcoin system

- Users –bitcoin clients
- Miners – full nodes, lite nodes
- Exchanges
- Merchants
- Developers
- Coin generation schedule
- Networks
- Cryptography
- Hash functions
- Transactions
- Blocks



Experimental Bitcoin Network

- Testnet –runs the same code as the mainnet, but can be run as an experiment.
- One can change the protocol and runs one's own bitcoin with
 - New free coins
 - Faster block generations time
 - Different Issuance schedule
 - Difficulty

A. *Experimenting With the Bitcoin Network/Protocol*

Studying the Bitcoin network and the interplay of nodes poses a challenge. By now, there are a few possibilities to approach to this task. One way is to connect to the mainnet, i.e., the live Bitcoin network, or the testnet [25]. The testnet is a global playground to experiment with the Bitcoin protocol and its scripting capabilities. It uses a separate, distinct block chain, and so-called *faucets* provide coins for free. Apart from a few minor parameter alterations, e.g., for faster block generation, the testnet mimics the mainnet and runs the same code as Bitcoin peers. Both approaches can be used to interact with or observe, usually with a highly connected passive peer, the network.

In contrast, local testing environments provide more control. Built into the Bitcoin reference software is a regression test mode (*regtest*). It can generate blocks on demand and create “private” coins with no real-world value. By doing this, it provides a safe harbor for testing new features. A similar approach is the *bitcoin-testnet-box* (github.com/freewil/bitcoin-testnet-box). Both have been designed for situations where interaction with random peers and blocks is not desired.

Joining and Maintaining the network

- A peer keeps 8 to 125 neighbors.
- When connect, they check for version, time synch and IP.
- Each peer keeps a list of active peers.
- Each peer broadcast its own IP address in an `addr` messages every 24hrs.

B. Joining and Maintaining the Network

Every peer in the Bitcoin network aims to maintain a minimum of eight connections in the overlay. That is, the peer actively tries to establish additional connections if this number is underrun. The number of eight connections can be significantly exceeded if incoming connections are accepted by a Bitcoin peer; usually a network participant does not handle more than 125 connections at a time (`maxconnections`). By default, peers listen on port 8333 for inbound connections. When peers establish a new connection, they perform an application layer handshake, consisting of `version` and `verack` messages. The messages include a timestamp for time synchronization, IP addresses, and the protocol version. Since Bitcoin version 0.7, IPv6 is supported.

In order to detect when peers have left, Bitcoin uses a soft-state approach. If 30 minutes have been passed since messages were last exchanged between neighbors, peers will transmit a heartbeat message to keep the connection alive. If 90 minutes have passed without any incoming message, the client will assume that its counterpart is offline. Bitcoin peers also keep track of not directly connected peers in the network. They maintain a list of recently active peers, including their IP address and a timestamp. Every peer broadcasts its own IP address in an `addr` message every 24 hours through the overlay. The

Peek network by `getaddr`

- A peer can ask neighbors for additional peers by issuing `getaddr`.
- To that, peers reply with `addr` message in which only 23% peers of their own active list.
- 2014, there were more than 872K IP addresses.

Besides unsolicited reception of `addr` messages, peers can ask neighbors for additional peers by sending a `getaddr` message. The response (`addr`) contains a random selection of 23% (but not more than 1,000) peers from the responder's list of recently active peers. (It seems that there are no particular rea-

without additional effort. In fact, it is a design goal of the Bitcoin network implementation to obfuscate the topology and to make sure that (local) attackers cannot fill up a peer's neighbor table with compromised IP addresses. Otherwise it would

In a study from November 2013 to January 2014, the authors of [115] asked a set of initial peers for information about other peers they know, by sending a `getaddr` to these peers. For every previously unknown peer thus discovered, they repeated the procedure and asked them for peers, too. In the first round they already discovered 111,475 IP addresses. After 37 rounds during the 37 days of the study, they discovered 872,648 distinct IP addresses in total. Geolocation lookups revealed that

Transaction and Propagation

C. Transaction and Block Propagation

Based on the unstructured overlay as discussed so far, information about new transactions and blocks is spread to the peers in order to form the distributed consensus. The mechanism is simple: messages are flooded through the network. Let

Figure 7. Assume Alice constructed a valid transaction. Before broadcasting the actual transaction data including all details of inputs and outputs, she sends an inventory (`inv`) message stating “I know about new transactions” to all of her neighbors. This message contains a list of transaction hashes (TXIDs), but not the actual transaction data.

