[[1]](#footnote-1)

Blockchain Analysis (April 2019)

Daniel Byrne, *Graduate Student, Southern Methodist University*, Benjamin Wilke, *Graduate Student, Southern Methodist University*, Brian Coari, *Graduate Student, Southern Methodist University,* YuMei Bennett, *Graduate Student, Southern Methodist University*,

*Abstract*— Blockchain technology is increasingly being developed for use in a broad range of applications, from financial transactions, digital voting systems, identity management, and asset tracking. The technology is robust, secure, flexible, and could one day bring about fundamental change in a whole host of common computer systems. The goal of this project is to provide a primer on Blockchain technology and an evaluation of an experimental Blockchain database.

*Index Terms*— Computer network management, Cooperative communication, Cryptographic protocols, Data transfer, Encryption, Distributed databases, Document handling, Fault tolerant control, Identity management systems, Information exchange, Internet, Network function virtualization, Network security, Peer-to-peer computing

# INTRODUCTION

B

lockchain is, at its heart, simply an immutable transaction log, often referred to as a digital ledger, which employs cryptographic hashing techniques in a distributed fashion to create a one-way transaction register which memorializes the characteristics of an interactions between parties and secures those actions with completed proofs of work and a distributed, decentralized, and public log. It was originally designed as a means to facilitate commerce on the Internet without the need for a trusted intermediary such as a bank or a monopolistic computing company like Microsoft or Google overseeing and injecting itself into transactions. Blockchain however, due to its robustness, versatility, and security, has been leveraged into a whole host of other applications for which trust, privacy, resilience and security are necessities.

This paper will explore the distributed ledger technology known as Blockchain. In this paper we will cover what Blockchain is, the theories and technologies behind Blockchain, its strengths and shortcomings, as well as its current and possible future uses. We will explain the different types of [Blockchains](https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/) (public, private, and federated), and the pros and cons of each. This paper will conclude with a description of an emerging application of Blockchain, [BigchainDB](https://www.bigchaindb.com/), which is a NoSQL database augmented with the core strengths of Blockchain, its decentralized nature, robustness to node failures, and its security.

# Related Work

## Early Cryptography

Until about the 1970s, cryptography was mainly the purview of state actors. However, in the 70s, two publications brought cryptography into the public eye the US Government’s DES,the Data Encryption Standard, and [Whitfield Diffie & Martin Hellmanand’s publication on public key cryptography](https://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf).

## DES

The Data Encryption Standard is a symmetric-key, block cipher algorithm for the electronic data encryption first developed at IBM, and then submitted as a candidate in a invitational study to the National Bureau of Standards, now the national Institute of Standards and Technology, following the agency’s invitation to help define a government wide standard encryption algorithm for the protection of sensitive, but unclassified electronic government data.7 IBM’s initial submission was modified to use S-blocks and to decrease the number of bits used for the key as suggested by the National Security Agency before final ratification of the standard. The final design was a symmetric key, 56-bit, block encryption algorithm using [S-block](https://ia801307.us.archive.org/33/items/AmericanCryptologyDuringtheColdWar1945-1989BookIII_RetrenchmentandReform1972-1980/American%20Cryptology%20During%20the%20Cold%20War%2C%201945-1989.%20Book%20III_%20Retrenchment%20and%20Reform%2C%201972-1980.pdf), or substitution blocks.

Symmetric-key or shared-key algorithms like DES are cryptographic algorithms the use the same or very mildly transformed keys for both encryption decryption of data. The keys enable the possibility of a secure communication link between 2 or more parties. The requirement of this type of secure channel is that all parties to the communication need to know the secret keys. This fact represents the main drawback to symmetric key algorithms as the number of communicating parties increases it becomes exceedingly more complex to engineer a secure means of transferring the secret keys before the encrypted communication using the keys can begin.

Block ciphers, like DES, are cryptographic algorithms which deterministically encrypt fixed length groups of bits. In a typical block cipher implementation data is split into equal size blocks which are then transformed with a secret key seeding a mathematical transformation. However, since each block is deterministic and unwavering, block ciphers are susceptible to an exploit called differential cryptanalysis in which output blocks are scanned for evidence of non-random behavior caused by some repeating value in the plaintext, and that information is used to reverse engineer the encryption keys.

