Bitcoin, Blockchain, and Cryptocurrencies

MGT 4074 Fintech & Crypto Tokens, Fall 2025

Joseph Hall

Georgia Institute of Technology

September 10, 2025

Roadmap

Roadmap

Brazil Abandons Blockchain for CBDC

Forbes:

> *Brazil's Central Bank is abandoning the blockchain component of Drex, its ambitious central bank digital currency project.*

Brazil Abandons Blockchain for CBDC

Forbes:

> *Brazil's Central Bank is abandoning the blockchain component of Drex, its ambitious central bank digital currency project.*
>
> *The project's original vision was a two-tiered structure. The first monetary layer was to be an environment exclusively for transactions among authorized participants - specifically regulated financial institutions running nodes on the Central Bank-controlled network. The second would be tokenized bank deposits issued by regulated institutions to end customers.*

Discussion

Q: What does CBDC stand for?

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"
A: The full history of transactions using the CBDC would be cryptographically encoded in a series of "blocks", "chained" together with each block starting with a reference to the previous block.

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"
A: The full history of transactions using the CBDC would be cryptographically encoded in a series of "blocks", "chained" together with each block starting with a reference to the previous block.
Q: What are the pros and cons of using a Blockchain, instead of a simple central database, to host CBDC?

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"
A: The full history of transactions using the CBDC would be cryptographically encoded in a series of "blocks", "chained" together with each block starting with a reference to the previous block.
Q: What are the pros and cons of using a Blockchain, instead of a simple central database, to host CBDC?
Benefits of Blockchain: Decentralization, verifiability, trustlessness

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"
A: The full history of transactions using the CBDC would be cryptographically encoded in a series of "blocks", "chained" together with each block starting with a reference to the previous block.
Q: What are the pros and cons of using a Blockchain, instead of a simple central database, to host CBDC?
Benefits of Blockchain: Decentralization, verifiability, trustlessness
Drawbacks of Blockchain: Computing power, privacy

Discussion

Q: What does CBDC stand for? **A:** Central Bank Digital Currency
Q: What does it mean for a CBDC to have a "blockchain component?"
A: The full history of transactions using the CBDC would be cryptographically encoded in a series of "blocks", "chained" together with each block starting with a reference to the previous block.
Q: What are the pros and cons of using a Blockchain, instead of a simple central database, to host CBDC?
Benefits of Blockchain: Decentralization, verifiability, trustlessness
Drawbacks of Blockchain: Computing power, privacy
TL;DR: Brazil has decided not to record transactions of the "Digital Real" onto a blockchain. Instead, it will utilize a central database.

Roadmap

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand
- Why am I teaching it then?

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand
- Why am I teaching it then?
  - It's fun

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand
- Why am I teaching it then?
  - It's fun
  - The better you understand the math, the better you'll understand what is and is not technologically possible

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand
- Why am I teaching it then?
  - It's fun
  - The better you understand the math, the better you'll understand what is and is not technologically possible
  - You might be better than me at math

Be Not Afraid

- Three math topics:
  - Hash functions
  - Digital signature schemes
  - Merkle Trees
- We do need to do some math to understand what is really going on here
  - You are not expected to fully understand
  - I don't completely understand
- Why am I teaching it then?
  - It's fun
  - The better you understand the math, the better you'll understand what is and is not technologically possible
  - You might be better than me at math
  - If a "technical person" knows this stuff worse than you, question their credentials

A Philosophical Question...

Does true randomness exist in our physical universe? Or is everything predetermined?

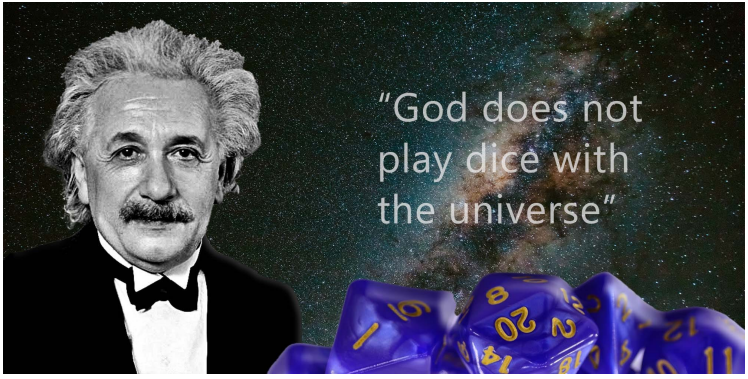
A Philosophical Question...