Alice’s neighbors will request data from specific transactions in a separate `getdata` message, if these transactions are so far unknown to them. This pull-based communication mech-

After Alice’s neighbors verified the transaction, they will make it available to all their neighbors in the same manner as Alice did, starting with an `inv` message.

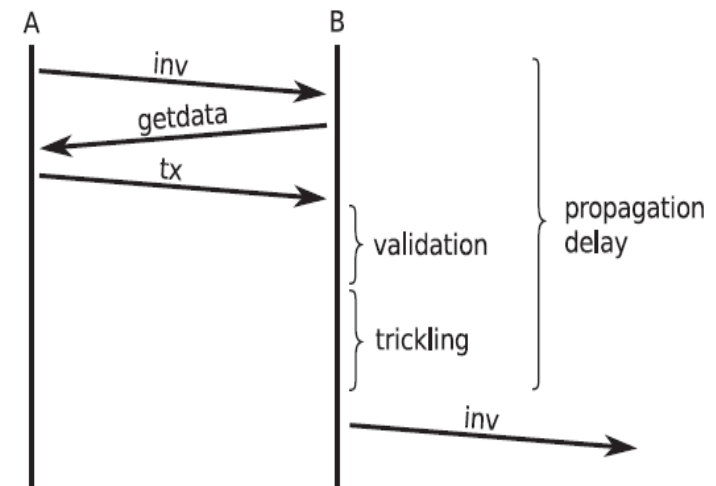
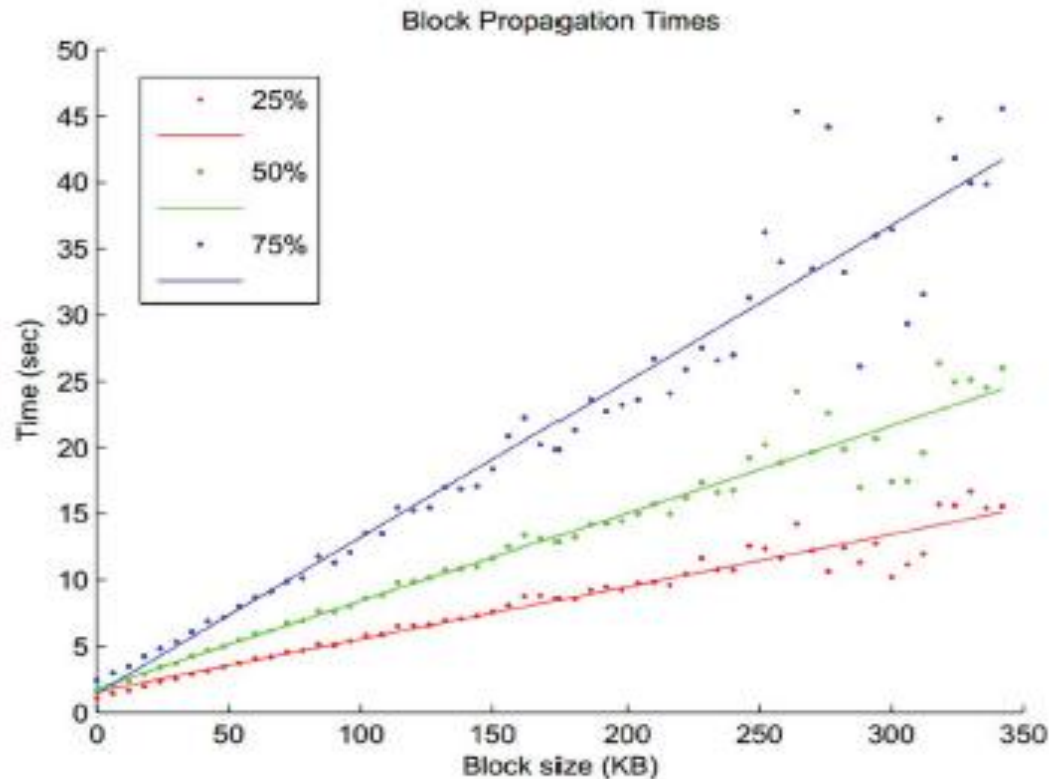