Substitution blocks, at the time a new feature in cryptography are a cryptographic algorithm in which an input of n bits are transformed into an output of m bits, and where n and m are not necessarily alike. DES incorporated the first widely used implementation of substitution blocks in its 6x4 s-blocks in which 6 input bits were transformed to 4 output bits. S-blocks serve to obfuscate the deterministic regularities arising from simple block ciphers thus making them more resistant to differential cryptanalysis.

## Early DES Opposition

During the DES public comment period, parts of the proposed standard were deemed classified, and thus could not be vetted by researchers in the field. Also, it was made known that the NIST enlisted the help of the NSA in modifying the initial IBM proposal, and that made people a little bit wary.

Due to the partially classified nature of the publication, the involvement of the NSA, and the introduction of heretofore unknown feature in the design called S-boxes, the initial specification was immediately met with wide skepticism in the academic realm. Whitefield Diffie and Martin Hellman, prominent cryptography researchers at Stanford University, panned the standard in an “Exhaustive Cryptanalysis of the NBS Data Encryption Standard”. Primarily their concern was that the 56-bit key proposed by the standard was too short to be secure.

There was also public skepticism that the classified implementation of the S-boxes, now known as substitution boxes, in the design represented some sort of backdoor introduced by the NSA to allow it to decrypt secure communications. It wasn’t until years later when S-boxes were rediscovered, that it was determined that the inclusion of S-boxes in the DES standard made it more secure against differential cryptanalysis.

However, the NSA did push for a reduction of the key size from the originally proposed 64bits to 48bits. IBM pushed back and the IBM and the NSA eventually compromised on a 56-bit key. However, this weakened the standard so much, according to Diffie and Hellman, that state actors and other organizations with deep pockets could theoretically build a computer to crack DES encryption through brute force. In Diffie and Hellman’s early public shaming of DES they made this point in their opening introduction.

Fittingly, in 1998, the Electronic Frontier Foundation built a $200,000 machine, well within the financial means of companies and especially governments, that cracked DES in a few days. The project was headed by John Gilmore, a founder of the Cypherpunk movement and a cryptographic academic and activist. The goal of the project was to demonstrate that DES was insecure, and that the US government had been telling deliberate lies about the security of DES for some time.

## Public Key Cryptography

Diffie and Hellman and went on to advocate for an asymmetric encryption algorithm with greater complexity of the 56-bit DES and the ability to agree upon a shared key over an unsecure channel. This process became known as public key encryption and they went on to eventually patent this concept with Ralph C. Merkle in 1977. [The Diffie-Hellman Key Exchange Protocol](https://ieeexplore.ieee.org/document/1055638) is what is can become to be named.

Interestingly though, the concept was actually first conceptualized and secretly described in a now unclassified [paper](https://www.gchq.gov.uk/sites/default/files/document_files/CESG_Research_Report_No_3006_0.pdf) by James H. Ellis, Clifford Cocks, and Malcolm J. Williamson of GCHQ, the British signals intelligence agency, in 1969.

The private key must remain secure, but the public key can be widely disseminated without fear that the encrypted data will be compromised. The security of this cryptographic system is hinged on the sheer complexity and time it would take to factor the product of the two very large prime keys with current technology.

## Cipher Punks

Blockchain technology grew out of the tenants of a group of high tech privacy enthusiasts calling themselves Cypherpunks. The term Cypherpunk is a compound word combining root words cypher or cipher, a coded message, and punk, in this instance referencing a member of a rebellious counterculture group advocating the widespread use of cryptography as a means to social and political change.

The Cypherpunk movement traces its roots to the early days of public key cryptography. With the initial academic uproar surrounding the distrust of the NSA’s involvement in the manipulation of DES, and the secrecy in which the implementation of the algorithm was shrouded in, there began the seeds of dissent and the onus for the beginnings of the Cypherpunk movement. The eerily prescient early focus of the small, mostly online group was on discussing individual privacy in a digital world, government monitoring, and central authority control of information, all issues which are in national conversation today.

The Cypherpunk group started out in the as in informal meetup between tech minded individuals with scofflaw tendencies founded by Eric Hughes, Tim May, and John Gilmore. These three together with 20 of their friends began meeting regularly in the offices of John Gilmore’s company,Cygnus, a stalwart of the burgeoning open source community, in the early 1990s.

Their meetup shortly morphed into a mailing list to broaden its appeal and to attract likeminded individuals from around the world.