Does true randomness exist in our physical universe? Or is everything predetermined?

- If you think everything is predetermined, and we fully figure out the laws of physics, doesn't that mean we can predict the future?

A Philosophical Question...

Does true randomness exist in our physical universe? Or is everything predetermined?



- If you think everything is predetermined, and we fully figure out the laws of physics, doesn't that mean we can predict the future?
- Anyone heard of the "Three Body Problem"?

Chaos

- Nature (and math) has many processes that are **chaotic**: deterministic, but unpredictable
  - Three-body problem
  - Physical dice, coins
  - Which numbers are prime?

Chaos

- Nature (and math) has many processes that are **chaotic**: deterministic, but unpredictable
  - Three-body problem
  - Physical dice, coins
  - Which numbers are prime?
- "Deterministically" driven by physics laws, but very sensitive to initial conditions, so from our perspective, as good as random

Chaos

- Nature (and math) has many processes that are **chaotic**: deterministic, but unpredictable
  - Three-body problem
  - Physical dice, coins
  - Which numbers are prime?
- "Deterministically" driven by physics laws, but very sensitive to initial conditions, so from our perspective, as good as random
  - We can simulate these (but always *some* error)

Chaos

- Nature (and math) has many processes that are **chaotic**: deterministic, but unpredictable
  - Three-body problem
  - Physical dice, coins
  - Which numbers are prime?
- "Deterministically" driven by physics laws, but very sensitive to initial conditions, so from our perspective, as good as random
  - We can simulate these (but always *some* error)
  - But small changes in initial conditions change outcomes substantially
  - "When a butterfly flaps its wings..."

Chaos

- Nature (and math) has many processes that are **chaotic**: deterministic, but unpredictable
  - Three-body problem
  - Physical dice, coins
  - Which numbers are prime?
- "Deterministically" driven by physics laws, but very sensitive to initial conditions, so from our perspective, as good as random
  - We can simulate these (but always *some* error)
  - But small changes in initial conditions change outcomes substantially
  - "When a butterfly flaps its wings..."
- Hash functions are **chaotic**: deterministic but unpredictable
  - **Deterministic**: The hash of a number never changes!
  - **Unpredictable**: The hashes of two nearby numbers are *very* far away, and there is *no pattern* to how far

A Chaotic Process

A Chaotic Process

Why does Cloudflare use lava lamps to help with encryption?

Randomness is extremely important for secure encryption. Each new key that a computer uses to encrypt data must be truly random, so that an attacker won't be able to figure out the key and decrypt the data. However, computers are designed to provide predictable, logical outputs based on a given input. They aren't designed to produce the random data needed for creating unpredictable encryption keys.

To produce the unpredictable, chaotic data necessary for strong encryption, a computer must have a source of random data. The "real world" turns out to be a great source for randomness, because events in the physical world are unpredictable.

Chaotic Functions

- A **hash function** takes some input string, and "chaotically" spits out an output string. Demo
- Servers (to which you log in) store hashes, not passwords!

Chaotic Functions

- A **hash function** takes some input string, and "chaotically" spits out an output string. Demo
- Servers (to which you log in) store hashes, not passwords!
- Intuitively, "chaos" gives us some very nice properties
  - **One-way:** Can calculate outcomes, but can't back out initial conditions
  - **Collision Resistance:** Lots of outcomes are possible, so two initial conditions rarely result in same outcome (a "collision")

Chaotic Functions

- A **hash function** takes some input string, and "chaotically" spits out an output string. Demo
- Servers (to which you log in) store hashes, not passwords!
- Intuitively, "chaos" gives us some very nice properties
  - **One-way:** Can calculate outcomes, but can't back out initial conditions
  - **Collision Resistance:** Lots of outcomes are possible, so two initial conditions rarely result in same outcome (a "collision")
- Practically: do some silly deterministic operation, a lot of times, until all the input information is scrambled:
  - Convert text to binary numbers, 1101010...
  - Flip the 4th, 13th, 18th from 1 to 0, or 0 to 1...
  - Make 5th, 17th, and 12th number vote on whether 3rd is 0 or 1...
  - Move 1st to 2nd, 2nd to 3rd...64th to 1st...