Fig. 7. Transaction propagation.

Block Propagation Time



Source: Yonatan Sompolinsky and Aviv Zohar: "Accelerating Bitcoin's Transaction Processing" 2014

Figure 3.10 Block propagation time. This graph shows the average time that it takes a block to reach various percentages of the nodes in the network.

Rebroadcasting, if necessary

into the block chain. Alice, as the originator of her transaction, is responsible for its distribution. She might hence need to rebroadcast it if the transaction did not get into the block chain, to make sure it gets considered in the next block.

- Alice is the one who needs to keep broadcasting her TX until it has gotten into a block.

Relaying Frequency

- 91% TSs relayed only once
- 6% TXs are relayed multiple times by multiple peers.

By observing the forwarded messages from a highly connected peer, [118] revealed three distinct relay patterns. With about 91% of all observed instances, the most common relay pattern involves lots of peers relaying a transaction only once. Since clients keep track of seen transactions and relay only new ones, this is exactly what one would expect. The second relay pattern involves a transaction received once or multiple times from a single peer. This is not very common (3%); it occurs when invalid transactions are broadcast and hence not relayed. The third relay pattern involves a transaction relayed by multiple peers and re-relayed by at least one of them (6%). The reason behind the occurrence of this pattern is that transaction originators are responsible for their transactions and might need to re-broadcast if they get forgotten.

Bitcoin Attacks

- Race attack
- Sybil attack
- 51% attack
- Double spending attack

TABLE III
BITCOIN ATTACK VECTORS AND VULNERABILITIES

Attack Vector	Description	Sec.
wallets	vulnerable to theft [54], [55]	III-A
key recovery	recovering private keys due to weak randomness [33], [56]	III-A
51% attack	achieves optimal Byzantine resilience, i. e., $2f + 1$ resilience [57]	III-B, V-D
double spending	double spending is and will always be possible [58]	III-B
block withholding	used for double spending [59], [60] and for selfish mining [61]	III-B, III-D
transaction malleability	altered TXIDs [62] to make sb. believe transactions have been failed [63]	III-C
timejacking	used to isolate peers [64] and to drift mining difficulties [65]	III-D
netsplit	facilitates double spends with more than one confirmation [66]	IV-B
scalability	depends on propagation delays [67] and fork resolving strategies [68]	IV-D
centralization	weakens the resilience of the currency [69], [70]	IV-D, V-E
DoS	peer blacklisting can be used to mount denial of service attacks [71], [72]	IV-E
transaction history	block chain analysis might reveal trade relationships [73]–[75]	V-B

Conclusion

- **Many Possibilities of Blockchain**
- **Verified by the market are Bitcoin and Ethereum**
- **To explore new territory, experiments are needed with budgets and man power invested.**
- **Needed are the research on regulation as regulations should be kept at the minimal level and more emphasis shall be on cultivation of new ideas.**
- **Obvious regulations should be in place right away**
 - Responsible investment culture
 - Improved clarity on exchange business and initial coin offerings
 - Price manipulation practice
 - Taxation on profits
 - Use of real name in cryptocurrency transaction? Security becomes problem.

References

- Shweta Lambda and Monika Sharma, "An Efficient Elliptic Curve Digital Signature Algorithm (ECDSA)," 2013 Int. Conf. Machine Intelligence and Research Advancement.

