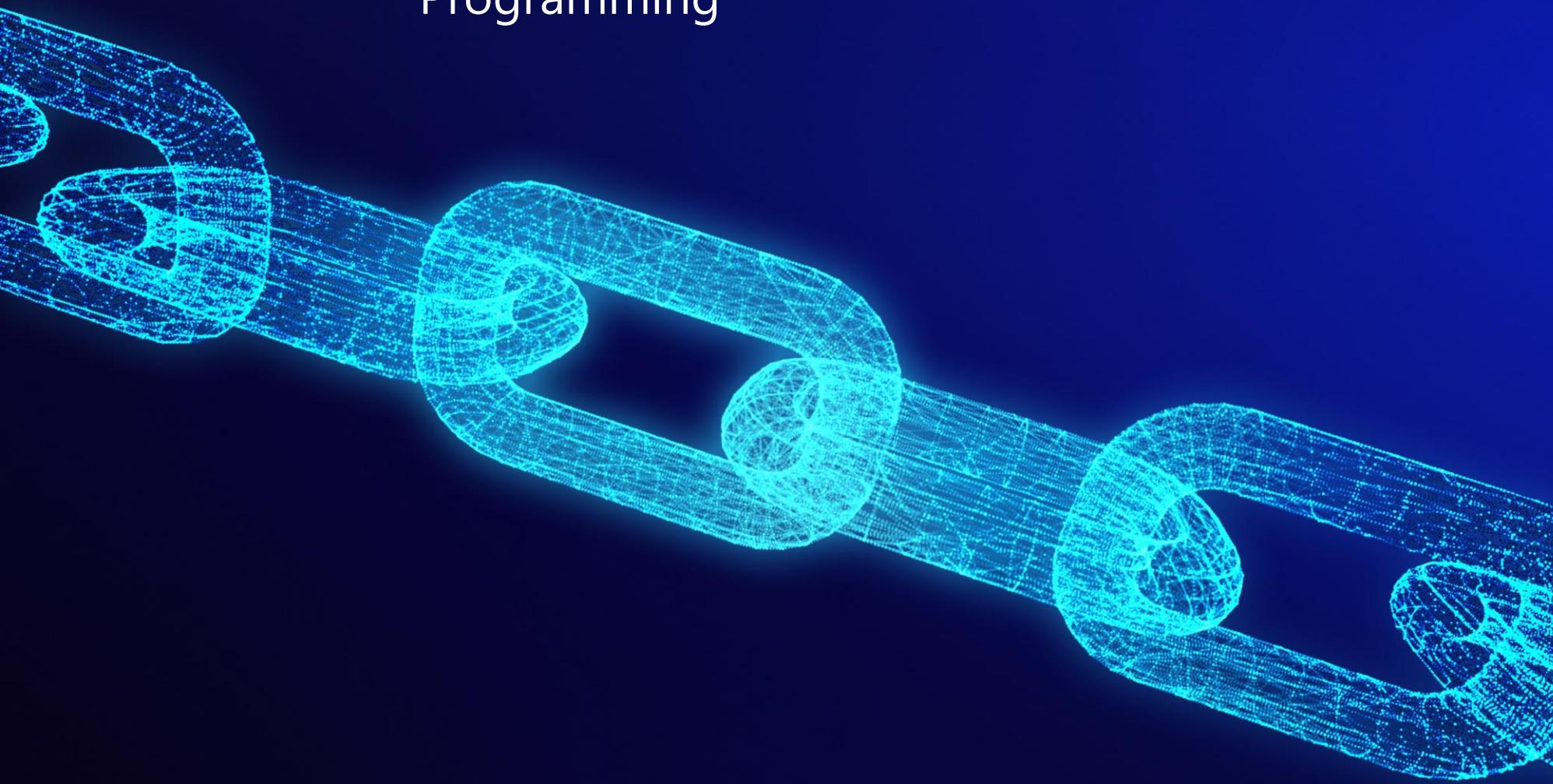


Fundamentals of Blockchain

Programming



Develop a passion for learning.

© 2018 Innovation in Software (2018)

EMERGING TECHNOLOGY



Blockchain has introduced the next generation of web applications.

What is Blockchain? The answer to that question is the source of considerable confusion. Blockchain is much more than Bitcoin, which is a cryptocurrency. With recent financial speculation on Bitcoin, interest in cryptocurrency has gone viral. Blockchain is part of the Bitcoin technology stack. There are a variety of implementations of Blockchain other than cryptocurrencies, such as Ethereum and Hyperledger.

WHAT IS BLOCKCHAIN?



The implementation of Bitcoin is a mystery to many; even those actively speculating on the cryptocurrency. One reason is there is nothing tangible – *no coins*. If Bitcoin is a mystery, Blockchain is a true enigma. Misinformation on the web adds to the confusion. Here is a simple explanation: Blockchain is a decentralized web-based application that hosts a distributed ledger. Imagine a single accounting spreadsheet managed simultaneously by multiple Microsoft Excel web servers.

IMPLEMENTATIONS



Blockchain is a once in a generation disruptive technology based on a simple concept: distributed ledger. For technology, once in a generation means about every two years. Bitcoin is the best example and presently disrupting financial markets worldwide. Ethereum and smart contracts are another example and has the potential to disrupt a variety of industries, including legal, healthcare, election voting, smart locks, and distributed autonomous organizations (DAO).

POP QUIZ: WHAT IS THE ANSWER?



10 MINUTES



What is a distributed autonomous organization and the benefits?

Cryptocurrency Before Bitcoin?



The last two years, our dependence on cash has lessened considerably. The majority of our interaction with money is digital (credit cards, ACH, debit cards, etc.) and may involve online commerce.

For most of us, our checking account is a bank statement (ledger) that is received periodically. We rarely if ever touch physical money. Even our bank statement is emailed to us.

How is that different from cryptocurrency?

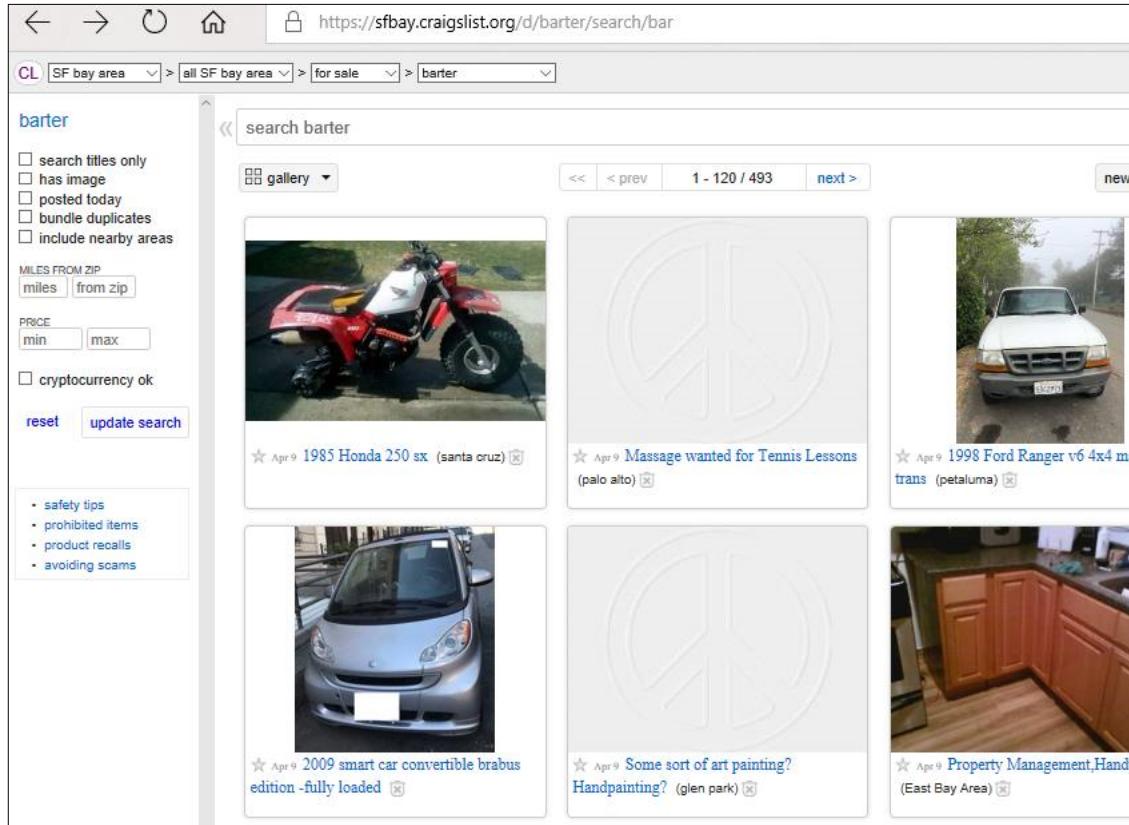
PAYPAL



Do you have a Paypal account? For many, this is their first experience with something similar to a cryptocurrency.

- All transactions are digital
- People can send you money
- You can send other money on the Paypal network
- *There is nothing physical*

NOT ALL ABOUT COINS



Blockchain has a much broader mission than exchanging monetary value. The distributed ledger of a blockchain can track *anything* of value. The only prerequisite is an ability to measure and then convey that value.

For example, craigslist.org has a barter section where people can exchange items of value without the transfer of cash. This is similar to a blockchain except there is a central, even though benevolent, authority.

LOGISTICS



Class Hours:

- Start time is 9am
- End time is 4:30pm
- Class times may vary slightly for specific classes
- Breaks mid-morning and afternoon (10 minutes)



Lunch:

- Lunch is 11:45am to 1pm
- Yes, 1 hour and 15 minutes
- Extra time for email, phone calls, or simply a walk.



Telecommunication:

- Turn off or set electronic devices to vibrate
- Reading or attending to devices can be distracting to other students
- Try to delay until breaks or after class

Miscellaneous

- Courseware
- Bathroom
- Fire drills

DONIS MARSHALL

**Blockchain Developer
Microsoft MVP
Microsoft Certified
Python, Java, Go,
and C++ developer
Author**

donis@innovationinsoftware.com



HISTORY LESSON

A secure, safe, distributed, and decentralized ledger was first described in the Journal of Cryptography 1991 written by Stuart Haber and W. Scott Stornetta. Here is the link.

<https://link.springer.com/article/10.1007/BF00196791>

This article describes a chain of secured blocks that form a ledger where value is transferred using transactions. The most known implementation of blockchain is Bitcoin.

The first person(s) to describe the concept of an actual blockchain was Satoshi Nakamoto in 2008. More about Satoshi Nakamoto later.

Blockchain 2.0 allows for the transfer of more than money with static transactions. This new methodology also supported programmable transactions which is the essence of smart contracts. The most recognize implementation of Blockchain 2.0 is Ethereum.

INTRODUCTION TO BLOCKCHAIN

Blockchain provides a technical framework for decentralized solutions.

Historically, centralized solutions have been more common than decentralized. The best example is banking where the bank manages the financial ledger of customers. The bank is the trusted authority and the clearing house of all transactions. But also a single-point of attack for assailants. For services rendered, the trusted authority charges fees for their services.

A non centralized solution removes the trusted authority in lieu of a peer to peer relationship. The result is a trustless network. The advantages include improved security, lower costs, scalability, and additional processing power. Yes, improved security in a trustless environment!

BLOCKCHAIN IMPLEMENTATIONS

Blockchain is in its relative infancy as a technology. Despite this, various technical solutions have already emerged. This is a short list.

- Ethereum
- Whisper
- Swarm
- Filecoin
- Hyperledger Fabric
- Bitcoin
- Litecoin

The complete list is sooooo much longer. There has been an explosion of interest in Blockchain recently.

ETHEREUM



Ethereum is a platform for running decentralized applications (Dapps) across nodes via smart contracts. Smart contracts are a misnomer; it is programmable code that is run as the target of a transaction. Transactions, powering Dapps, can be sent between users and contracts. They can also be sent between contracts.

Ethereum is a general purpose blockchain with an established ecosystem.

WHISPER



Whisper is a distributed messaging protocol that allows Decentralized Applications (Dapps) on the same blockchain to message each other. Consider this to be RPC for Dapps. The messages are temporary. Messages are topic based where nodes register topics of interest while filtering the remainder. Unicast, multicast, and broadcast of messages are allowed.

Whisper are used with dapps that collaborate on transactions.

SWARM



Swarm is a peer to peer (P2P) platform for distributed storage and replication where nodes can contribute their resources to the blockchain. This approach creates almost unlimited scaling. It is also used as distributed storage for Dapps.

This is an incentive based system where usage patterns affect availability.

Swarm is part of the Ethereum stack.

FILECOIN



Filecoin is a platform for decentralized storage versus distributed storage, such as Google Drive. It is an excellent tool to leverage unused storage accumulating on machines around the world.