Chaotic Functions

- A **hash function** takes some input string, and "chaotically" spits out an output string. Demo
- Servers (to which you log in) store hashes, not passwords!
- Intuitively, "chaos" gives us some very nice properties
  - **One-way:** Can calculate outcomes, but can't back out initial conditions
  - **Collision Resistance:** Lots of outcomes are possible, so two initial conditions rarely result in same outcome (a "collision")
- Practically: do some silly deterministic operation, a lot of times, until all the input information is scrambled:
  - Convert text to binary numbers, 1101010...
  - Flip the 4th, 13th, 18th from 1 to 0, or 0 to 1...
  - Make 5th, 17th, and 12th number vote on whether 3rd is 0 or 1...
  - Move 1st to 2nd, 2nd to 3rd...64th to 1st...
- Tl;dr, record your kid doing random stuff, and repeat it a lot of times

Hash Functions

A cryptographic hash function is a mapping $H : M \to D$ from a set M of possible messages of any length to a set D of digests, for example the set of strings of 0s and 1s of a fixed length, such as 256. See Ramzan (2012).

[1] digital signature schemes do preserve some properties of x as we will see

Hash Functions

A cryptographic hash function is a mapping $H : M \to D$ from a set M of possible messages of any length to a set D of digests, for example the set of strings of 0s and 1s of a fixed length, such as 256. See Ramzan (2012).

For usefulness, a cryptographic hash function H should be easily computable and, among other properties, be:

1. **Collision resistant**, meaning that H is "nearly injective." That is, it should be extremely rare to have $H(x) = H(y)$ unless $x = y$.

[1] digital signature schemes do preserve some properties of x as we will see

Hash Functions

A cryptographic hash function is a mapping $H : M \to D$ from a set M of possible messages of any length to a set D of digests, for example the set of strings of 0s and 1s of a fixed length, such as 256. See Ramzan (2012).

For usefulness, a cryptographic hash function H should be easily computable and, among other properties, be:

1. **Collision resistant**, meaning that H is "nearly injective." That is, it should be extremely rare to have $H(x) = H(y)$ unless $x = y$.
2. **Difficult to invert**. That is, it should be extremely hard to guess x, or properties of x[1], from $H(x)$.

[1] digital signature schemes do preserve some properties of x as we will see

Hash Functions

A cryptographic hash function is a mapping $H : M \to D$ from a set M of possible messages of any length to a set D of digests, for example the set of strings of 0s and 1s of a fixed length, such as 256. See Ramzan (2012).

For usefulness, a cryptographic hash function H should be easily computable and, among other properties, be:

1. **Collision resistant**, meaning that H is "nearly injective." That is, it should be extremely rare to have $H(x) = H(y)$ unless $x = y$.
2. **Difficult to invert**. That is, it should be extremely hard to guess x, or properties of x[1], from $H(x)$.
3. **Extremely sensitive**. Even small changes in the message should usually produce large changes in the digest.

[1] digital signature schemes do preserve some properties of x as we will see

Hash Functions

A cryptographic hash function is a mapping $H : M \to D$ from a set M of possible messages of any length to a set D of digests, for example the set of strings of 0s and 1s of a fixed length, such as 256. See Ramzan (2012).

For usefulness, a cryptographic hash function H should be easily computable and, among other properties, be:

1. **Collision resistant**, meaning that H is "nearly injective." That is, it should be extremely rare to have $H(x) = H(y)$ unless $x = y$.
2. **Difficult to invert**. That is, it should be extremely hard to guess x, or properties of x[1], from $H(x)$.
3. **Extremely sensitive**. Even small changes in the message should usually produce large changes in the digest.

In **digital signature schemes** (as we will see), a "digest" is a **signed message**.

[1] digital signature schemes do preserve some properties of x as we will see

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Answer: Yes, it is. Observe that $H(x) - H(y) = x - y$ in this case, so if $x \neq y$, then $H(x) \neq H(y)$

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Answer: Yes, it is. Observe that $H(x) - H(y) = x - y$ in this case, so if $x \neq y$, then $H(x) \neq H(y)$

Question: is $H(.)$ difficult to invert?

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Answer: Yes, it is. Observe that $H(x) - H(y) = x - y$ in this case, so if $x \neq y$, then $H(x) \neq H(y)$

Question: is $H(.)$ difficult to invert?

Answer: No, it is easy to invert. In this case $H^{-1}(x) = x + 1$

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Answer: Yes, it is. Observe that $H(x) - H(y) = x - y$ in this case, so if $x \neq y$, then $H(x) \neq H(y)$

Question: is $H(.)$ difficult to invert?

Answer: No, it is easy to invert. In this case $H^{-1}(x) = x + 1$

Question: is $H(.)$ a useful cryptographic **hash function**?