One of their primary concern was that of governments and large powerful corporate entities capturing information, as John Gilmore described the problem in a speech at the first ACM conference on Computers in March 1991,

"In most of Europe, phone companies don't record the phone numbers when you call, and they don't show up on your bill. They only tick off the charges on a meter. Now, I was told that this is partly because the Nazis used the call records that they used to have, to track and identify the opposition after taking over those countries in World War II. They don't keep those records anymore." [9]

The mailing list’s topics and tenor favored privacy in communications, self-revelation, financial privacy, anonymity, and pseudonyms. It also derided and actively opposed government data collection, forced self-revelation, and censorship. It is notable that an early and very active, member of the Cypherpunk mailing list was [Julian Assange](https://cryptome.org/0001/assange-cpunks.htm) of Wikileaks infamy, who has made it his life’s mission to expose corruption and abuse of power.

Privacy in communication was a primary concern of the movement, but equally important to the movement were discussions around financial privacy, as Eric Hughes, one of the founders, puts it in his [Cypherpunk Manifesto](https://www.activism.net/cypherpunk/manifesto.html),

"When I purchase a magazine at a store and hand cash to the clerk, there is no need to know who I am. When I ask my electronic mail provider to send and receive messages, my provider need not know to whom I am speaking or what I am saying or what others are saying to me; my provider only need know how to get the message there and how much I owe them in fees. When my identity is revealed by the underlying mechanism of the transaction, I have no privacy. I cannot here selectively reveal myself; I must always reveal myself."

## Bitcoin

From this primordial stew of technologist, university researchers, cryptographers, a love of privacy in communication and in financial transactions, and a general mistrust of central authorities an anonymous,short (9 pages), but groundbreaking [paper](http://nakamotostudies.org/wp-content/uploads/2018/03/bitcoin.pdf), was mailed to the Cypherpunk mailing list in 2008 by Satoshi Nakamoto, fittingly an alias, describing an electronic currency, Bitcoin, based upon cryptographic principals akin to asymmetric encryption, one-way functions, but in reverse. While Bitcoin was not the first digital currency created, it was the first to solve the problem of double spending and the first to do it without the need for a central authority regulating the transaction.

# Blockchain Technology

One of Blockchain’s core strengths is its ability to allow for a verifiable transaction to take place without the need for central database or oversight authority. Traditional transactions requires a central authority overseeing or simply recording the transaction to prevent fraud and to provide a verifiable record of was has transpired. Blockchain supplants this requirement with “cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party”. The consequences of this breakthrough are hard to overstate.

A Blockchain transaction begins with two willing parties to a transaction. The payor’s equity stake, or coin, is recorded as a transaction in a chain of cryptographic blocks secured with his public key, a hash, and the private key of the person he bought his equity stake from and embedded in a “proof of work” block. The payer then digitally signs a hash of this prior transaction and the public key of the next owner with his private key and then adds this to the end of the coin. The payee can then verify the signatures to verify the chain of ownership. This one-way cryptographic hash is computationally impractical to reverse, and thus this interaction secures the ownership chain; however, it doesn’t provide a means to prevent a bad actor from duplicating (double spending) or reversing a transaction.

The traditional means of ensuring the authenticity of a one-way transaction and to eliminate the problem of double spending is to use a trusted third party such as a bank or some governing body overseeing the transaction. While this works, it relies on the goodness of the intermediary and a measure of blind faith. Unfortunately, the intermediary might have other loyalties, an institution could be manned by rogue employees, or the entity could be inundated with entitlement mentality. Thus, this uncertainty manifests itself as an increase in the cost of performing a transaction. It also and has an inherent and an unfortunate but accepted level of grift and risk associated with it.

To get around the need for a central authority, the author(s) of Bitcoin, the first Blockchain technology, employ a proof of work concept in which honest “miner” nodes compete with one another to search for the solution to a difficult secure hash. When a transaction happens, it is added to a pool of other recent transactions. These transactions are picked up by special nodes in a peer to peer distributed network called miners. These miner nodes then search for a specific result to a complex hash comprised from the previous block’s hash, a nonce, and a collection of recent transactions that begins with a specific number of zeros bits. The complexity in finding this hash increases exponentially with the number of zeros specified.