HW#2

- Problem 1:
- Name three parts in the bitcoin protocol that is protected by the hash function. Name one part if any that is not protected by the cryptography.
- Who chooses bitcoin transactions to be included in a block?
- Who is responsible for making a transaction included in a block?
- How does one make sure that a transaction is included in the block chain?

Double Spend Race Attack

A announces a TX showing A sends B 1 BTC at the end of time t_0 .

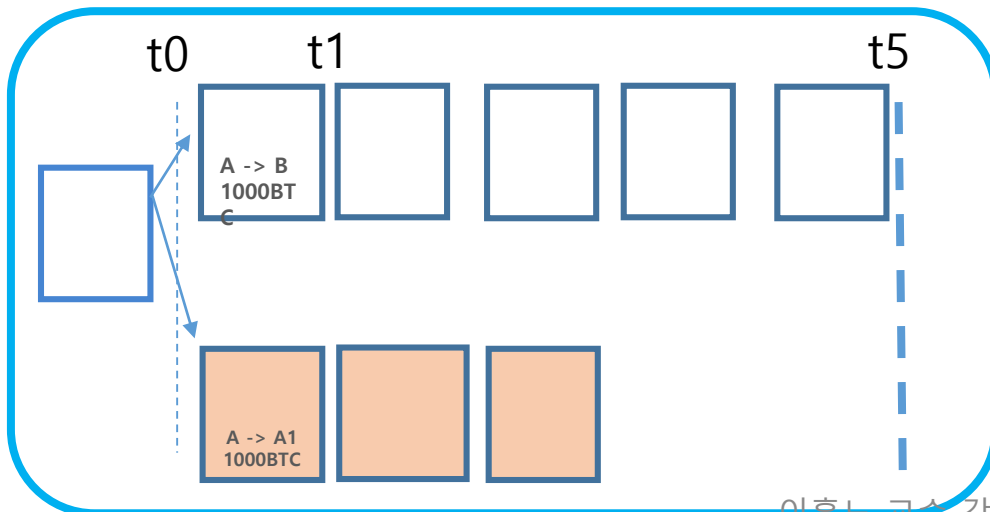
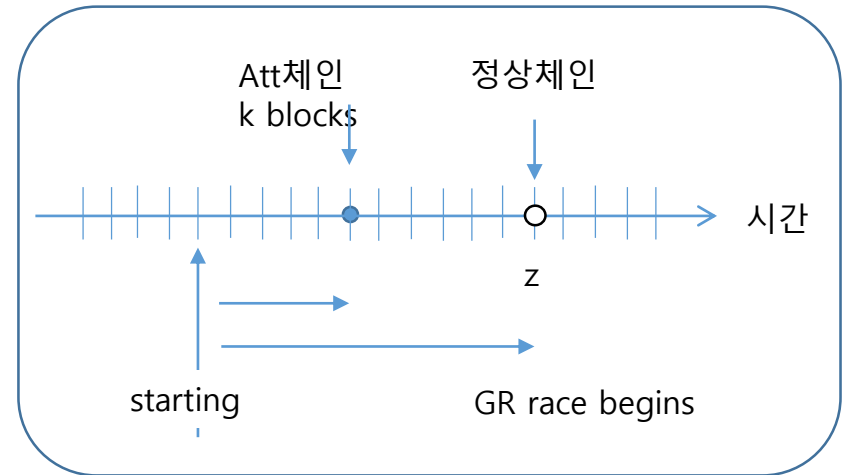
This TX gets into a block (1 confirmation) at t_1 .

B waits until he gets the 5th confirmation at t_5 .

A is the attacker.

A starts preparing a double spend attack at t_0 . Namely, A grows its own chain. In his chain, he has replaced the TX A->B 1000BTC with a TX, A -> A1 1000BTC. A1 is another public key of A.

At t_5 , A has mined 3 blocks and needs to decide if he continues to grow his own chain or not.