Collision Resistance and Invertability

Question: Suppose a function H maps the integers 1, 2, 3, 4, 5, 6 to the integers 0, 1, 2, 3, 4, 5 by letting H(x) = x - 1. For example, H(5) = 4. Is H collision resistant?

Answer: Yes, it is. Observe that $H(x) - H(y) = x - y$ in this case, so if $x \neq y$, then $H(x) \neq H(y)$

Question: is $H(.)$ difficult to invert?

Answer: No, it is easy to invert. In this case $H^{-1}(x) = x + 1$

Question: is $H(.)$ a useful cryptographic **hash function**?

Answer: No, hash functions must be hard to invert.

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information
  - Easy to compute $H(x)$ given $H(\cdot)$ and x

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information
  - Easy to compute $H(x)$ given $H(\cdot)$ and x
  - If someone changes x to $\tilde{x}$, even by one letter, $H(\tilde{x})$ will be very different!
  - Used in checksums for file downloading

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information
  - Easy to compute $H(x)$ given $H(\cdot)$ and x
  - If someone changes x to $\tilde{x}$, even by one letter, $H(\tilde{x})$ will be very different!
  - Used in checksums for file downloading
- Application: **Mining and proof-of-work**
  - No way to discover x given $H(x)$ beside **brute force**!

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information
  - Easy to compute $H(x)$ given $H(\cdot)$ and x
  - If someone changes x to $\tilde{x}$, even by one letter, $H(\tilde{x})$ will be very different!
  - Used in checksums for file downloading
- Application: **Mining and proof-of-work**
  - No way to discover x given $H(x)$ beside **brute force**!
  - The Secure Hash Algorithm 1 ("**SHA1**") takes between 2^{63} - 2^{80} evaluations to find a collision (x, y such that $H(x) = H(y)$) (depending on how clever you are)

Hash Functions

In blockchains, hash functions have two main uses:

- By publishing $y = H(x)$, anyone can check that they know the real x without x ever becoming public information
  - Easy to compute $H(x)$ given $H(\cdot)$ and x
  - If someone changes x to $\tilde{x}$, even by one letter, $H(\tilde{x})$ will be very different!
  - Used in checksums for file downloading
- Application: **Mining and proof-of-work**
  - No way to discover x given $H(x)$ beside **brute force**!
  - The Secure Hash Algorithm 1 ("**SHA1**") takes between 2^{63} - 2^{80} evaluations to find a collision (x, y such that $H(x) = H(y)$) (depending on how clever you are)
  - Bitcoins are mined by brute-forcing hashes to find xs that produce $H(x)$ with leading zeroes.

Modular Arithmetic

Powers of 3	3^1	3^2	3^3	3^4	3^5	3^6	3^7	3^8	3^9
Value	3	9	27	81	243	729	2187	6561	19683
Closest multiple of 7	0	7	21	77	238	728	2184	6559	19677
Remainder div. 7	3	2	6	4	5	1	3	2	6

- We use $x \mod y$ to mean, "remainder of x when divided by y", like:

$$81 \mod 7 = 4$$

- In python, $x \mod y$ is just $x \,\%\, y$
- In R, it's $x \,\%\%\, y$

How do I prove that I signed something?

Requirements:

- JPH has a public key everyone knows
- JPH has a private key only JPH knows
  - **Non-invertability** means the public key doesn't reveal the private key

How do I prove that I signed something?

Requirements:

- JPH has a public key everyone knows
- JPH has a private key only JPH knows
  - **Non-invertability** means the public key doesn't reveal the private key
- JPH can "sign" messages: use the private key to alter the message in a way that proves JPH created his public key
- Public key allows **verification**, but not **signature production**

How do I prove that I signed something?

Requirements:

- JPH has a public key everyone knows
- JPH has a private key only JPH knows
  - **Non-invertability** means the public key doesn't reveal the private key
- JPH can "sign" messages: use the private key to alter the message in a way that proves JPH created his public key
- Public key allows **verification**, but not **signature production**
  - Using public key, anyone can verify signature correctness: JPH used his private key to sign the message

How do I prove that I signed something?