When the hash is found, the miner who found it, adds that block to the end of the chain of blocks and widely publishes it to the network. Other miner nodes working on that block will accept the block if all transactions in the block are valid and not already spent. They signal their approval by then stopping trying to find the current block, since it has already been found, and proceed onto the next block using the recently accepted block’s hash as the previous block.

The Ledger is divided into Blocks of equal sizes and each block is connected to its previous block using the cryptographic hash of the data in previous block. By convention, the longest valid chain is always considered to be the source of truth. Shorter chains or chains with the same length but with a later timestamp, or chains with unverified blocks are discarded in favor the earliest and longest verifiable chain.

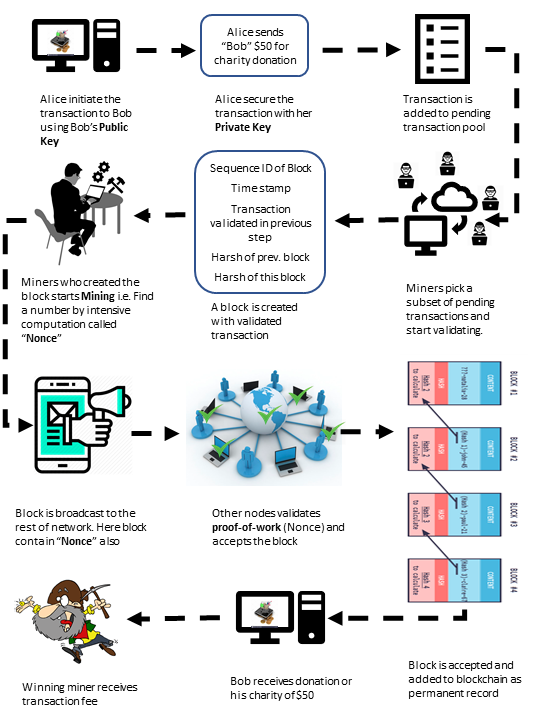
Every new block added to a chain of other blocks represents a non-trivial amount of work measured in electricity and time. If someone tries to change the value of a record in any block, all the blocks ahead of it are invalidated, thus providing an indication that the data has been tampered. To modify a past block, an attacker would have to redo the proof-of-work of a block and all blocks after it to catch up with and surpass the work of the honest nodes constantly and continuously finding new blocks. Assuming that the majority of computing power lies within the pool of honest nodes. The probability of this happening becomes vanishingly small. Conversely, if dishonest nodes do overtake honest nodes in the network, the value of the digital assets would plummet due to the inherent loss of trust, thus negating the spoils of the heist.

Since the ledger is distributed amongst every participant in the network, both miners and transaction participants, and not controlled by a central authority, publishing a fake transaction log again is limited by the ability of the bad actor to completely redo the proofs-of-work in the chain after its dishonest entry. Ironically then security is provided by publicity.

To incentivize honest nodes to continuously find new blocks and to process transactions to stay ahead of dishonest nodes attempts at rewriting history, miners finding a new block are awarded with a new coin. By convention this coin is recorded as the first transaction in a new block.

Here are the steps involved:

1. On a Blockchain, transactions are happening between different nodes on the network all the time. In our example, Alice initiate a transaction to Bob.
2. Alice’s wallet encrypt her coin with Bob’s public key and signs the transaction with her private key.
3. The transaction will not be added to the Blockchain right away. Instead it is added to pending transaction pool.
4. Special nodes, called miners, pick up several transactions from the pool.
5. Miners first validate all the transactions to see that all the transactions are fair and valid and does not contain any fraudulent transaction with problems like enough resources/fund in Alice wallet, duplicate transaction etc. (Validation rules are application specific.) These validation rules are called transactional consensus.
6. Once the transactions are validated, the miners will attempt to create a block of the transactions. This block contains a sequential ID, the transactions picked by the miner, a SHA256 Hash of all the values and a hash of the previous block.
7. The Blockchain Consensus then requires that a miner resolves a computational-intense mathematical problem to sign the generated block before broadcasting it to the network.
8. The other miners validate the solution of the problem and if validated accept the block.
9. Once over 50% of the Miners on the blockchain network accepts the block, the block is added to the blockchain and distributed to all nodes on the network.
10. Bob finally sees $50 in his charity donation wallet.
11. The winning miner who created the accepted block is awarded with transaction fee and/or a new coin.