Attack Success Probability(q, z)

$$= \sum_{k=0}^{\infty} \begin{cases} (q/p)^{z+1-k} & k < z \\ 1 & k \geq z \end{cases} \text{Poisson}(\lambda = zq/p)$$

λ is the average number of blocks that the attacker mines in z unit of time

$$= \sum_{k=0}^{\infty} \begin{cases} (q/p)^{z+1-k} & k < z \\ 1 & k \geq z \end{cases} \frac{(zq/p)^k e^{-zq/p}}{k!}$$

HW#2 Problem 2

- We assume for this problem that the attack success probability $p_a(q, z)$ of the bitcoin white paper as given previous page is correct.
 - Let $p+q = 1$, where q = probability the attacker finds the next block and p = probability an honest node finds the next block.
 - Let assume Gambler's ruin is valid.
 - The recipient waits z blocks.
 - Note that I made a fine adjustment, i.e. the red colored **+1**. This is done to reflect that the attack is in fact successful only when the attacker's chain is at least one block longer than the honest chain; it is not successful yet when the chains are at the same length.
 - Note that the amount at stake M_s is high 1000 BTC.

- Evaluate $p_a(q, z)$ for different q and z . Let z varied for $z = 1, 2, 3, 4, 5, 6$. Let q varied $q = 0.05, 0.1, 0.15, 0.2$. Draw all the results into a single figure for comparison purpose and to see the trend.
- Given the result above, let us evaluate the attacker's economic gain for his attack decision. Assume the attacker purchases Ebit E10 ASICs. Use the HW#1's published hashrate.
 - How much money he has to invest to have the hash rate of 10 percent, i.e., $q = 0.1$?
 - How much money he expects to gain from launching the race attack when ($z = 5, q = 0.1$) (let us ignore the electric bill)?
 - Repeat the calculation for different q , $q = 0.05, 0.1$ and 0.2 .
 - How much money he expects to make by simply joining as an honest node.
 - Given the calculations above, which is more rational? Being attacker or being honest.

HW#2 Problem 3

- What is the average size of a transaction in bitcoin network?
- What is the limit of a block in bitcoin network?
- What is the average rate a block is mined?
- Obtain the service rate R_s which is the number of transactions the bitcoin network can execute per second.
- Let the request rate R_q be the number of transactions requested for users in the bitcoin network make per second.
- What occurs to the transactions requested when $R_s \geq R_q$?
- What occurs when $R_s < R_q$?

HW#2 Problem 3

- How long does it take on the average for a block to reach 50% of the nodes in the bitcoin network?
- When Alice the originator of a transaction needs to rebroadcast her transaction?
- What is the percentage of transaction announcements never relayed more than once?
- How many connections are managed by a peer in the bitcoin network? Give the minimum and the maximum number of connections.
- On what port the peers are listening to their neighbors?
- Who issues an `addr` message? What is included in this message? How frequently is it sent?
- Who issues an `getaddr` message? What purpose does it serve? What is the difference with `addr`?

Exercises on *Transaction*

- Name all the fields included in a transaction?
- What are the two fields in a transaction used as pointers in later transactions?
- Is the size of a transaction fixed? If not, why is it vary?
- What is UTXO? What is the command that can be used to find all UTXOs?
- How to create the TXID of a transaction?

Difficulty at Block #500000

- What is the published hash rate at blockchain.info?
<https://blockchain.info/ko/charts/hash-rate>
- **What was the difficulty at Block #500000?**
- **From the difficulty, calculate the target? How many leading zero bits does it require?**
- **How many hashes does it take to mine a block on the average?**
- If you had used Ebit E10 ASIC, how long does it take on the average to mine a single block? Give your answers in terms of minutes, days and years.
- To mine a block in 10 min on the average, how many Ebit E10 ASICs do you need to have?

HW#2

- What is short selling? What good does it offer to a market? In what market, a short seller would make profit from short selling?
- What is Kimchi premium in Dec. 2017? What caused it?