Requirements:

- JPH has a public key everyone knows
- JPH has a private key only JPH knows
  - **Non-invertability** means the public key doesn't reveal the private key
- JPH can "sign" messages: use the private key to alter the message in a way that proves JPH created his public key
- Public key allows **verification**, but not **signature production**
  - Using public key, anyone can verify signature correctness: JPH used his private key to sign the message
  - But, using the public key, can't generate new valid signatures

Digital Signatures, Formally

A **Digital Signature Scheme** is a combination of Three Functions:

- A function to generate **Public Keys**: $PrK \Rightarrow PuK$

Digital Signatures, Formally

A **Digital Signature Scheme** is a combination of Three Functions:

- A function to generate **Public Keys**: $PrK \Rightarrow PuK$
- A function to generate **Signed Messages**: $(PrK, M) \Rightarrow M'$

Digital Signatures, Formally

A **Digital Signature Scheme** is a combination of Three Functions:

- A function to generate **Public Keys**: $PrK \Rightarrow PuK$
- A function to generate **Signed Messages**: $(PrK, M) \Rightarrow M'$
- A function to **Verify** signed messages: $(PuK, M, M') \Rightarrow \{0, 1\}$

Digital Signatures, Formally

A **Digital Signature Scheme** is a combination of Three Functions:

- A function to generate **Public Keys**: $PrK \Rightarrow PuK$
- A function to generate **Signed Messages**: $(PrK, M) \Rightarrow M'$
- A function to **Verify** signed messages: $(PuK, M, M') \Rightarrow \{0, 1\}$

These functions ($\Rightarrow$) are one-way, chaotic functions!

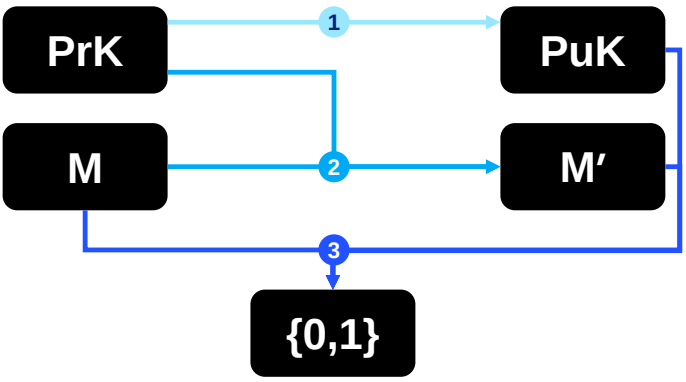
Digital Signatures, Formally

A **Digital Signature Scheme** is a combination of Three Functions:

- A function to generate **Public Keys**: $PrK \Rightarrow PuK$
- A function to generate **Signed Messages**: $(PrK, M) \Rightarrow M'$
- A function to **Verify** signed messages: $(PuK, M, M') \Rightarrow \{0, 1\}$

These functions ($\Rightarrow$) are one-way, chaotic functions!
In Bitcoin, these "messages" encode a list of transfers of blocks from one owner to another!

Digital Signature Schemes, Visually



Ok, but how do these signatures actually work?

ECE 6208 teaches this; it is beyond my own understanding.

The easiest to implement from scratch, that I'm aware of, is called ElGamal.

Ok, but how do these signatures actually work?

ECE 6208 teaches this; it is beyond my own understanding.

The easiest to implement from scratch, that I'm aware of, is called ElGamal. ("The Beauty" in Arabic)

In the News: Stablecoin Price War

From Bloomberg's Matt Levine:

> I have always thought that issuing stablecoins is a great business. The way it works is:
>
> 1. People give you dollars in exchange for cryptographic tokens that you make up.
> 2. You invest their dollars in safe short-term assets, like a government money market fund yielding about 4%.
> 3. You promise to exchange their tokens back into dollars back whenever they want.
> 4. Without interest.
>
> It's like banking, but without interest or branches or credit risk: Instead of taking people's money and using it to make loans, you take their money and park it in Treasury bills and keep the interest for yourself. Your costs are low and your revenue is high.

In the News: Stablecoin Price War

From Bloomberg's Matt Levine:

> I have always thought that issuing stablecoins is a great business. The way it works is:
>
> 1. People give you dollars in exchange for cryptographic tokens that you make up.
> 2. You invest their dollars in safe short-term assets, like a government money market fund yielding about 4%.
> 3. You promise to exchange their tokens back into dollars back whenever they want.
> 4. Without interest.
>
> It's like banking, but without interest or branches or credit risk: Instead of taking people's money and using it to make loans, you take their money and park it in Treasury bills and keep the interest for yourself. Your costs are low and your revenue is high.

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?
A: The GENIUS Act prohibits interest on stablecoins!

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?
A: The GENIUS Act prohibits interest on stablecoins!
Q: So what do they do?