Filecoin uses the InterPlanetary File System (IPFS), which is peer-to-peer hypermedia protocol, to reliably and securely transfer files. Each file is identified with a unique hash. The unique hash is used for indexing.

This is a compensation system where renumeration is done in Filecoin tokens (FIL).

HYPERLEDGER FABRIC



Hyperledger Fabric is a permissioned blockchain supporting global cross-industry transactions via smart contracts. Members of the Hyperledger Fabric network register through a Membership Service Provider (MSP).

Hyperledger Fabric is more extensible than other blockchain implementations. Instead of adopting a single universal standard there is support for pluggable implementations of different components.

Disparate ledgers can be created using channels. Groups of participants can create channels as partitions from other groups. Each channel can have separate rules that apply to particular participants.

BITCOIN



Bitcoin is a cryptocurrency and probably the most popular implementation of blockchain and has become a worldwide phenomenon. Ultimately, Bitcoin is a decentralized payment system not requiring centralized control from a traditional financial institution. Bitcoin is a public blockchain orchestrated by a series of nodes.

Bitcoin protocol was defined by Satoshi Nakamoto, who is an anonymous person or persons. You can download the protocol here:

<https://bitcoin.org/bitcoin.pdf>

BITCOIN 2



Bitcoin is a:

- Bitcoin protocol
- Decentralized peer to peer network
- Public transactional ledger
- Consensus network
- Set of rules and guidelines

BITCOIN - 3



Bitcoin is:

- Collaborative
- Secure
- Borderless

Bitcoins are created via a process called mining. Miners must solve a complex mathematical problem to process a Bitcoin transaction.

CODE EXAMPLES IN THIS COURSE

Example with prefix	Type
aulColors	array of unsigned long
bBusy	boolean
chInitial	char
dwLightYears	double word
nSize	integer
dbPi	double
pFoo	pointer
pszOwner	pointer to zero-terminated string
szLastName	zero-terminated string

Either actual or pseudo code is presented for the source in this course.

If the implementation is straightforward, then the exact code is presented. When the solution is discretionary, pseudo code is presented.

Hungarian notation is used for the pseudo code, where the prefix indicates the type. When ambiguous or not clear, comments are provided.

Lab 1- Cryptocurrency



Cryptocurrency

Blockchain is not a passing fad. New blockchain projects are being funded virtually everyday. There is an enormous amount of interest and investment in this sector of technology.

Probably the best example is the ecosystem that has sprouted up around cryptocurrency. Websites, APIs, SDK, analysts, and much more now exist to serve this segment and increasing exponentially.

Explore the websites presented on the next few pages. Even if you are not interested in cryptocurrencies, it will give you some understanding of the potential of blockchain solutions.

This is just a sampling of websites. There many more.

The most detail market analysis of cryptocurrencies. Click the links for each currency for more specific information.

The screenshot shows the OnchainFX.com homepage with the following key elements:

- Header:** "OCFX Cryptoasset rankings & metrics".
- Top Marketcap:** Total Y2050 Marketcap: \$386,607,245,947 and Total Current Marketcap: \$244,567,449,425.
- Bitcoin Dominance:** 49.01%.
- Table:** A main table listing the top 22 cryptocurrencies by marketcap. Columns include: #, Flag, Name, Price USD, 24hr Change vs USD, Y2050 Marketcap (implied), Current Marketcap, 24hr Vol, and Supply % Issued.
- Filters and Options:**
 - Flagged Assets:** Shows you haven't flagged any assets yet.
 - Recent Quotes:** Shows you haven't viewed any assets yet.
 - Daily Movers:** Sub-sections for Top Gainers (LSK, MAID, ZRX, STEEM, IOT) and Top Losers (BNB, OMG, VEN).
 - Cryptoasset Indexes:** Bletchley 10, 20, 40.
 - Sector Watch:** Daily Winners (Content Creation, Timestamping Services, Internet of Things) and Daily Losers (Exchange Platform).
 - Relative Sizes:** Shows relative sizes by Current Marketcap.
 - Search:** A search bar.
 - Choose Columns:** A dropdown menu for selecting columns, including checkboxes for Flag, Logo, Price USD, Price BTC, 24hr Change vs USD, 24hr Change vs BTC, Y2050 Marketcap (implied), Current Marketcap, 24hr Trade Vol, and Age.
 - Supply:** A dropdown menu for supply-related metrics.
 - All Time High:** A dropdown menu for historical performance metrics.
 - On-chain Data:** A dropdown menu for on-chain transaction metrics.

BLOCKCHAIN.INFO

BLOCKCHAIN.INFO is one of the most popular blockchain websites. Explore the options under the DATA menu. We will be using the search option in class.

BLOCKCHAIN [WALLET](#) [DATA](#) [API](#) [ABOUT](#)

Q BLOCK, HASH, TRANSACTION, ETC... [GET A FREE WALLET](#)

LATEST BLOCKS [SEE MORE →](#)

Height	Age	Transactions	Total Sent	Relayed By	Size (kB)	Weight (kWU)
517496	7 minutes	407	1,672.44 BTC	BTC.com	1,023.94	3,827.26
517495	9 minutes	519	2,065.02 BTC	AntPool	999.79	3,899.49
517494	15 minutes	164	212.86 BTC	BTC.TOP	969.09	3,827.81
517493	15 minutes	1294	5,844.05 BTC	AntPool	585.58	2,076.66

NEW TO DIGITAL CURRENCIES?
Like paper money and gold before it, bitcoin and ether allow parties to exchange value. Unlike their predecessors, they are digital and decentralized. For the first time in history, people can exchange value without intermediaries which translates to greater control of funds and lower fees.

[BUY BITCOIN →](#) [LEARN MORE →](#) [GET A FREE WALLET →](#)

SEARCH
You may enter a block height, address, block hash, transaction hash, hash160, or ipv4 address...

Address / ip / SHA hash [Search](#)

TRANSACTIONS PER DAY
The number of bitcoin transactions in the last 24 hours.

1 8 0 8 7 8

Transactions since Sun Apr 08 2018 11:09:52 PM.

1 BTC = \$6,770.52
[Interactive Chart →](#)



BLOCKEXPLORER.COM

BlockExplorer.com is another popular blockchain website. It provides market analysis of cryptocurrencies, technical detail of the Bitcoin blockchain, videos, and more.

Block Explorer News Market Bitcoin cash Zcash Blocks Status [Buy Bitcoin with CC!](#)

Search for block, transaction or address ✓ Conn 72 · Height 517502 Scan BTC

Latest Blocks

Height	Age	Transactions	Mined by	Size
517498	17 minutes ago	348		127360
517497	20 minutes ago	851		708452
517496	28 minutes ago	407		934523
517495	31 minutes ago	519	AntMiner	966652
517494	36 minutes ago	164		952991

[See all blocks](#)

Latest Transactions

Hash	Value Out
448d8b436f1d94ab23e09b9c60305fd97310161224b...	0.199987 BTC
4113213ef503874d882114c14dc909960f99e5fede...	1.49477796 BTC
ce76eb4ee4b76f1c181f1deaf88c78e28b73a18926c5...	16.1862286 BTC
7219063ce32ad0ddcd7164f0a578ba57763e82113398...	74.95957728 BTC
b1091063dd9065d7d46536d967061ca116f8649496...	0.01325526 BTC
b96b208379a2c8e3da87a2e7b76a62729574ea17b9...	2.16450234 BTC
41ce81b3d83587e4006d3870d7c79118b35ef4985a...	0.33261194 BTC
776c34e3444bd273dc8909fdec177831ef6150ace2...	0.00319606 BTC
f6a00dbeaf777957879eb3e1d496e3cb7c6ed96dc56...	0.664538 BTC
95c163aa6c2c8337e3cea1355ec9737110be96f214e...	0.00498106 BTC

Block Explorer Videos

- 'Store Of Value'
- Rick Reacts: "Coinbase insider trading?" - This story was planted!
- Rick Reacts: Network Effect of Bitcoin Legacy (BTC) is precisely zero.
- Rick Falkvinge, Christel Dahlkvær, and Arne Schwabe of

Block Explorer News

- Ethereum Mulls Hard Fork to Maintain ASIC Resistance
- Video Game Giant Unity to Develop SDK For Kik's ICO Token
- American Voters To Use Blockchain-Based Mobile App

BLOCKTRAIL.COM

BLOCKTRAIL.COM provides market analysis on cryptocurrencies. It has a straightforward GUI. Open the developer link to view programming tools, such as an API and SDK, that help simplify blockchain development.

The screenshot shows the homepage of Blocktrail.com. At the top, there is a green header bar with the Blocktrail logo (a stylized 'B' inside a circle), navigation links for 'Wallet', 'Explorer', and 'Developer', and user account links for 'Sign Up' and 'Login'. A yellow banner below the header states 'BLOCKTRAIL has been acquired by BITMAIN' with a link to 'Read The Announcement'. The main content area features a large photograph of a man and a woman sitting at a desk, looking at a laptop screen together. Overlaid on the photo is the text 'A Powerful Bitcoin Wallet' and a subtext 'Blocktrail is a Bitcoin wallet that puts you in full control of your Bitcoin'. Below this are buttons for 'Create Wallet', 'Get it on Android', and 'Get it on iOS'. In the bottom left corner of the main content area, there is a small inset showing screenshots of the Blocktrail mobile app interface on a tablet and smartphone. To the right of this inset, the text 'The Smartest Way To Own Bitcoin' is displayed, followed by the subtext 'Bitcoin empowers you to send and receive payments without trust or permission from any third-party.'

BIT\$CREENER.COM

Visit Bit\\$screener.com. It has a great mobile application for phones to monitor the market value of various cryptocurrencies.



BLOCKCHAIN – DEFINED AGAIN

Blockchain is a decentralized application that manages a public distributed ledger of transactions. Each transaction is a transfer of value – not necessarily monetary value. The value could for example be a lien for a vehicle. The blockchain is a secured, immutable, and order list of blocks (i.e., a chain of blocks). Each block contains one or more transactions, a sender and receiver, and an identifier.

There is a peer to peer network without a trusted central authority, the ultimate democracy, where nodes (computers) collaborate to manage the blockchain were decisions are made by consensus. There is not a central authority for authenticating and authorizing transactions.

Each participating node maintains a copy of the blockchain ledger. Since the ledger is replicated across several nodes, there is not a single point of attack, which is one of the many security features of the blockchain.

CENTRAL AUTHORITIES



Many businesses, particularly the financial sector, represent centralization of control in a trusted authority. This is most prevalent in industries or business where customers cannot trust one another, such as banking. The central authority provides a framework for trust, such as regulatory guidelines, the processing of transactions, coordinating activities, and authentication.

However, most trusted authorities require compensation for their services. The customer has the financial burden of supporting the central authority, which is sometimes arbitrary.

Central authority affects affordability, scalability, security, privacy, and policy.

Blockchain is a counterpoint to centralized authorities providing a frictionless solution for the digital economy.

AFFORDABILITY

Central authorities typically expect payment for services. Transaction fees are a common means of reimbursement for services rendered. For example, the transaction fee for a title search when purchasing a home. If the title search could be accomplished without a centralized authority and automated, the fee could be greatly reduced or even eliminated. The collective cost savings would be considerable. This could be done with a blockchain where titles are kept on a public ledger.

Turing Pharmaceutical acquired the drug Daraprim and immediately raised the price 5,000%. Daraprim can be essential for those having a compromised immune system. The stated reason was to fund additional research. Martin Shkreli the President of Turing became rich but achieved an unparalleled level of infamy. Mr. Shkreli became known as the "most hated man in America".

<https://stanford.io/2lxr6Jp>

PRIVACY

Central authorities are tasked with maintaining your privacy, which is different than security. Security is a break-in where a third party steals your personal and sometimes sensitive information. But users also rely on central authorities not to give away or sell personal information. This is a sacred trust placed with central authorities. However, this trust is often broken. With Blockchain, privacy is always maintained through strong cryptographic techniques and inherent to the model.

Facebook shared private information for 87 million loyal customers to Cambridge Analytica, which used the information for political purposes. This was done without the consent of users.

<https://nyti.ms/2HP4Dr3>

POLICY

A central authority can administer policies that affect users. This is a problem especially when policies are counter to the well-being of the user. Furthermore, some policies are subjective leaving room for interpretation and abuse. Most importantly this can lead to censorship of content or free expression. In Blockchain, policies are transparent and not subjective.

Governments often impact the policy of online companies. This means policies can depend on where a website is browsed. For example, the shadow of the Chinese government looms large on technology companies operating online in China. For example, content perceived critical of the Chinese government is often censored. Even the largest internet companies are not immune to this sort of pressure.