# Blockchain Types

To do

# Blockchain Applications

In the year 2017, Blockchain became the second most popular search word on Gartner's website, and Distributed Ledger Technology (DLT) will continue to gain utmost significance across many industries.

Blockchain will help address several modern-day security concerns, including issues with contracts, identity, and fraud management. Blockchain-based lists will allow online retailers and financial organizations to conveniently vet their customers and fight against fraudulent activities.

The following are top 20 real-world blockchain applications across the industry according to CoinSwitch desk.

***Cybersecurity:***

*Guardtime* - a keyless signature which is currently used to secure the health records of one million Estonian citizens.

*REMME* - a decentralized authentication system which aims to replace logins and passwords with SSL certificates stored on a blockchain.

***Healthcare:***

*Gem* - working with the Centre for Disease Control(CDC) to put disease outbreak data onto a blockchain.

*Simply Vital Health* - ConnectingCare which tracks the health progress of patients, and Health Nexus, which aims to provide decentralized blockchain patient records.

***Banking and Financial:***

*Barclays* - Barclays has launched a number of technological initiatives which involves tracking financial transactions, combating fraud and compliance.

*Augur* - Allows the creation of blockchain-based predictions markets for the trading of derivatives and other financial instruments in a decentralized ecosystem.

***Manufacturing:***

*Provenance* - provide a blockchain-based provenance record of transparency within supply chains.

*TransActivgrid* - allowing members to locally produce and sell energy, with the goal of reducing costs involved in energy distribution.

***Government:***

*Dubai* - 30 government departments formed a committee dedicated to investigating opportunities across health records, shipping, business registration and preventing the spread of conflict diamonds.

*South Korea* - Samsung is creating blockchain solutions for the South Korean government which will be put into use in public safety and transport applications.

***Charity:***

*Bitgive* - aims at providing greater transparency to charity donations and reduce operation overhead.

***Retail:***

*OpenBazaar* - a decentralized market where no middleman is involved in trading of goods and services.

*Loyyal* - universal blockchain-based framework which aims at allowing customers to combine trade and loyalty reward points in newer ways so that retailers can offer more sophisticated loyalty packages.

***Real Estate:***

*Ubiquity* - a blockchain-driven system for tracking the complicated legal process which creates friction and expenses in real estate transfer.

***Transportation and Tourism:***

*Arcade City* - An application which aims at beating Uber at their own game by moving ride sharing and car hiring onto the blockchain.

*Webjet* - an online travel portal, which is developing a blockchain solution to allow stock of empty hotel rooms to be efficiently tracked and traded.

***Media:***

*Kodak* - developing a blockchain system for tracking intellectual property rights and payments to photographers.

*Ujo Music* - record and track royalties for musicians and allows them to create a record of ownership of their work.

***Smart Contracts:***

*Slock* - an Ethereum based internet-of-things (IoT) platform, which is using this application so that people can rent anything from bicycles to apartments by unlocking a smart lock after both parties agreed on the terms of the contract.

But if you are developing a blockchain-based application yourself, you'll probably need to provide support and maintenance for it. Research shows that the cost of support and maintenance for software is at least as great as the cost of developing it. This explains why software vendors charge so much for product support and maintenance. A good rule of thumb: If you develop an application that uses blockchains yourself, expect to pay at least as much throughout the application lifecycle as you do during development.

# BigchainDB

[BigchainDB](https://www.bigchaindb.com/) is a database that builds upon the strengths of a traditional NoSQL database, MongoDB, and augments it with the strengths of Blockchain. NoSQL databases are known for their speed, scalability, and flexibility yet susceptible to attacks which aim to change its data. Blockchain is renowned for its immutability, security, and decentralized control, but is often regarded as slow. For instance it can sometimes take 10 minutes for a transaction to be committed to a Bitcoin block. Databases running modern web apps need transaction times in milliseconds. By taking the best from both systems, the offspring of these two wedded technologies is better than its parents.