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?
A: The GENIUS Act prohibits interest on stablecoins!
Q: So what do they do?
A: They pay a fee to exchanges to make their stablecoin the native stablecoin of the exchanges. The exchanges then rebate that fee to the end users, running around the GENIUS Act regulation.

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?
A: The GENIUS Act prohibits interest on stablecoins!
Q: So what do they do?
A: They pay a fee to exchanges to make their stablecoin the native stablecoin of the exchanges. The exchanges then rebate that fee to the end users, running around the GENIUS Act regulation.
Q: Why don't the regulators like this outcome?

In the News: Stablecoin Price War

Q: What incentives does this create for stablecoin issuers?
A: Incentives to pay interest and attract more deposits
Q: What is the problem with this plan?
A: The GENIUS Act prohibits interest on stablecoins!
Q: So what do they do?
A: They pay a fee to exchanges to make their stablecoin the native stablecoin of the exchanges. The exchanges then rebate that fee to the end users, running around the GENIUS Act regulation.
Q: Why don't the regulators like this outcome?
A: If deposits flow out of banks and into stablecoins, we will lose the liquidity-transforming, loan-supplying function of banks!

In the News: Stablecoin Price War

So where does this leave us?

> That is controversial. The Financial Times reported last month that "banking lobbies including the American Bankers Association, the Bank Policy Institute and the Consumer Bankers Association last week warned lawmakers of a 'loophole' in regulation that will let some crypto exchanges indirectly pay interest to stablecoin holders," which might "spark mass deposit outflows ... if customers choose to earn yield by holding stablecoins at crypto exchanges rather than coins or cash dollars at banks."
>
> But it also means that issuing stablecoins is not as great a business as I once thought. It *was*, when there were only a few stablecoins and the stablecoins acquired their network effects naturally. But in the long run, to be a competitive stablecoin, you might have to pay most or all of your interest income to crypto exchanges, and the exchanges might have to pass most or all of it back to the customers. "You hold people's dollars for them, you collect interest, and you don't give them any of it" really is a good business model, but it's too good to last.

In the News: Stablecoin Price War

So where does this leave us?

> That is controversial. The Financial Times reported last month that "banking lobbies including the American Bankers Association, the Bank Policy Institute and the Consumer Bankers Association last week warned lawmakers of a 'loophole' in regulation that will let some crypto exchanges indirectly pay interest to stablecoin holders," which might "spark mass deposit outflows ... if customers choose to earn yield by holding stablecoins at crypto exchanges rather than coins or cash dollars at banks."
>
> But it also means that issuing stablecoins is not as great a business as I once thought. It *was*, when there were only a few stablecoins and the stablecoins acquired their network effects naturally. But in the long run, to be a competitive stablecoin, you might have to pay most or all of your interest income to crypto exchanges, and the exchanges might have to pass most or all of it back to the customers. "You hold people's dollars for them, you collect interest, and you don't give them any of it" really is a good business model, but it's too good to last.

Uses of Merkle Trees

A Merkle Tree is a way of efficiently storing the required information to ensure that a set of documents or messages have not been tampered with or corrupted.

Uses of Merkle Trees

A Merkle Tree is a way of efficiently storing the required information to ensure that a set of documents or messages have not been tampered with or corrupted.

Merkle Trees are used in many applications to solve problems such as:

- Maintaining integrity of files stored on Dropbox.com, or *file outsourcing*
- Proving a transaction between Alice and Bob occurred, or *set membership*
- Transmitting files over unreliable channels, or *anti-entropy*

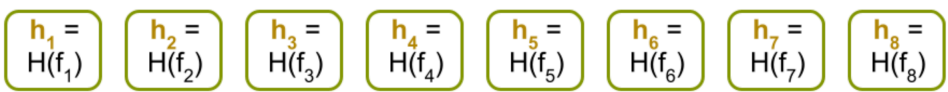
Construciton of Merkle Trees

Start with, say, 8 files $(f_1 ... f_8)$, and a hash function $H(.)$

Construciton of Merkle Trees

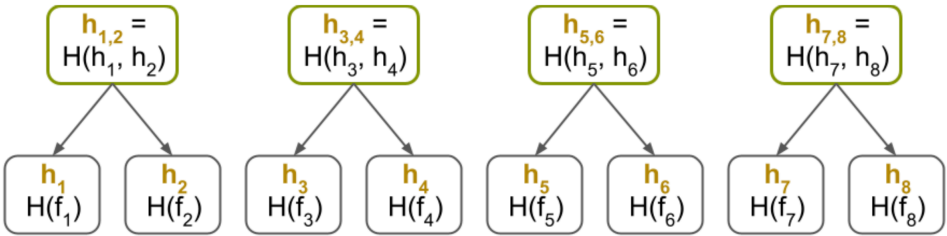
Start with, say, 8 files $(f_1 ... f_8)$, and a hash function $H(.)$