<https://nyti.ms/2uRsO6v>

DECENTRALIZED NETWORK

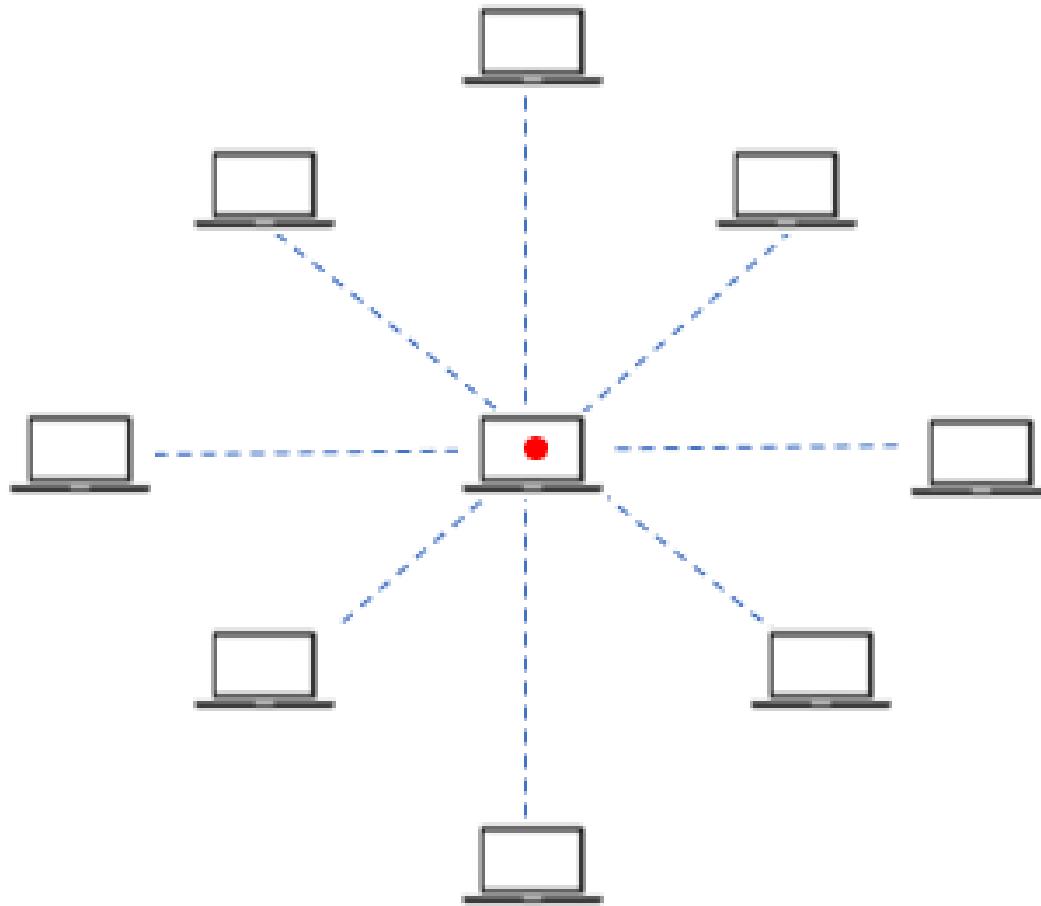
Blockchain is a decentralized network, which is one of the advantages. The three dominant models for network systems: is centralized, decentralized, and distributed.

Centralized: this is the satellite or client / server model. There is one primary computer (server) where work resides, while servicing requests from other computers (client/satellites). The satellites contribute limited or no processing power to the work.

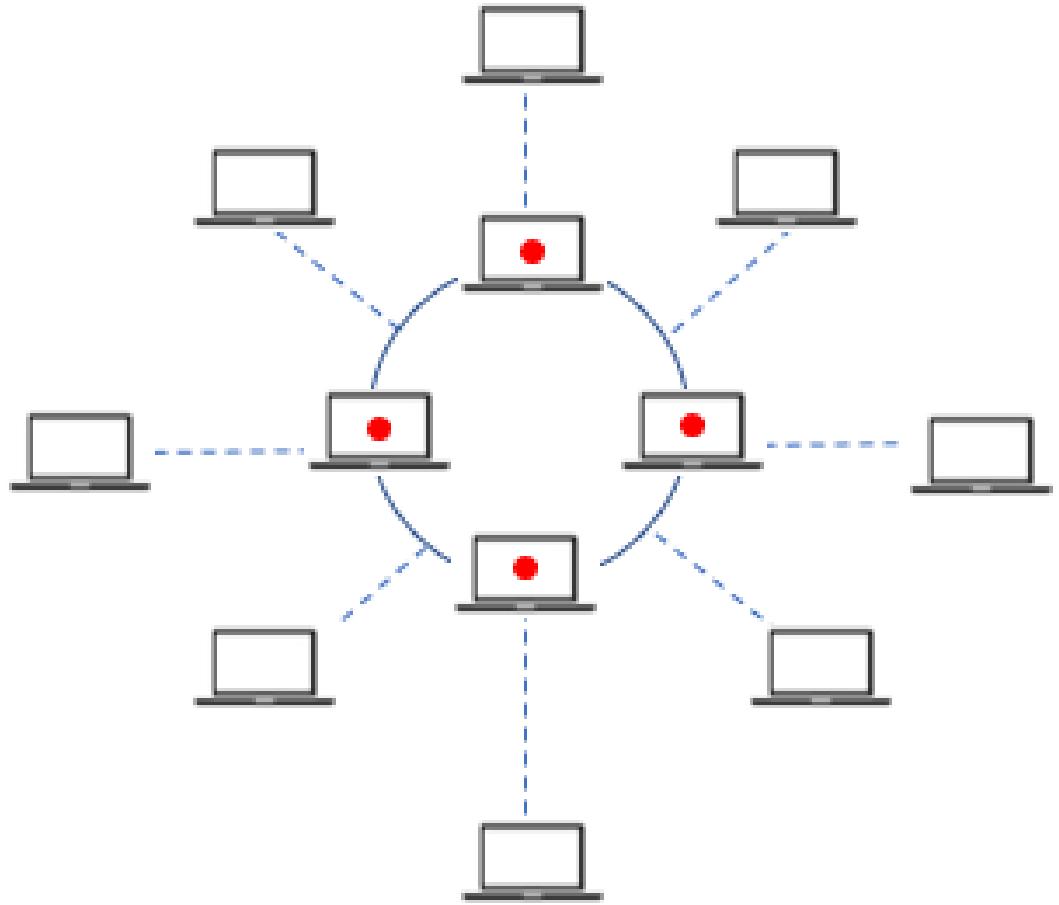
Distributed: this is the collaborative model where various aspects of the work are distributed across several computers. The work is decomposed into tasks which are then assigned to various servers.

Decentralized: there are several computers in a decentralized architecture – each working independently. The application may be running on multiple servers. The computers may communicate and make collective decisions about the state of the network.

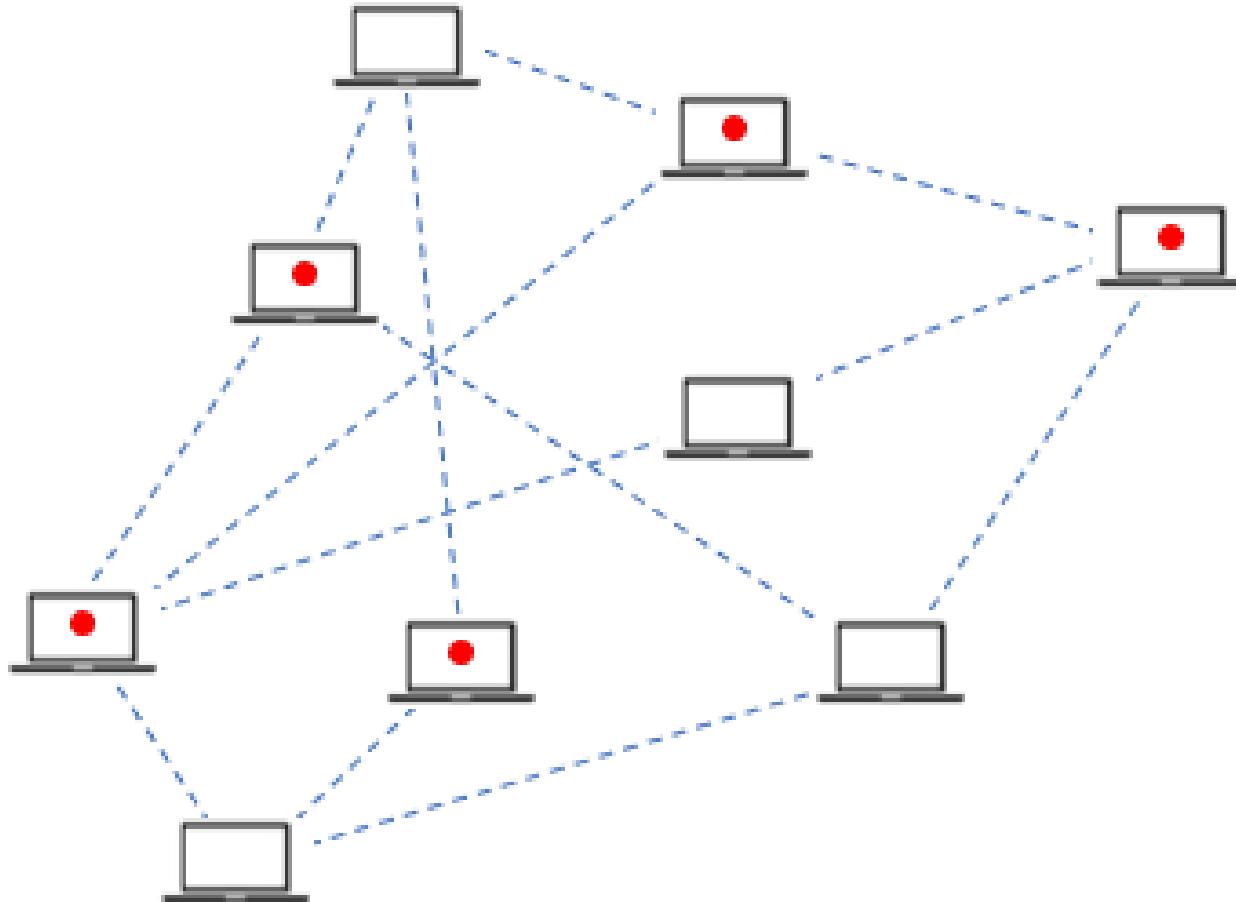
CENTRALIZED NETWORK - DIAGRAM



DISTRIBUTED NETWORK - DIAGRAM



DECENTRALIZED NETWORK - DIAGRAM



PROS AND CONS

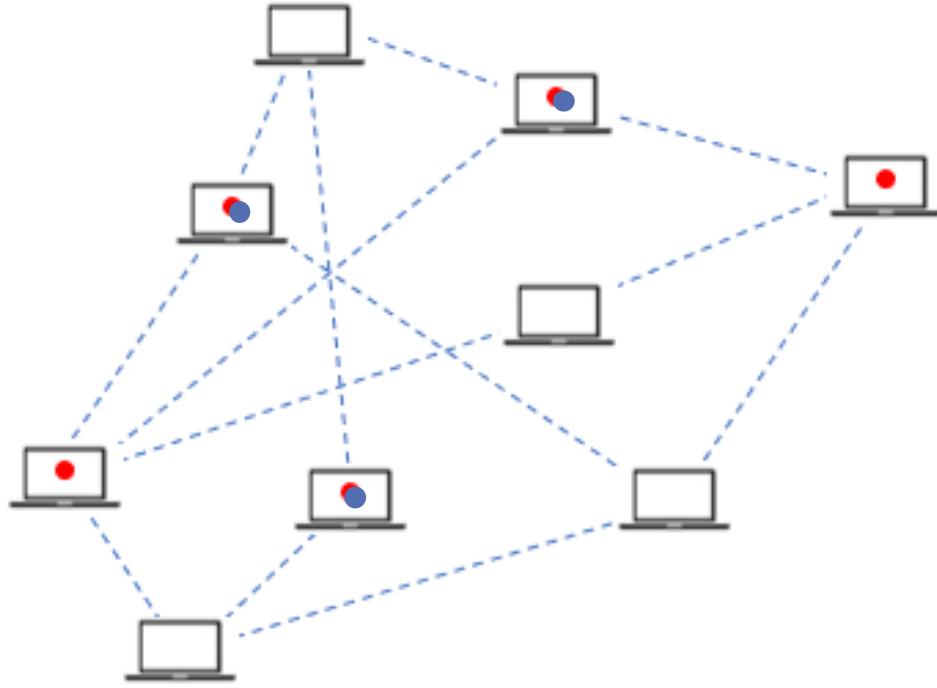
- Points of failure. Centralized systems are easier to manage but have a single point of failure.
- Fault / Tolerance. Centralized systems are highly unstable. Kill the central server to destabilize the entire network. Decentralized are resilient because of the redundancy of effort.
- Scalability. Centralized has limited scalability. Distributed is more scalable. Decentralized is infinitely scalability.
- Development. Centralized is simple and technically easier to develop. Decentralized requires more considerations and more technically complex.
- Security. Centralized is less secure with a single point of attack. Decentralized, especially blockchain, become more secure as the application scales.