BigchainDB Features:

1. Decentralized Control
2. Transaction Immutability
3. Standard Queries
4. Fast Commits
5. Sybil Tolerance

## Decentralized Control

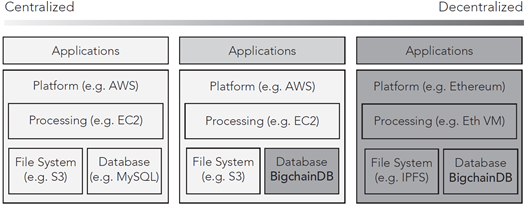
BigchainDB is designed to be [decentralized](http://docs.bigchaindb.com/en/latest/decentralized.html) and distributed but yet synchronized, which makes it ideal for multi-organizational business networks, business to business transactions, or simply for ensuring redundancy and high availability. It also encourages organizations to come out from behind their firewalls and to share data. BigChainDB is built upon a network of peer nodes, all of which contain an exact eventually consistent copy of the same database. The beauty of this design is that there is no one single point of failure, and no one single point of control.

Ideally each node in a BigChainDB network is controlled by a different entity. Even if the nodes are all within the same company, each node is better suited to be controlled by different divisions or different persons in the company. As in life, the strength of a BigChainDB network comes from its diversity. The greater the geographic, organizational or other types of diversity present in the network the more fault tolerant it will be.

There is no supervisor node that has can override or supersede the authority of any other node.. All nodes are equivalent peers running the same version of software and performing the same set of actions.

A collection of nodes in a BigChainDB implementation is called a Consortium. Consortium rules are set forth for each node to follow as a precondition for inclusion in the network, and to provide a tailored system of governance for the network. These rules are flexible and consequently implementation specific. For instance the rules for a B2B operation probably should be different than the rules for an asset tracking system.

As in traditional blockchain implementations like Bitcoin, BigchainDB relies on a consensus of the nodes agreeing on a new block before the new block will be committed to the cryptographic chain. Therefore, if someone hacked into a node, and tried to commit compromising data to the chain, the other nodes would reject it. A BigchainDB network can only be impacted if more than 1/3 of the nodes are compromised. If there is at least 2/3 consensus then the network will still allow a commit on the next block. This also means that even if 1/3 is affected, the the data would at least be safe from tampering unless 2/3 of the nodes were compromised.





## Transaction Immutability

Blockchains ledgers are generally considered immutable, unchanging over time. This is of course a generalization as nothing is impossible, it is just highly unlikely in the cryptographic, probabilistic sense of the word in blockchain. BigchainDB is immutable in the sense that the data is highly resistant to change whether that change be intentional tampering or unintentional through such things as natural disasters or server failures.

Blockchain immutability can be achieved in a number of ways, and some compromises over the original Bitcoin design needed to be made for the specific requirements of a database. For instance Bitcoin Then it uses a proof-of-work backed by a monetary incentive to make it very expensive for a hacker to try and surpass the honest nodes. This proof of work method however is slow, too slow to be used by a database, and so BigchainDB does not use it.

[Tendermint](https://www.tendermint.com/docs/introduction/what-is-tendermint.html) is a distributed Byzantine-fault tolerant state machine replication protocol. It is tolerant to failure of up to 1/3 of the nodes, it is eventually consistent as every node will see the same transaction log, and will compute the same state in isolation. BigchainDB cryptographically signs to verify ownership and hashes to detect tampering each transaction. Then BigchaingDB uses Tendermint to replicate and secure the its cryptographic transaction logs across a peer to peer network.

BigchainDB only exposes Create and Transfer API methods. There is no concept of deleting or updating a record. Even if the underlying MongoDB database in a node were to be modified with MongoDB native Update and Delete commands. The hashes would then be violated, and the other nodes would not accept the changes.



## Fast Commits

Commits on a network depend on the performance of the internode communication channels, the speed of the servers, and the number of nodes. However, commits are not limited to finding a complex proof of work hash. Thus, while numbers will vary, commits on a Tendermint based network like BigchainDB range from a few milliseconds to a few seconds, which is orders of magnitude faster than Bitcoin.

## Standard Queries

A users on BigchainDB can [query](https://docs.bigchaindb.com/en/latest/query.html) all stored data, all transactions, assets and metadata in a BigchainDB using the standard MongoDB query engine. Users can even use at least a subset of [SQL](https://studio3t.com/knowledge-base/articles/sql-query/) with the right tools and drivers.

## Sybil Tolerance

Blockchain networks like Bitcoin allow nodes to freely join the network. This opens up the possibility that a bad actor could inject enough nodes into the network to control the the balance of computing power, thus controlling the acceptance of valid blocks. This is referred to as a a Sybil attack. Bitcoin limits the possibility of a Sybil attacks by making sure the incentives for honest nodes outweigh the incentives for dishonest ones. Conversely, in a BigchainDB network, nodes must be added to the network by the governing organization, and so Sybil attacks are a non-issue.