Start by hashing each file $H(f_i)$:



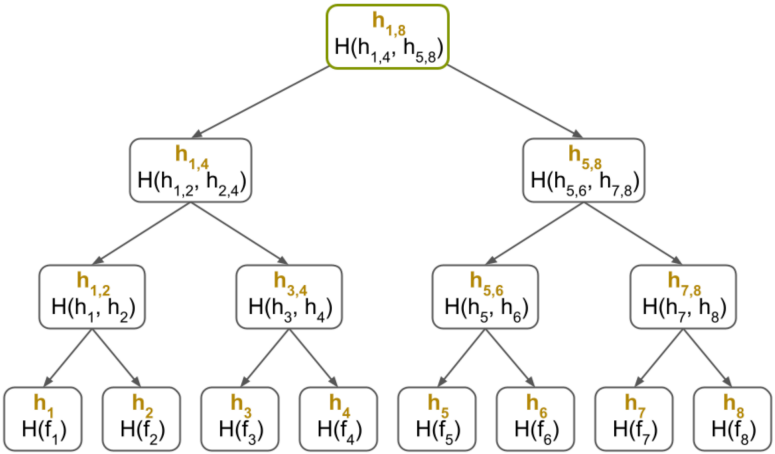
Construction of Merkle Trees

Then, create the second layer of the tree by hashing adjacent nodes together (this can be as simple as concatenating them!)



Construction of Merkle Trees

Continue until you have created an entire "tree":



Construction of Merkle Trees

Question: Why is this tree upside-down?

Construction of Merkle Trees

Question: Why is this tree upside-down?
Answer: Because Computer Scientists have never gone outside, a necessary prerequisite to see a real tree

Verification with Merkle Trees

Upload your files $(f_1 ... f_8)$ **and** your Merkle Tree to Dropbox, storing only the **Merkle Root** $h_{1,8}$ locally.

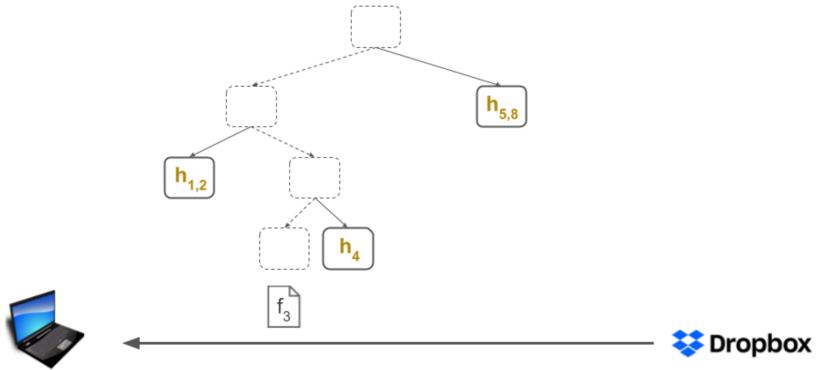
Verification with Merkle Trees

Upload your files $(f_1 ... f_8)$ **and** your Merkle Tree to Dropbox, storing only the **Merkle Root** $h_{1,8}$ locally.

Later, ask for the file you want, say f_3, but make sure to also ask for the **Merkle Proof**: the hashes needed to compute $h_{1,8}$ from the desired **Merkle Leaf** (f_3)

Verification with Merkle Trees

The **Merkle Proof** lets you verify that you downloaded the real, original f_3:



Advantages of Merkle Trees

Question: Why not just store the hash of every file? Wouldn't that be simpler?

Advantages of Merkle Trees

Question: Why not just store the hash of every file? Wouldn't that be simpler?
Answer: That would require storing 1 hash per document, so your local storage would scale linearly with number of files. The Merkle Tree requires storing **only** the Merkle Root, a single hash, **for any number of files**!

Advantages of Merkle Trees

Question: Why not just store the hash of every file? Wouldn't that be simpler?
Answer: That would require storing 1 hash per document, so your local storage would scale linearly with number of files. The Merkle Tree requires storing **only** the Merkle Root, a single hash, **for any number of files**!

Question: The server stores (h_4, $h_{1,2}$, and $h_{5,8}$) for you. Can't the server modify these to make the Merkle Proof appear true, even if f_3 is changed to $\tilde{f}_3$?