PEER TO PEER

In a Peer-to-Peer (P2P) network, there are no master / slave relationship between computers. Responsibility does not reside with a central server, such as a client / server network. In a P2P network, all nodes (computer) share computing responsibility and work together collaboratively. This is the opposite of the client / server model where the main processing power remain with designated servers, which are handling requests for clients.

The blockchain is a P2P network.

DECENTRALIZED APPLICATION (DAPP)



Decentralized applications (Dapps) are when multiple instances of an application run independently on several computers on a P2P blockchain network.

More about Dapps in the module on Ethereum and smart contracts.

- Dapp instance
- Server node

ATTRIBUTES OF A BLOCKCHAIN

Blockchain is a protocol for a decentralized application hosting a distributed ledger. Blockchains that follow that protocol share similar attributes.

- No central authority
- Immutable
- Secure
- Transparent
- Privacy
- Scalable
- Available

NO CENTRAL AUTHORITY



Blockchain is a decentralized network of nodes that have various nodes, such as a miner, a full node, or simple payment verification (SPV). They work together and eliminate the need of a central authority. This includes sharing the responsibility for making the network both secure, immutable, and forming a consensus.

IMMUTABLE



The Blockchain manages an immutable chain of blocks and transactions. Examining the blockchain will expose all of the transactions since the blockchain was created. The blockchain is extendable. You can add but not change or delete blocks. A malicious node that changes the blockchain would be discovered by the other nodes with a copy of the same ledger. They will form a consensus to reject the change.

CONSENSUS



Speaking of consensus...

In the Blockchain architecture, there is no central authority trusted to make decisions on behalf of the network. Blockchain is a trustless network without a central authority. Instead decisions are made by consensus by the nodes. The majority of nodes must agree upon accepting a new block into the blockchain, which contains transactions.

SECURE



The Blockchain protocol assures the chain is securable.

- Each block is hashed – including the hash of the previous block.
- Blockchains are immutable which is verified by consensus.
- Transactions are signed
- Multiple copies of the valid block chain are kept. Attacks require an enormous amount of computational capability.

TRANSPARENT



Blockchains are public and the transactions are available for inspection. For example, anyone can inspect every bitcoin transaction on a blockchain since inception. Transparency contributes to a secure environment.

PRIVACY



The Blockchain protocol supports transparency without compromising privacy. Bitcoin is an example. Every transaction is visible to inspection. You know where money is sent and the amount of the transaction. However, the sender and receiver are abstracted. In addition, private keys are held in a secure location and not publicly available.

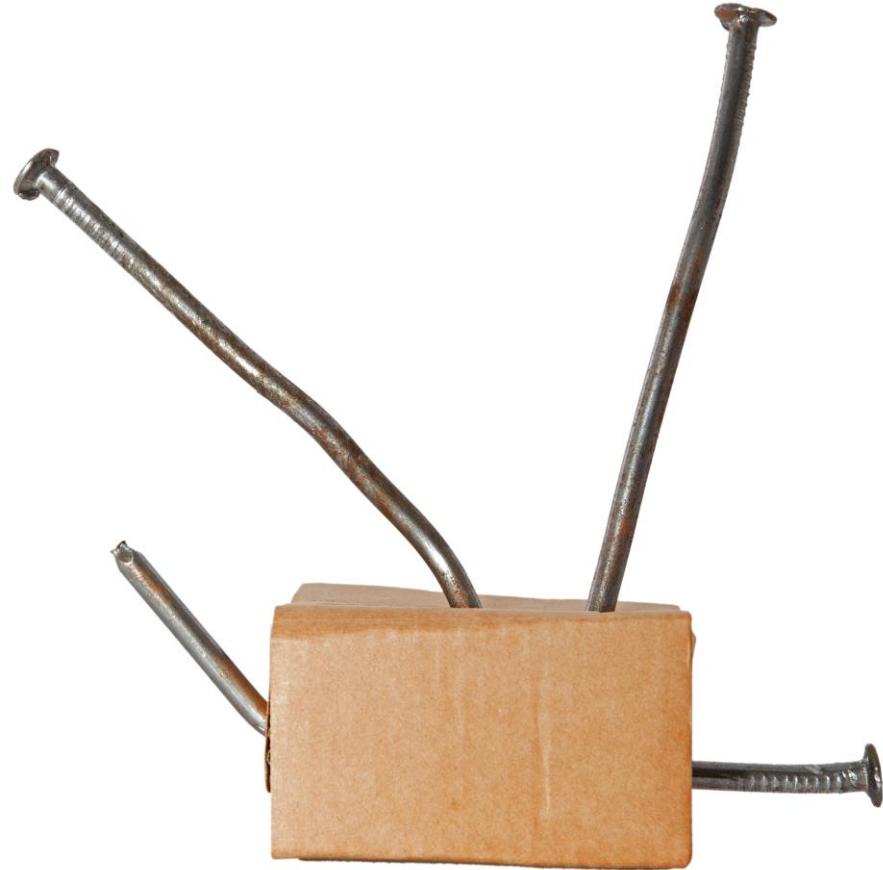
SCALABLE



Blockchains are scalable; just add additional nodes. However, running nodes cost money: computers, electricity, networking, and so on. Nodes receive incentives for their effort, which are periodically adjusted. Incentives can be increased to attract additional miners and full nodes as needed. This means that theoretically blockchains can scale indefinitely. It is simply a measure of compensation.

Of course for Bitcoin, subsidies are paid in coins.

RESILIENT



A blockchain may bend but unlikely to break. As a decentralized network with several nodes hosting the work, the blockchain eliminate dependencies. This is especially true as the various roles are replicated across the internet. Some nodes may be susceptible, from security attacks, internet, or machine failures, but it is unlikely that all of the nodes across a blockchain will fail simultaneously. For this reason, blockchains are incredibly resilient.

AVAILABLE



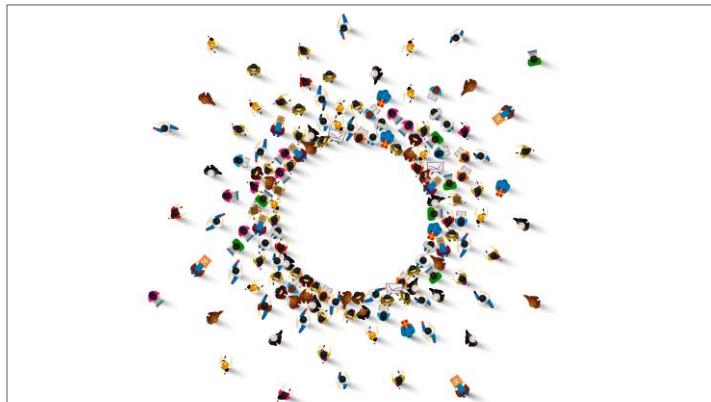
The blockchain is a web-based application. Like most web applications, a blockchain is open for business 24/7 – always available.

PUBLIC VERSUS PRIVATE

Public
Blockchain



Private
Blockchain



Blockchains can be either or public or private. Both are decentralized P2P networks that host a distributed ledger that is replicated across nodes. Public blockchains, such as Bitcoin, are accessible to anyone and highly scalable. Private blockchains are controlled by an entity, such as a company, that limits access to the network. That is the biggest difference between a private and public blockchain – a central authority.

For a private blockchain, the central authority grants access to the blockchain and assigns permissions. For that reason, private blockchains are also referred to as permissioned blockchains.

PUBLIC VERSUS PRIVATE - 2

Private blockchains may have different incentives. For example, miners may not be independent and therefore not require compensation.

Private blockchains can be more secure, such as residing behind a firewall. Everyone, such as the users, miners, and full nodes, could be in the same domain. This means some activities, such as Proof of Work (PoW), could require less work or not needed at all.

Hyperledger Fabric is an example of a permissioned blockchains.

PUBLIC VERSUS PRIVATE - ADVANTAGES

Private blockchain:

- Protected behind a firewall or other security
- More efficient because consensus *may not* be necessary
- More responsive since proof of work *may not* be required
- Not transparent
- Permissioned access

Public blockchain:

- No single point of failure
- Massively scalable
- More resilient
- Transparent
- Available to everyone

Lab 2- Explore a Blockchain



EXPLORE A BLOCKCHAIN - 1

In this lab, you will explore a blockchain using BlockExplorer.com. The goal of this lab is to learn how to navigate a blockchain and explore terminology. Links are provided to facilitate exploration.

Open BlockExplorer.com.

The screenshot shows the homepage of Block Explorer. At the top, there's a yellow header bar with the Block Explorer logo, navigation links for News, Market, Bitcoin cash, Zcash, Blocks, Status, and a pink 'Buy Bitcoin with CC!' button. Below the header is a search bar and a status message: 'Scan 73 - Height 517689' with a QR code and a 'Scan' button. The main content area features a large image of two gold Bitcoin coins. Below the image, a banner reads 'Blockchain Expo World Series 2018 London: April 18-19'. To the right of the banner are 'Recent Posts' and 'Guides' sections with links to news articles. The 'Recent Posts' section includes links to 'Canadian Bank BMO (Bank of Montreal) Restricts Cryptocurrency Transactions', 'Blockchain Expo World Series 2018 London: April 18-19', 'Monero's Scheduled April 2018 Hard Fork Successfully Implemented', 'India Bans Banks Serving Cryptocurrency Exchanges, Locals Fight Back', and 'Future Tech Expo Announces Randi Zuckerberg as Headliner'. The 'Latest Blocks' section displays three recent blocks with details like height, age, transactions, miner, and size. The 'Latest Transactions' section lists several transactions with their hash, value, and output. On the right side, there's a 'Subscribe for updates!' form with an input field for email address and a 'Subscribe' button. Below that is an 'About Block Explorer' section with text about the tool, its GitHub repository, and its API. It also mentions the Testnet and Mainnet blockchains.

EXPLORE A BLOCKCHAIN - 2

Latest Transactions	
Hash	Value Out
3a38fdc3aaddf17f7f5cebc7602...	0.00051397 BTC
cee997aa6f2dbe352c3a27cec0c...	0.22167847 BTC
a3aa8a6f81b2076da625f6263c88...	0.0119633 BTC
412e9b2adacb4df0035da6ac362...	0.0135 BTC
2e110449c433b6ab6bc55cfb197...	8.58961302 BTC
81dd151616d9ed498a786d520cb...	4.95215756 BTC
e2615e4cafe3405eb0c1da1dfe19...	0.10659724 BTC
2a9fac259cb618b74bd2384e809...	0.22851914 BTC
824a3777bdcd23e2a70c5ea7183...	0.04549808 BTC

Find the list of “Latest Transactions”. You can watch additional transactions emerge.

Click on any transaction in the list.

You should see the Transaction window now.

EXPLORE A BLOCKCHAIN - 3

The screenshot shows a web browser displaying the Block Explorer website. The URL in the address bar is <https://blockexplorer.com/tx/736314d34264ccc025323ebc0cf217eb1ca00b47459c8b236c2d9d1a05bbf5c9>. The page title is "Transaction". At the top, there are navigation links for Block Explorer, News, Market, Bitcoin cash, Zcash, Blocks, Status, and a "Buy Bitcoin with CC!" button. Below the title, it says "✓ Conn 73 Height 517669". There are also "Scan" and "BTC" buttons. The main content area has two sections: "Summary" and "Details".
Summary:

Size	698 (bytes)
Fee Rate	0.00021487106017191977 BTC per kB
Received Time	Apr 11, 2018 12:32:48 AM
Mined Time	N/A
Included in Block	Unconfirmed
LockTime	517669

Details:

1Gebx74NK3wfSJtZ2ELPJVA289chCqwrmd	59.06385215 BTC
1FV7ME7m3AQre98LPJ6wC7SxL8Fy9Cc 0.01018168 BTC (U)	
1AXTuVA8QhaPwKkL5D1hjbxr25DLsqtkA6 0.14645 BTC (U)	
19Pb3t3W4LaPX4Lka3bz1VWTvivsk3XA 0.001824 BTC (U)	
1MSY55ZeHQvqQwwPz1wAnkDdLRVAmayGRr 0.00134261 BTC (U)	
18AqUEpERFY5bDvBxLTohVasUPeRkJV8p 0.0021 BTC (U)	
Show more	

At the bottom left, it says "FEE: 0.00014998 BTC". At the bottom right, there are buttons for "UNCONFIRMED TRANSACTION!" and "59.06370217 BTC". A red circle highlights the first output address and its amount.

On the Transaction screen, look at the bottom for transaction details. Notice whether transaction is confirmed or not confirmed. If not confirmed, the transaction has not been mined into a block.