# BigchainDB Demo

In this tutorial we cover:

* Creating User Identities
* Creating Digital Assets
* Assigning ownership to assets through a transaction
* Transferring assets between two identities
* Querying the blockchain for a record of these transactions

We are fast approaching a time as a society in which ownership of an asset like a bicycle or a car, or any other type of asset need not be represented by a signed piece of paper which can be lost stolen or forged. Ownership can be represented as a digital asset in a immutable cryptographically secured database in much more secure and verifiable manner than traditional means.

Our team has prepared a demonstration of a Simple BigchainDB network hosted on AWS. In this demonstration we take some liberties and we do not follow the best practice of diversifying the nodes as they are all going to be hosted on the same provider, AWS, and obviously controlled by the same users, us. However, these are reasonable simplifications for a development or demonstration environment.

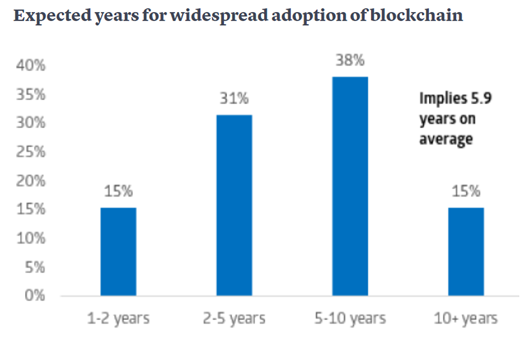
Details of the setup process are available in [BigChainDBProject\_AWSImplementation.pdf](file:///C:\dev\DS7349BlockChain\BigchainDB\BigChainDBProject_AWSImplementation.pdf). A IPython Juypter notebook, the project documentation is available in [BigChainDBProject\_PythonDriver.pdf](file:///C:\dev\DS7349BlockChain\BigChainDBProject_PythonDriver.pdf).

# Blockchain outlook

In theory, blockchain cuts costs and increases transaction efficiency because the technology eliminates the need for a third-party intermediary, such as a bank. Bitcoin is the first application of blockchain technology. Its price, and that of many other cryptocurrencies, skyrocketed, then becoming apparent to many that blockchain technology in the financial payment industry isn’t necessarily the best solution in the near term. It requires a mind shift, culture change, systematically reorganize the society we currently live in.

The application of blockchain has now shifted to various areas such as smart contracts, digital identity, assest tracking, clearing, and settlement. Some of the major factors driving the growth of the blockchain technology market are transparency and immutability, faster transactions, and reduced total cost of ownership. The blockchain technology also offers other key benefits such as trustless exchange, durability and reliability, and empowers the users to control all their information and transactions.

Several analysts expect blockchain will first prove its value in supply chain management, where Cisco and IBM are both working on applying the technology. Still, industry experts project on average that the technology will take 5.9 years to gain widespread adoption, according to a Cowen survey.



When it comes to proving the tangible benefits of the blockchain technology that many expect to change the world, it is still very early days and the earliest viable applications will likely be in supply chain management, rather than financial services, executives say.

As more organizations put their human and financial resources behind blockchain and come to better realize how it can improve their business processes and their bottom lines, we expect the technology to gain significant traction, as its cost savings, competitive advantages, and ROI benefits become more pronounced.

The view further down the road is an inspiring one. We see blockchain enabling a completely new level of information exchange both within—and across—industries. As connections are made between blockchain and other emerging technologies, particularly the cloud and automation, we see the potential for blockchain to help organizations create and realize new value for businesses beyond anything we can imagine with existing technologies.