Keep selecting addresses, even switching transactions if necessary, until you find someone with money in their wallet (final balance).

You should be viewing the address page at the moment.

EXPLORE A BLOCKCHAIN - 4

The screenshot shows a Bitcoin Block Explorer interface. At the top, there are tabs for Block Explorer, News, Market, Bitcoin cash, Zcash, Blocks, Status, and a prominent pink button 'Buy Bitcoin with CC!'. Below the tabs, a search bar says 'Search for block, transaction or address' and shows 'Coin 73 - Height 517669'. There are also 'Scan' and 'BTC +' buttons.

Address: 0.34756799 BTC
Address: 3MGAP2FWrDrxRxZ5zSYQjsZLmtotrcn2sd

Summary confirmed:

Total Received	25580.13972395 BTC
Total Sent	25579.79215596 BTC
Final Balance	0.34756799 BTC

No. Transactions: 91016

Unconfirmed:

Unconfirmed Tx Balance	0.00044317 BTC
No. Transactions	5

Transactions:

Two sections of transactions are shown, each with a red box highlighting a specific transaction. The first section has a red circle around the QR code for the final balance of 0.34756799 BTC.

FEE: 0.00008996 BTC UNCONFIRMED TRANSACTION! 0.26755571 BTC

FEE: 0.00008996 BTC UNCONFIRMED TRANSACTION! 0.0197931 BTC

The second section also has a red circle around the QR code for the final balance of 0.34756799 BTC.

FEE: 0.00008996 BTC UNCONFIRMED TRANSACTION! 0.0197931 BTC

Do the math for this user! Does the amount received minus total sent equal the final balance?

Notice the transactions at the bottom. Some are confirmed; others are not.

Select the link for one of the confirmed transactions.

EXPLORE A BLOCKCHAIN - 5

The screenshot shows a block explorer interface for a Bitcoin transaction. At the top, there's a navigation bar with links for Block Explorer, News, Market, Bitcoin cash, Zcash, Blocks, Status, and a prominent pink button that says "Buy Bitcoin with CC!". Below the navigation is a search bar with placeholder text "Search for block, transaction or address" and a dropdown menu showing "✓ - Conn 73 - Height 517669". There are also "Scan" and "BTC" buttons.

The main content area is titled "Transaction" and displays the transaction ID "a5e276f2b834cfffe35bcd9fa9307512886eec2accf8c65d708b7b6709cfbb31". Under the "Summary" section, there are several fields: Size (3541 bytes), Fee Rate (0.000019545326179045467 BTC per kB), Received Time (Apr 7, 2018 6:55:01 AM), Mined Time (Apr 7, 2018 6:55:01 AM), Included in Block (000000000000000000004702d0996df37ed359270178d2ff839bce16f6d36190ed), and LockTime (517016). A red oval highlights the Fee Rate field.

The "Details" section shows the transaction inputs and outputs. It starts with an input from the previous block: "1PENPxz9whkwy5V6PAAvxTh2rn7bgE5h1n" with a value of "655.31079363 BTC". This is followed by five outputs:

Recipient Address	Amount (BTC)
1BLWfhx7rxT9iWLtByTrby3lbV1MKCS42	0.05460907 BTC (U)
16PKwnq7Erg6KP598Hhk4E6ZahsEjNMi49	0.0540245 BTC (S)
164bB6hzVaFvPmqwhjBdI49HKzx8tiojmly	0.05496862 BTC (S)
36FzTWUgvzXh6q2hmnEtPruxJGQvhBjuu	0.05428736 BTC (S)
3BLF1KPJPr6RosRpBoqFkAdB5gcBUUzytC	0.0537733 BTC (S)

Below the outputs, there are buttons for "Show more", "FEE: 0.00006921 BTC", "604 CONFIRMATIONS", and "655.31072442 BTC".

Do the math for this user! Does the amount received minus total sent equal the final balance?

Notice the transactions at the bottom. Some are confirmed; others are not.

Select the link for one of the confirmed transactions.

EXPLORE A BLOCKCHAIN - 5

The screenshot shows a blockchain explorer interface with the following details:

Block Explorer (highlighted) News Market Bitcoin cash Zcash Blocks Status Buy Bitcoin with CC!

Search for block, transaction or address: a5e276f2b834dfffe35bcd9fa9307512886eec2accf8c65d708b7b6709cfbb31

Conn 73 Height 517669 Scan BTC ▾

Transaction

Summary

Size	3541 (bytes)
Fee Rate	0.000019545326179045467 BTC per kB
Received Time	Apr 7, 2018 6:55:01 AM
Mined Time	Apr 7, 2018 6:55:01 AM
Included in Block	00000000000000000000004702d0996df37ed359270178d2ff839bce16f6d36190ed
LockTime	517016

Details

Transaction ID: a5e276f2b834dfffe35bcd9fa9307512886eec2accf8c65d708b7b6709cfbb31
Mined Apr 7, 2018 6:55:01 AM

1PENPxz9whkwy5V6PAAvxTh2m7bgE5h1n	655.31079363 BTC
1BLWfxz7rxT9iwlBtByTrby3ibV1MKC542	0.05460907 BTC (5)
16PKwnq7Erg6KP598HHk4E6ZahsEJNMI49	0.0540245 BTC (5)
164bB6hzVaFvPmqwhjBdl9HKzx8tjojmty	0.05496862 BTC (5)
36F2TWUgvzXHh6q2hmmEtPRuxJGQvhBjuu	0.05428736 BTC (5)
3BLF1KPJPr6RosRpBoqFkAdB5gcBUUZytC	0.0537733 BTC (5)

FEE: 0.00006921 BTC 604 CONFIRMATIONS 655.31072442 BTC

Notice the details of the block, including when the block was mined. The date should be similar to the creation date of the transaction.

Also notice a link to the block containing this transaction. Let us few the block.

EXPLORE A BLOCKCHAIN - 6

Block Explorer News Market Bitcoin cash Zcash Blocks Status Buy Bitcoin with CC!

Search for block, transaction or address Conn 73 Height 517067 Scan BTC

Block #517067

BlockHash 0000000000000000000000004702d0996df37ed359270178d2ff839bce16f6d36190ed

Summary

Number Of Transactions	1314
Height	517067 (Mainchain)
Block Reward	12.5 BTC
Timestamp	Apr 7, 2018 6:55:01 AM
Mined by	
Merkle Root	1351a9fb7cc1823be43893466388d...
Previous Block	517066
Next Block	517068

Transactions

2ae f410194afc6aa893340850be2ae09f1316c13deca62d7bf6ecbb2e2d6d878
mined Apr 7, 2018 6:55:01 AM

No Inputs (Newly Generated Coins)	1C1mCxRukix1KfegAY5zQQJv7samAciZpv	12.5532991 BTC (\$)
	Unparsed address [0]	0 BTC (U)
	605 CONFIRMATIONS	12.5532991 BTC

b7c44c50dd042ecb13a1cca8183245e5bcbb99553d7da172eecc7719b6ae0a6eac
mined Apr 7, 2018 6:55:01 AM

1MmwU8zvcvFehZD2ASPF8ZKceskYyD1sY	0.1 BTC	17zE9MijuxheYLwhMdQPktxEqjW6UH9	0.00775232 BTC (U)
158c2btTk7imGyNLrjf52ctTeekFxpbimSMQ	0.00527786 BTC	14yqcQzhym8oUCzt6iFadshn3k58uVNts8	0.0952778 BTC (\$)
FEE: 0.00224774 BTC		605 CONFIRMATIONS	0.10303012 BTC

View of the block.

EXPLORE A BLOCKCHAIN - 7

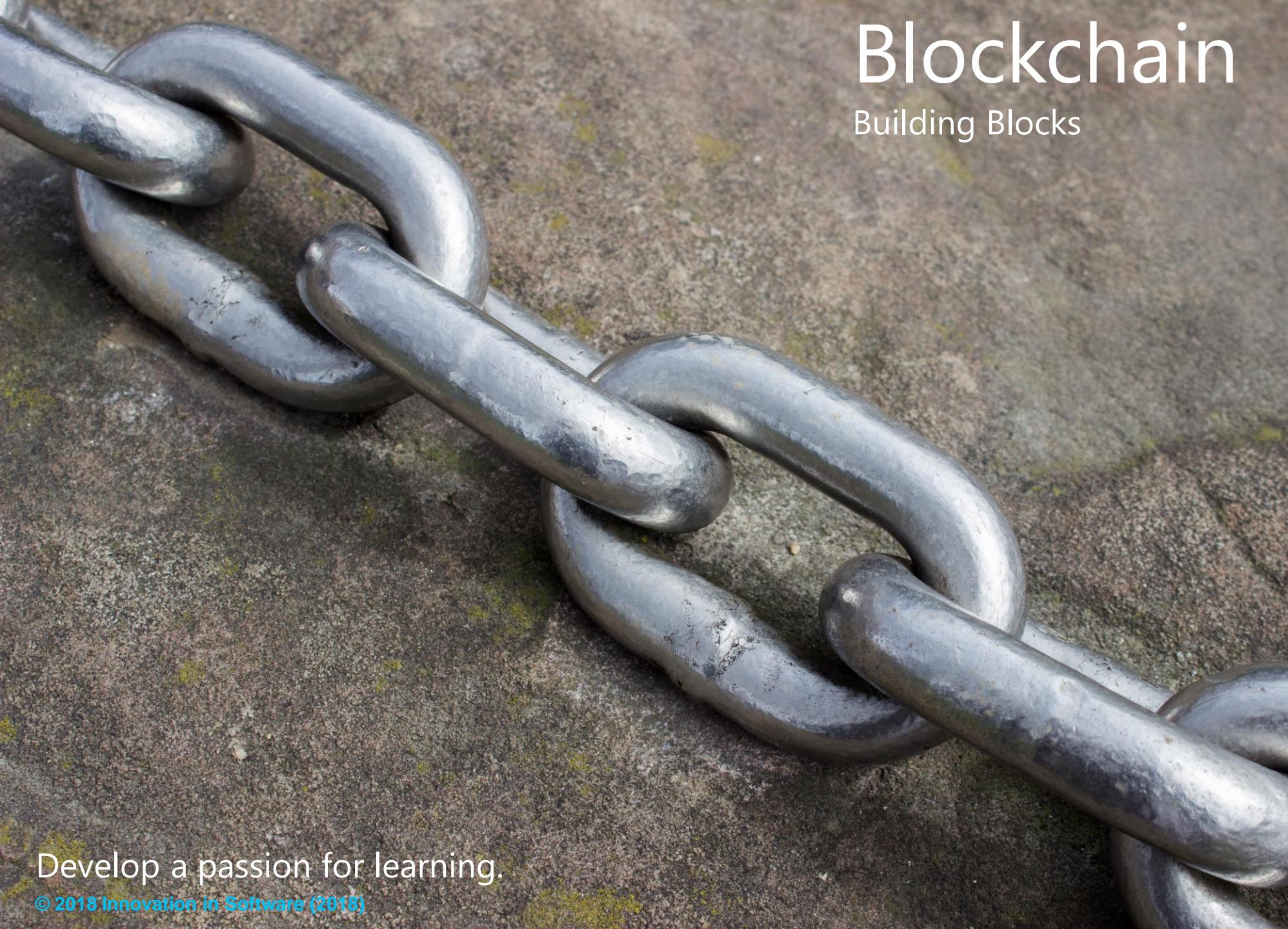
Blocks by date.

Height	Timestamp	Transactions	Mined by	Size
517671	Apr 11, 2018 12:58:04 AM	1732		833142
517670	Apr 11, 2018 12:46:53 AM	2566		959363
517669	Apr 11, 2018 12:17:02 AM	452	SlushPool	298426
517668	Apr 11, 2018 12:13:48 AM	569		203448
517667	Apr 11, 2018 12:09:36 AM	619		265468
517666	Apr 11, 2018 12:03:32 AM	46	AntMiner	16528
517665	Apr 11, 2018 12:03:15 AM	1787		777228
517664	Apr 10, 2018 11:46:22 PM	1151		449210
517663	Apr 10, 2018 11:36:05 PM	740	AntMiner	289015
517662	Apr 10, 2018 11:30:30 PM	1845		798077
517661	Apr 10, 2018 11:19:47 PM	2114		949736
517660	Apr 10, 2018 10:56:08 PM	1166	AntMiner	584774
517659	Apr 10, 2018 10:42:41 PM	334		126233
517658	Apr 10, 2018 10:39:40 PM	689		275263
517657	Apr 10, 2018 10:32:15 PM	1164		645404
517656	Apr 10, 2018 10:20:05 PM	591	BTCC Pool	343678
517655	Apr 10, 2018 10:14:52 PM	1046		402126
517654	Apr 10, 2018 10:03:48 PM	672	SlushPool	298405
517653	Apr 10, 2018 9:57:29 PM	664		319828
517652	Apr 10, 2018 9:50:19 PM	19		4810
517651	Apr 10, 2018 9:50:12 PM	1398		643737
517650	Apr 10, 2018 9:33:57 PM	2437		930063
517649	Apr 10, 2018 9:07:10 PM	92		26357
517648	Apr 10, 2018 9:06:13 PM	353		130620
517647	Apr 10, 2018 9:02:39 PM	27		6710
517646	Apr 10, 2018 9:02:19 PM	26		7087
517645	Apr 10, 2018 9:02:06 PM	943		427053

Choose the Blocks link for a listing of the blockchain.

Notice the column for Mined By. Some of the names of miners are included in this column.

Click on the link for one of the miners, such as SlushPool, for more information.



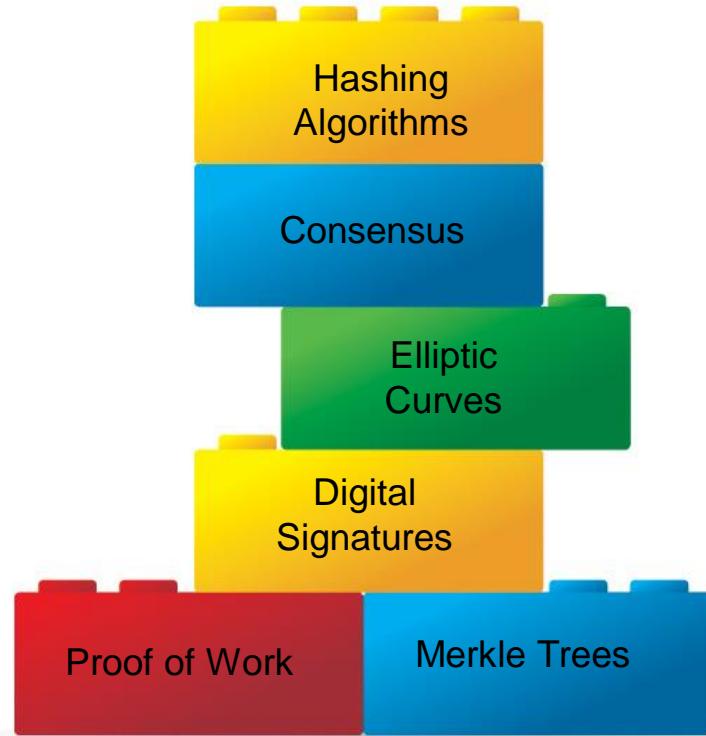
Blockchain

Building Blocks

Develop a passion for learning.

© 2018 Innovation in Software (2018)

BUILDING BLOCKS



There are a variety of concepts, or building blocks, that are required to implement a blockchain. This module covers those topics, such as hashing algorithms, digital signatures, and more.

HASHING

Hashing transforms a byte array into a fix length and unique value, which is a called a digest. The digest cannot be reversed. Even the smallest change to the source data can change the results completely.

SHA-2 (Secure Hash Algorithm 2) is the one-way compression routine used with blockchains.

For verification of data integrity; you compare the hash to a previously calculated hash. If the hash has changed, the original data has been compromised. For example, computing and comparing the hash of a block can verify whether the blockchain, more than a specific block, has been modified or tampered with.

SHA256

The SHA-2 family includes six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits:

- SHA-224
- **SHA-256**
- SHA-384
- SHA-512
- SHA-512/224
- SHA-512/256

SHA-256 is the most commonly used hash algorithm in blockchain.

SHA-256

You can test the SHA-256 algorithm at this website.

<http://passwordsgenerator.net/sha256-hash-generator/>

The screenshot shows the 'SHA256 Hash Generator' tool. At the top left is a green 'START NOW' button. To its right is a box containing '3 Easy Steps': 1. Click 'Start Now', 2. Free Access - No Sign up!, 3. Start Converting. Below this is a note from 'EasyFileConverter'. The main area has a text input field labeled 'Enter your text below:' with a large empty text area. Below the text area are two blue buttons: 'Generate' and 'Clear All'. To the right of these buttons is a checkbox labeled 'Treat each line as a separate string'. At the bottom left is a note 'SHA256 Hash of your string:' followed by a large empty text area. At the very bottom left is a copyright notice: 'Copyright © 2012 - 2016 PasswordsGenerator.net. All Rights Reserved.'

HASHING - CODE

Here is a general purpose function that converts string data to a hash.

The string must be converted to a byte array and then later returned to a Unicode string.

The function Hash2HexString converts the hash to a human readable string.

```
public static String SetHash(String data)
    throws UnsupportedEncodingException,
    NoSuchAlgorithmException {

    byte[] hash=null;
    String stringHash;
    MessageDigest digest =
        MessageDigest.getInstance("SHA-256");
    hash = digest.digest(
        data.getBytes(StandardCharsets.UTF_8));
    stringHash=Utilities.Hash2HexString(hash);
    return stringHash;
}
```

HASHING - ANNOTATED

sole parameter is a Unicode string (1)

declare a byte array to store hash (2)

returns a message digest object for SHA-256 (3)

convert string to byte array which is then hashed (4)

convert hash to readable text (5)

```
public static String SetHash(String data) 1  
    throws UnsupportedEncodingException,  
    NoSuchAlgorithmException {  
  
    byte[] hash=null; 2  
    String stringHash;  
    MessageDigest digest = 3  
        MessageDigest.getInstance("SHA-256");  
    hash = digest.digest(  
        data.getBytes(StandardCharsets.UTF_8)); 4  
    stringHash=Utilities.Hash2HexString(hash);  
    return stringHash;  
}
```

HUMAN READABLE HASH

```
public static String Hash2HexString(  
    byte [] hash) {  
  
    String buffer="";  
    for(byte character: hash) {  
        buffer+=Integer.toHexString(  
            0xFF & character);  
    }  
    return buffer;  
}
```

Hashes are a random sequence of byte values. As such, the hash may not be displayable. Hash2HexString iterates each character of the hash and incrementally transforms a hash into a displayable string.

This is the important code:

```
buffer+=Integer.toHexString(  
    0xFF & character);
```

The 8-bit byte, which is signed in Java, is sign-extended to a 32-bit int. Apply the 0xFF mask to the byte undoes the sign extension.

You might also consider converting the hash to Base58.

WALLET



Wallets are an optional feature of blockchains. A wallet contains an address for a user or group. The wallet also has a calculated value, which represents the net value of the transactions sent to and from this address.

Some wallet implementations use the keypair from public key cryptography. The private key is used for authentication, while the public key is the wallet address. For that reason, you can distribute the public key to anyone wanting to send you something of value.

WALLET - 2



The first bitcoin address is:

1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa,

and supposedly belongs to
Satoshi Nakamoto.

DIGITAL SIGNATURES

Public-key cryptography is used extensively in blockchains.

- Confirm the identity of a sender
- Verify ownership of something of value
- Avoids sender repudiation

You sign data with your private key to create a digital signature. The signature can be confirmed later using the public key.

Digital signatures are not encryption and the original data cannot be discovered from the signed data.

DIGITAL FINGERPRINT



Digital fingerprints prove the authenticity of data (i.e., data is not tampered with). Digital fingerprints refers to the output of a one-way function. This allows you to identify large amounts of data or files through a small fixed sized byte representation, which is called a digest. For Blockchain, hash are used for this purpose. Importantly, you cannot reverse engineer the source data from the digest.

ELLIPTIC CURVE CRYPTOGRAPHY

The public key created for blockchains must be truly unique. Collision could result in something of value being sent to the wrong person or destination. Private keys need to be sufficiently randomized to prevent potential problems. With elliptic curve cryptography the possible combinations are sufficiently large to virtually prevent any collisions. For elliptical curve algorithms a public key is a location on curve, represented by X and Y coordinates.

Using elliptic curve multiplication, a one way cryptography function, you can generate the public key.

Here is an excellent introduction to elliptic curve cryptography.

<https://bit.ly/2fzPr8O>

ELLIPTICAL CURVE - CODE

This code creates a public and private key using an elliptic curve. For convenience, a elliptic curve provider from Bouncycastle.org is used:

<https://bit.ly/2EsZgf5>

For the function KeyPairGenerator.getInstance, "ECDSA" selects the Elliptic Curve Digital Signature Algorithm from the Bouncycastle provider ("bc").

```
KeyPairGenerator  
keyGen=KeyPairGenerator.getInstance("ECDSA", "BC");  
  
SecureRandom  
random=SecureRandom.getInstance("SHA1PRNG");  
  
ECGenParameterSpec ecSpec=new  
ECGenParameterSpec("prime192v1");  
  
keyGen.initialize(ecSpec, random);  
  
KeyPair keyPair=keyGen.generateKeyPair();  
  
privateKey=keyPair.getPrivate();  
  
publicKey=keyPair.getPublic();
```

ELLIPTICAL CURVE - ANNOTATED

get an object for creating a public / private key pair (1)

get a pseudo random generator based on SHA1 (2)

use the X9.62 curve prime192v1 1 (3)

Initialize the key generator for an elliptic curve and a source for randomness (4)

create public and private key (5)

extract the private key (6)

extract the public key (7)

```
KeyPairGenerator keyGen=KeyPairGenerator.getInstance(  
    "ECDSA", "BC");
```

```
SecureRandom random=SecureRandom.getInstance("SHA1PRNG");
```

```
ECGenParameterSpec ecSpec=new  
    ECGenParameterSpec("prime192v1");
```

```
keyGen.initialize(ecSpec, random);
```

```
KeyPair keyPair=keyGen.generateKeyPair();
```

```
privateKey=keyPair.getPrivate();
```

```
publicKey=keyPair.getPublic();
```

1

2

3

4

5

6

7

ENCRYPTION

Although Bitcoin does not use encryption, you may however leverage encryption in your blockchain to keep sensitive data secret. There is no rule against that.

You can use an encryption algorithm to translate data into a secret, which is called a cipher. Decryption is the process of converting a cipher back to the original data.

Symmetric encryption is done with a single key. The key is used for both encryption and decryption.

Asymmetric encryption requires a keypair. The private key is used to encrypt data, while the public is used decrypt.

ENCRYPTION - CODE

The goal of encryption algorithms is to prevent the decryption of a cipher without using the key. The likelihood should be equivalent to brute force, which is typically impractical. Longer keys increases the difficulty of decrypting a cipher.

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import javax.crypto.Cipher;

public class Sample {
    public static void main(String [] args) throws Exception {
        // generate public and private keys
        KeyPair keyPair = buildKeyPair();
        PublicKey pubKey = keyPair.getPublic();
        PrivateKey privateKey = keyPair.getPrivate();
    }
}
```

ENCRYPTION – CODE - CONTINUED

```
// encrypt the message  
byte [] encrypted = encrypt(privateKey, "This is a secret message");  
System.out.println(new String(encrypted)); // <>encrypted message>  
  
// decrypt the message  
byte[] secret = decrypt(pubKey, encrypted);  
System.out.println(new String(secret)); // This is a secret message  
}  
  
public static KeyPair buildKeyPair() throws NoSuchAlgorithmException {  
    final int keySize = 2048;  
    KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");  
    keyPairGenerator.initialize(keySize);  
    return keyPairGenerator.genKeyPair();  
}
```

ENCRYPTION – CODE - CONTINUED