References

1. https://blockgeeks.com/guides/what-is-blockchain-technology/ "Judd Bagley (2018 , Sep . ). What is Blockchain Technology? A Step-by-Step Guide For Beginners. Blockgeeks."
2. https://medium.com/blockchain-review/how-does-the-blockchain-work-for-dummies-explained-simply-9f94d386e093 "Collin Thompson 2016 Oct. How does the Blockchain Work? (Part 1). The Blockchain Review."
3. https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/ "Collin Thompson (2016 Oct.). Blockchains & Distributed Ledger Technologies. BlockchainHub."
4. https://web.archive.org/web/20140226205104/http://origin-www.computer.org/csdl/mags/co/1977/06/01646525.pdf "Exhaustive Cryptanalysis of the NBS Data Encryption Standard"
5. https://www.foo.be/docs/eff-des-cracker/book/crackingdessecre00elec.pdf "Electronic Frontier Foundation (1998), Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design, Electronic Frontier Foundation, ISBN ISBN: 1-56592-520-3"
6. https://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf "Diffie, Whitfield & Martin Hellman (Nov. 1976), "New Directions in Cryptography", IEEE Transactions on Information Theory IT-22: 644-654"
7. http://# "It was created by IBM's (International Business Machines) Walter Tuchman (1997). "A brief history of the data encryption standard". Internet besieged: countering cyberspace scofflaws. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA. pp. 275–280."
8. https://ia801307.us.archive.org/33/items/AmericanCryptologyDuringtheColdWar1945-1989BookIII\_RetrenchmentandReform1972-1980/American%20Cryptology%20During%20the%20Cold%20War%2C%201945-1989.%20Book%20III\_%20Retrenchment%20and%20Reform%2C%201972-1980.pdf "Thomas R. Johnson (2009-12-18). "American Cryptology during the Cold War, 1945-1989.Book III: Retrenchment and Reform, 1972-1980, page 232" (PDF)."
9. http://www.toad.com/gnu/cfp.talk.txt "John Gilmore (March 1991), Privacy, Technology, and the Open Society"
10. https://www.activism.net/cypherpunk/manifesto.html "A Cypherpunk's Manifesto"
11. https://ieeexplore.ieee.org/document/1055638 "Diffie, Whitfield; Hellman, Martin E. (November 1976). "New Directions in Cryptography" (PDF). IEEE Transactions on Information Theory. 22 (6): 644–654. doi:10.1109/TIT.1976.1055638. Archived (PDF) from the original on 2014-11-29."
12. https://patents.google.com/patent/US4200770 "Cryptographic apparatus and method"
13. https://www.gchq.gov.uk/sites/default/files/document\_files/CESG\_Research\_Report\_No\_3006\_0.pdf "The Possibility of Secure Non-Secret Digital Encryption"
14. http://en.citizendium.org/wiki/Cypherpunk/Citable\_Version "Cypherpunk/Citable Version"
15. https://cryptome.org/0001/assange-cpunks.htm "An Archive of Julian Assange's Cypherpunk Mailing List Posts"
16. https://www.bigchaindb.com/ "BigchainDB"
17. http://nakamotostudies.org/wp-content/uploads/2018/03/bitcoin.pdf "Bitcoin: A Peer-to-Peer Electronic Cash System"
18. \https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf "BigchainDB 2.0, The Blockchain Database"
19. https://www.tendermint.com/docs/introduction/what-is-tendermint.html "What is Tendermint?"
20. http://docs.bigchaindb.com/en/latest/decentralized.html "How BigchainDB is Decentralized"
21. https://docs.bigchaindb.com/en/latest/immutable.html "How BigchainDB is Immutable"
22. https://www.bigchaindb.com/developers/guide/key-concepts-of-bigchaindb/ "Key concepts of BigchainDB"
23. https://studio3t.com/knowledge-base/articles/sql-query/ "SQL Query"
24. https://docs.bigchaindb.com/en/latest/query.html "Querying BigchainDB"
25. https://github.com/bigchaindb/bigchaindb-driver "bigchaindb-driver"
26. https://www.python.org/ "Python"
27. https://www.mongodb.com/download-center "MongoDB"
28. https://mmonit.com/monit/ "Monit"
29. http://www.allitebooks.com/cryptography-and-network-security-5th-edition/ "CRYPTOGRAPHY and NETWORK SECURITY PRINCIPLES AND PRACTICE FIFTH EDITION, pg. 94"

1. “This work was submitted as part of the MSDS 7349 Section 404 “Data and Network Security” course at Southern Methodist University as part of the Spring 2019 semester.

   YuMei Bennett (e-mail: yumei.bennett@oracle.com), Daniel Byrne (byrned@mail.smu.edu), Benjamin Wilke (e-mail: bwilke@mail.smu.edu), and Brian Coari (e-mail: bcoari@mail.smu.edu) are graduate students in the Data Science program at Southern Methodist University, PO Box 750100, Dallas, TX 75275 [↑](#footnote-ref-1)