```
public static byte[] encrypt(PrivateKey privateKey, String message) throws Exception {  
    Cipher cipher = Cipher.getInstance("RSA");  
    cipher.init(Cipher.ENCRYPT_MODE, privateKey);  
  
    return cipher.doFinal(message.getBytes());  
}  
  
public static byte[] decrypt(PublicKey publicKey, byte [] encrypted) throws Exception {  
    Cipher cipher = Cipher.getInstance("RSA");  
    cipher.init(Cipher.DECRYPT_MODE, publicKey);  
  
    return cipher.doFinal(encrypted);  
}  
}
```

RSA

RSA is the most popular algorithm for encrypting and decrypting data and was defined in 1978. RSA is a moniker for Ron Rivest, Adi Shamir, and Leonard Adleman, who together for published the RSA algorithm. It is an asymmetric algorithm relying on a keypair.

POP QUIZ: CRYPTOGRAPHY ALGORITHMS



5 MINUTES



What is the difference from using SHA-256 versus RSA for encryption and decryption?

CONSENSUS

The blockchain is a decentralized network without a central authority. How often have you heard this already? Without a trusted authority, there must be another mechanism to make decisions for the network. The nodes themselves make the important decisions through consensus. There are a variety of ways that consensus is used within a blockchain, including forking and accepting new blocks.

PROOF OF WORK

Work must be done to add a block to the blockchain. Completing the complex algorithm is considered *proof of work*. Performing proof of work to add a block to the blockchain is called mining. Proof of work is an investigation (mining) a miner (node) performs, which often requires installing a powerful computer or mining rig to solve a complex mathematical puzzle. This verifies not only the block but the transactions contained within the block.

Multiple miners on the network compete to be the first to find a solution for the mathematical problem pertaining to the candidate block. The first miner to solve the problem announces their solution simultaneously to the entire network.

Sometimes the work is made artificially *difficult* to protect the security of the blockchain. This makes it computational improbable for a hacker to attack the entire blockchain. Overtime mining becomes more efficient and requires less time to complete. For that reason, the complexity of work periodically increases to maintain a consistent amount of effort to add a block.

HASHCASH

Hashcash is part of the proof of work performed by the mining algorithm for various cryptocurrencies. Hashcash entails completing a predetermined amount of work that can be quickly verified. Hashcash was originally used to protect against email spam and denial of service attacks.

For email, a Hashcash stamp is added to the header, which requires some effort. This modest overhead can scale to an enormous amount of effort (and cost) for a spammer and acts as a deterrent.

Alternatives to Hashcash includes Cuckoo Cycle and Scrypt.

PROOF OF WORK - BITCOIN

For bitcoin, blocks are added about every ten minutes. This is controlled by the Hashcash mining algorithm that is implemented for blockchains. A block is mined when the current hash is prefixed with a certain number of zeroes. The numbers of zeros is the difficulty. This must be accomplished through brute force and iterating a nonce. For example, a difficulty of three would mine blocks until these results are found. More about this later.

0003WLR3k2X2

0000RA94212W

000Q112RT43E

000PEQ12T543

PROOF OF WORK – BITCOIN - CODE

the *difficult* parameter sets the level of difficulty (1)

creates the comparison string for determining if the mathematical puzzle has been resolved (2)

is a solution found? (3)

increase the nonce (4)

recalculate the hash (5)

mining completed (6)

```
public void mineBlock(int difficult) {  
    1  
    String target=new String(  
        new char[difficult]).replace('\0', '0');  
  
    while(!hash.substring(0, difficult).equals(target)) {  
        2  
        nonce++;  
        3  
        hash=calculateHash();  
        4  
    }  
    5  
    System.out.println("Block mined!! : "+hash);  
    6  
}
```

PROOF OF STAKE

Proof of Stake (PoS) is emerging as an alternative to Proof of Work where a consensus algorithm is used to validate blocks. In PoS, blocks are assigned to miners using a randomization algorithm. The selection is influenced by the amount of cryptocurrency (the stake) held by the miner and how long. This replaces selecting miners by computation power with miners that have a stake in the health of the blockchain.

Randomization negates the arms race that has emerged with miners. Power centralizes on the miners with the most computational power, which itself creates a security vulnerability. The conundrum is that the powerful miners will receive most of the compensation. This disparity means that powerful miners can acquire more resources and increase their dominance.

GREENER SOLUTION



Proof of work encourages more and more computational power as a winning strategy for mining blocks. This consumes a lot of energy! Proof of Stake takes away the energy and computational power requirement of PoW and replaces it with stake

This is the reason that several cryptocurrencies have adopted PoS.

- Blackcoin
- Lisk
- Nxt
- Peercoin

PROOF OF STAKE - SCENARIO



There are four miners for a new cryptocurrency called VegasCoin:

- Bob 35 coins
- Rebecca 40 coins
- Sarah 5 coins
- Herbert 20 coins

The miners will be selected in a random but weighted order. Bob has a 35% chance of being selected, Rebecca 40%, and so on.

MERKLE TREE

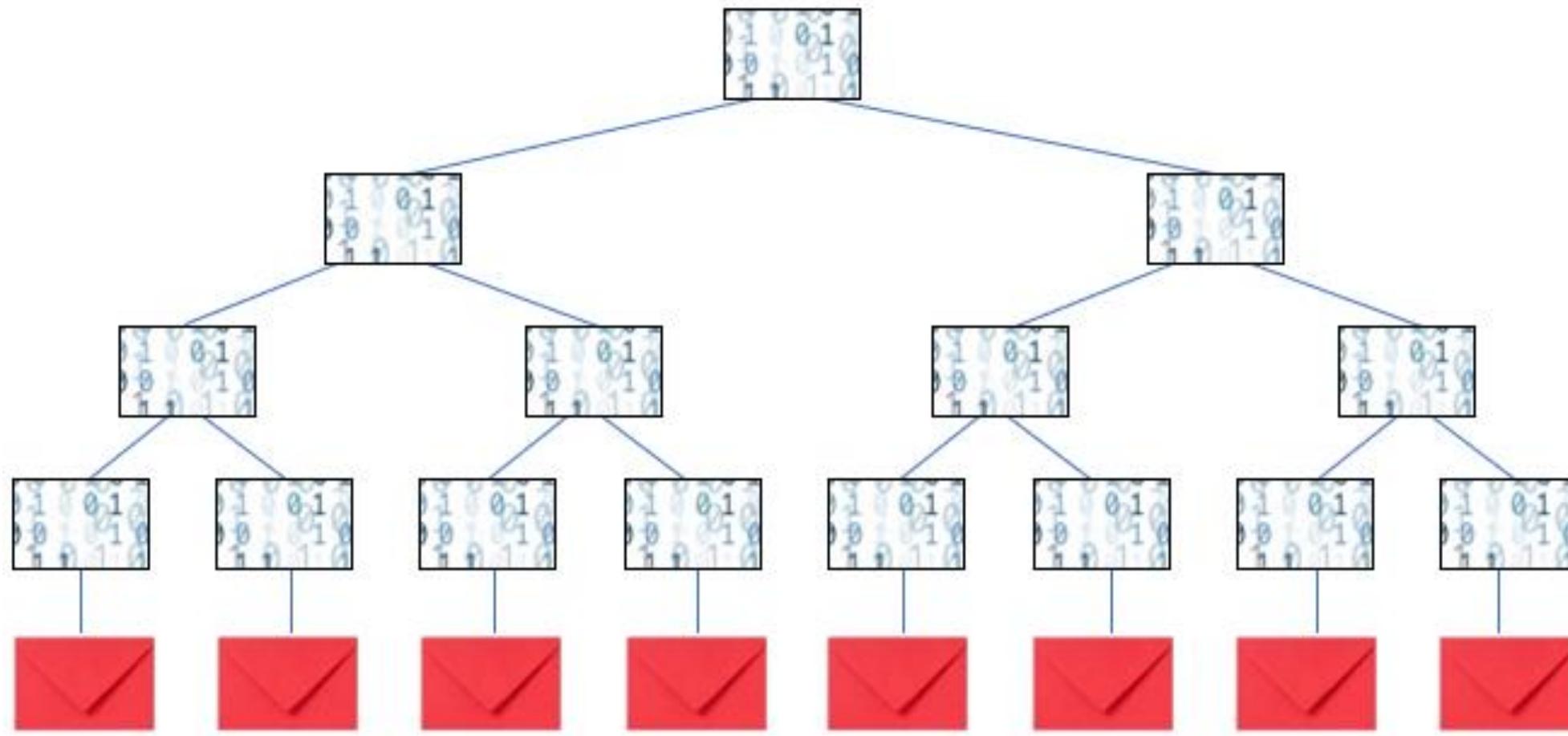
A Merkle Tree is a binary structure that efficiently and quickly allows the verification of large amounts of data. This structure has an important role in blockchain for secure verification of transactions. The Merkle Tree creates a digital fingerprint (hash) that is effective in determining if a transaction is contained within a block.

Merkle Trees are created by hashing pairs of nodes and then pairs of hashes until the top of the tree is reached – root hash. Hashing is typically done using a SHA-2 algorithm.

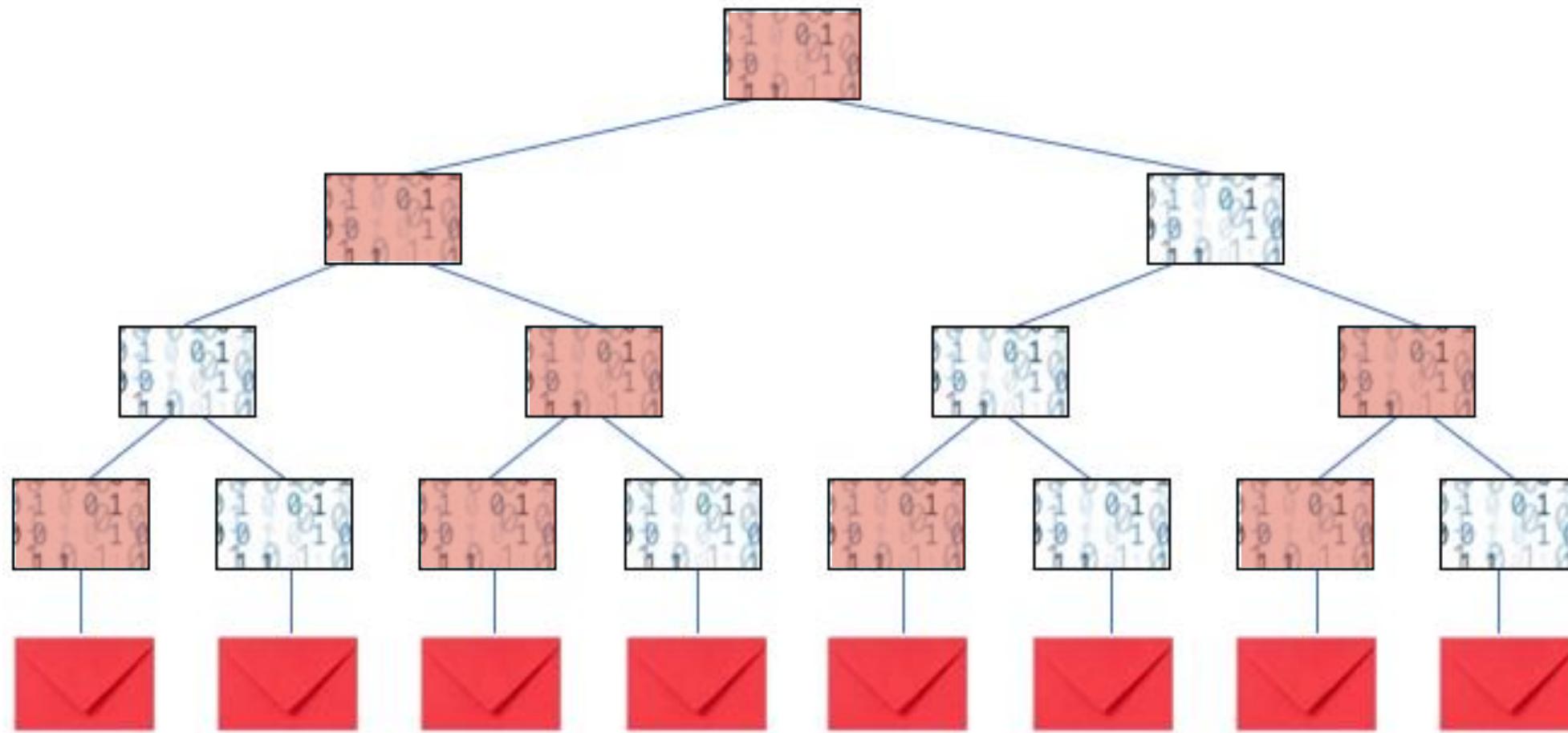
For a Bitcoin blockchain, the root hash is stored in the block header.

One advantage of a Merkle Tree is the capability to download and verify individual branches. Another advantage is that a Merkle Tree requires minimum memory to verify large numbers of transactions.

MERKLE TREE DIAGRAM



MERKLE TREE DIAGRAM - 2



SIMPLE PAYMENT VERIFICATION (SPV)

A Simplified Payment Verification (SPV) node is an efficiency that allows for the verification of transactions without downloading all the block data of the relevant blockchain. Merkle trees are used extensively by SPV nodes. The SPV node downloads only metadata, including the Merkle root. A SPV node downloads less than a gigabyte or two (depending on the size of the blockchain) versus tens of gigabytes of data.

As mentioned, SPV nodes do not have all the data. For Bitcoin, SPVs only receive block headers; but not the block body, which includes the transactions. Even without the block data (i.e., transactions), the SPV node can still verify a transaction with the assistance of a full node.

TRANSACTION VERIFICATION SPV

Scenario: A SPV node wants to confirm that a transaction exists within a block on the blockchain.

When the SPV receives a transaction, it can broadcast a request to the full nodes that exist on the network. The question / message is "does this transaction exist"? Full nodes benefit from having all of the data from the blockchain. They can iterate through the blocks, recreating the Merkle Tree, until the transaction is located. The Merkle Tree is recreated using buddies.

Once found, the full node sends the block hash and Merkle buddies to the SPV node. The SPV can then confirm the transaction exists and the correct block.

Lab 3- Merkle Root



MERKLE TREE

In this lab, create a Merkle Tree.
You will be provided the
framework of an application to
complete.

Merkle Trees are integral part of
the blockchain.

Import various packages as
needed.



MERKLE TREE

Create a class call MerkleTree.

Here is the method for hashing strings. You can enter "as is".

```
public String getSHA2HexValue(String str) {  
    byte[] cipher_byte;  
    try{  
        MessageDigest md = MessageDigest.getInstance("SHA-256");  
        md.update(str.getBytes());  
        cipher_byte = md.digest();  
        StringBuilder sb = new StringBuilder(2 * cipher_byte.length);  
        for(byte b: cipher_byte) {  
            sb.append(String.format("%02x", b&0xff) );  
        }  
        return sb.toString();  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return "";  
}
```

MERKLE TREE 2

Before the getSHA2HexValue function, define another function called GetNextMerkleRow. The function has a single parameter: List<string>. It also returns a List<string>.

This function accepts the current row of the Merkle Tree and returns the next row. You will build the Merkle Tree bottom-up (i.e., starting with the bottom row).

Define a List<string> for storing the next row of the Merkle Tree.

Define an index counter.

MERKLE TREE 3

Still in the GetNextMerkleRow function, iterate elements of the current row in the Merkle Tree with a while loop. Continue until the count exceeds the elements along the row. Combine each item with a buddy and then hash.

- Get left element at index counter

```
String left = tempTxList.get(index);
```

- Get right element at index counter +1. If right element does not exist, assign empty string.
- Combine left and right string. Hash the resulting string with the helper function.
- Add to the list for the next row
- Increment the count

Return the next row in the Merkle Tree.

MERKLE TREE 4

At the top of the class, create the data members:

- Define *elements* as a variable which is a List<String>. This is the source elements for the Merkle Tree.
- Define root which is a string. This will contain the root hash of the Merkle Tree.

Create a public constructor for the MerkleTree class, which accepts a List<String> as a parameter. Initialize *elements* with the parameter. Set *root* to be an empty string.

Create an accessor function that returns the root string.

MERKLE TREE 5

Create a new function: Merkle_Tree. You will call Merkle_Tree to create the actual Merkle Tree.

In the function, first define a List<String> variable called tempTxList. Initialize with the source elements of the Merkle Tree: elements. In a for loop, copy the items from elements into tempTxList.

Get the bottom row of the Merkle Tree. Call GetNextMerkleRow with tempTxList as the parameter. Save the bottom row into a new temporary list: List<String>.

In a while loop, loop until there are no remaining rows in the Merkle Tree:

```
while (newTxList.size() != 1) {
```

Inside the loop, get the next row of the MerkleTree. Call GetNextMerkleRow with the temporary list and save results back into same variable.

After the while loop is complete, you are at the top of the Merkle Tree. Assign the first element of the temporary list to the root.

MERKLE TREE 6

Test the Merkle Tree with the following code:

```
List<String> tempTxList = new ArrayList<String>();  
tempTxList.add("a");  
tempTxList.add("b");  
tempTxList.add("c");  
tempTxList.add("d");  
tempTxList.add("e");  
  
MerkleTree merkleTree = new MerkleTree(tempTxList);  
merkleTree.merkle_tree();  
System.out.println("root : " + merkleTree.getRoot());
```

Blockchain / Blocks



Develop a passion for learning.

© 2018 Innovation in Software (2018)

BLOCKCHAIN

The blockchain is an ordered, cross-linked, sequence of blocks; where each block contains transactions.

Blocks are uniquely identified with a SHA-256 cryptographical hash. Each block also references the previous block in the structure by its hash. The previous block is called the parent block, while the current block is the child.

Blockchains create a generational relationship for blocks in the chain. If you modified a parent block, the change will reverberate through all descendants, which is expensive. All the descendant hashes would have to be recalculated and proof of work redone. This computational overhead makes a blockchain virtually immutable. The longer the blockchain the more immutable.

POP QUIZ: PARENTS AND CHILDREN



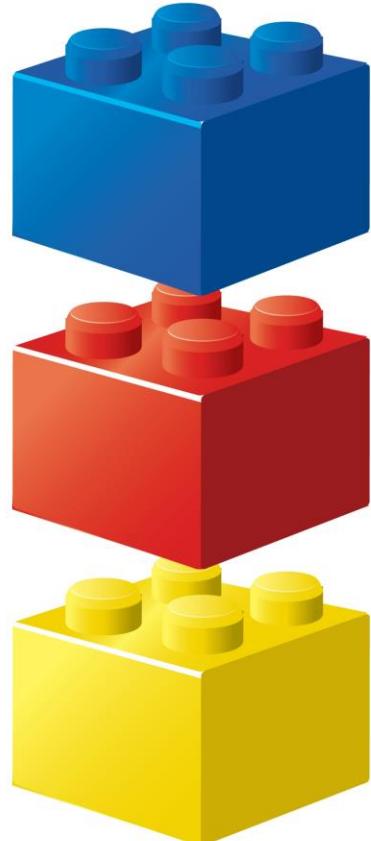
10 MINUTES



Can a blockchain block have more than one parent?

Can a blockchain have more than one child?

GENESIS BLOCK



Every blockchain must have a first block. This is called the genesis block. Starting with the latest block, previous hashes create a chain to the first block, which is the genesis block (i.e., Block 0). The previous hash for the genesis block is null.

The blockchain height is the distance from the genesis block to the top of the chain.

BLOCK DETAILS

The block contains transactions that have been broadcast to nodes of the blockchain. The transactions are the “data” of the block and represent transfer of value.

For Bitcoin, a block is separated into a block header and body. For example, hashes from previous blocks are stored in the block header, while the body contains transactions. The block is 80 bytes, while the average body is around 125kb. There are many other fields in a block.

.

BLOCK SCHEMA

Size	Field	Description
4 bytes	Block size	The size of the overall block
80 bytes	Block header	Block hasher, Merkle root, etc.
1-9 bytes	Transaction counter	Number of transactions in body
	Transactions	Transactions in block

This is the overall schema for blockchain blocks. Minor deviations for your particular blockchain is not only accepted but assumed. Consider the block schema to be the base type which you can extend as necessary.

BLOCK HEADER

Size	Field	Description
4 bytes	Version	Reference to a specific Bitcoin protocol
32 bytes	Hash (previous block)	Hash of the previous block
32 bytes	Merkle root	Reference to the transactions
4 bytes	Timestamp	Block creation time
4 bytes	Difficulty	POW difficulty
4 bytes	Nonce	POW counter

The block header includes a reference to the previous block. Version number to pair with the correct implementation. The difficulty, timestamp, and nonce are used for mining the block. Finally, the block contains a Merkle root, which is a reference to the payload (i.e., transactions).

Note: the block does not contain the hash of the current block.

POP QUIZ: BLOCK HASH



5 MINUTES



Is there is a practical reason that the hash of the current block is not contained in the header?

BLOCK IDENTIFIER

Blocks are identified with the hash or height.

The block hash is calculated using the SHA-256 algorithm with the block header as input. It is always calculated at the destination. However, the block hash is stored as the previous hash in the child block.

Some blockchains save the block hash in a separate list for quick indexing.

BLOCK IDENTIFIER - CONTINUED

Row height can also be used to identify a block within a blockchain. First block in block chain is block⁰, which is the genesis block. This is the hash of block⁰ of the Bitcoin blockchain:

000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

For Bitcoin, the block height on January 1, 2017 was 446,000.

POP QUIZ: BLOCK HEIGHT



5 MINUTES



Is the block height for a specific block always unique?

Can you find the genesis block using either blockchain.info or blockexplorer.com?

<https://blockchain.info/block/{blockHash}>

<https://blockexplorer.com/block/{blockHash}>

HASH

Hash the elements of the block you do not want tampered with using the SHA-256 algorithm.

```
function CalculateHash  
    szIntermediateResult=szPreviousHash+szTimeStamp+  
        createMerkle(transactions)  
    szCurrentHash=CreateHashAsString(szIntermediateResult)  
    return szCurrentHash
```

BLOCKCHAIN

Creating a block chain is simply adding one block to the next.

```
function entryPoint
    arrayBlockchain
        block1=blockchain.add(new block, nothing)
        block2=blockchain.add(new block, block1)
        block3=blockchain.add(new block, block2)
```

IS BLOCK CHAIN VALID

You can confirm the correctness of the blockchain by iterating the blocks and examining the current and previous hashes. Rehash the blocks and compare to the saved hashes within the blocks to confirm that the immutable blockchain has not changed. Finally, confirm that each block is mined. Blockchain should not include any blocks that are not mined.

ISCHAINVALID

```
function IsChainValid

    blockCurrent=Block

    blockPrevious=Block

    szHashTarget=zeros[difficulty]

    for(increment i=1 to intSizeOfBlockchain)

        currentBlock=Block(i)

        previousBlock=Block(i-1)

        if(!currentBlock.hash. ==  currentBlock.CalculateHash())

            return false;

        if(!previousBlock.hash ==  currentBlock.previousHash)

            return false;

        if(!currentBlock.hash.substring[0,3] ==  hashTarget)

            return false;
```

DOUBLE SPENDING

Double spending is one of the important problems that central authority prevents. This means transferring the same value twice. With a cryptocurrency, this means spending available coins twice (i.e., overdrawing the account). With a real estate blockchain, that could mean selling a house twice to different people. As the central authority, a bank prevents that in the real world.

For a blockchain, double spending is preventing by the overhead of proof of work and consensus.

LONGEST BLOCKCHAIN

Blockchain hosts a distributed ledger, where the ledger is replicated to each node. The longest valid blockchain is accepted as the correct entity for the network. What prevents someone from modifying a block and then submitting a longer chain. Remember, the blockchain is supposed to be immutable. Proof of work prevents this type of interference. The hashcash proof of work means it takes considerable effort to create new blocks. Remining the entire chain to present an altered chain would require more effort than feasible; especially while other block are being added in realtime.

BLOCK FORKS

As mentioned, the block hash is unique; but block height is not necessarily unique. Block forks can create duplicate height. When do forks occur? Block forks can temporarily occur as the result of PoW. Two nodes complete POW almost simultaneously and broadcast the same block. When that occurs, the longest blockchain rules applies and the nodes select the appropriate block.

Lab 4- First Blockchain



FIRST BLOCKCHAIN - 1

In this lab, you will create your first blockchain. Have fun!

Import various packages as needed.

Download and install Gson for Eclipse. For help, watch this video:

<https://bit.ly/2JBucO1>

Create a new Java project. The initial class is *Block*. Define the initial data members:

- hash: string (not normally saved in block)
- previousHash: string
- data: string
- timestamp: long
- blockId:int
- count:static int

FIRST BLOCKCHAIN - 2

Also create four public accessor functions that return the value of each data members, except blockId and count, for example getHash.

Create a new class called StringUtil. In the class, define a function that will hash data. The function will accept a single string parameter and return a string (i.e., the hash).

Inside a try block:

- Convert the string parameter into bytes with getBytes function. The string is format is Unicode.
- Define a MessageDigest object with getInstance to create a one-way hash. The hash name is "SHA-256".
- Create the hash with MessageDigest.digest. The single parameter are the bytes from the string parameter. The result is the hash in bytes.
- Define and initialize an empty StringBuffer called hexString.

FIRST BLOCKCHAIN - 3

Still in the hashing function, convert the hash into a readable string.

- Define a for loop that iterates the bytes of the hash.
- Convert each byte to an hexadecimal character:

```
String hex=Integer.toHexString(0xFF & hash[i]);  
if(hex.length() == 1) hexString.append('0');  
hexString.append(hex);
```

Return hexString from function

If exception is raised, throw a new RuntimeException.

FIRST BLOCKCHAIN - 4

In the Block class, add a public function: CalculateHash. The function returns a string. The function will create a hash for the Block class.

- Define a string with combines previousHash+string version of timestamp+data
- Call the hashing utility with the string
- Return the hash from the method

Create a public constructor to initialize the Block class.

- Two parameters: data:string and previousHash:string
- Assign data and previousHash to the identical members of the class
- Assign timestamp the current date using the Date class
- Update hash with CalculateHash function
- Increment the static count variable
- Assign blockId the count.

FIRST BLOCKCHAIN - 5

Define a new class from main – the entry point for your application.

You need a blockchain! In the class, create an ArrayList generic of Block types.

In main, add blocks to ArrayList:

```
blockchain.add(new Block("Hi im the first block", "0"));
```

Add three more blocks to the blockchain using arbitrary data.

In for loop, display each block with the following data:

- blockId
- blockHash
- previousHash

FIRST BLOCKCHAIN - 6



Extra credit:

Earlier in the module, pseudo code was presented to test the validity of a blockchain. Create a function called IsChainValid. After creating the blockchain, test correctness of blockChain with IsChainValid. Display an appropriate message after calling function.