Advantages of Merkle Trees

Question: Why not just store the hash of every file? Wouldn't that be simpler?
Answer: That would require storing 1 hash per document, so your local storage would scale linearly with number of files. The Merkle Tree requires storing **only** the Merkle Root, a single hash, **for any number of files**!

Question: The server stores (h_4, $h_{1,2}$, and $h_{5,8}$) for you. Can't the server modify these to make the Merkle Proof appear true, even if f_3 is changed to $\tilde{f}_3$?
Answer: The cryptographic property of **Non-Invertability** ensures that it is prohibitively difficult to create a false Merkle proof.

Advantages of Merkle Trees

Question: Why not just store the hash of every file? Wouldn't that be simpler?
Answer: That would require storing 1 hash per document, so your local storage would scale linearly with number of files. The Merkle Tree requires storing **only** the Merkle Root, a single hash, **for any number of files**!

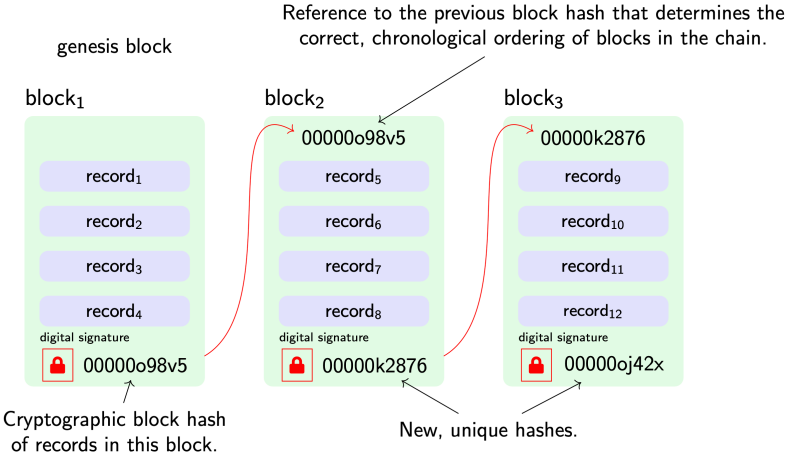
Question: The server stores (h_4, $h_{1,2}$, and $h_{5,8}$) for you. Can't the server modify these to make the Merkle Proof appear true, even if f_3 is changed to $\tilde{f}_3$?
Answer: The cryptographic property of **Non-Invertability** ensures that it is prohibitively difficult to create a false Merkle proof.

For other (analogous) uses of Merkle Trees including Transaction Verification and Transmitting files robustly over error-prone communication channels, see Tomescu (2020)

Roadmap

What is a Blockchain?

A **Blockchain** is a list of records that are linked together by hash functions



A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?
  - No: I don't have the private keys to create messages that can be verified as true.

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?
  - No: I don't have the private keys to create messages that can be verified as true.
- Cancel or revert messages?

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?
  - No: I don't have the private keys to create messages that can be verified as true.
- Cancel or revert messages?
  - Yes: I can refuse to include a new block or roll back the blockchain to a previous state.

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?
  - No: I don't have the private keys to create messages that can be verified as true.
- Cancel or revert messages?
  - Yes: I can refuse to include a new block or roll back the blockchain to a previous state.
- Selectively cancel three messages ago, but not the last two?

A Centralized Blockchain

Say that I, JPH, own and operate a private server which records a Blockchain. I don't know the users' private keys, but I control everything that is written on the blockchain. Can I:

- Create false messages?
  - No: I don't have the private keys to create messages that can be verified as true.
- Cancel or revert messages?
  - Yes: I can refuse to include a new block or roll back the blockchain to a previous state.
- Selectively cancel three messages ago, but not the last two?
  - No: the hashes of the last two messages would be wrong if the middle hash is missing.

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

When are these better than just a spreadsheet?

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

When are these better than just a spreadsheet?

- Often they aren't: Chase, VISA, Zelle, and Federal Reserves are all just spreadsheets at the end of the day

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

When are these better than just a spreadsheet?

- Often they aren't: Chase, VISA, Zelle, and Federal Reserves are all just spreadsheets at the end of the day
- Centralized Blockchains add an extra level of security: create a transaction history that can't be arbitrarily tampered with – only rolled back entirely or stopped pre-emptively.

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

When are these better than just a spreadsheet?

- Often they aren't: Chase, VISA, Zelle, and Federal Reserves are all just spreadsheets at the end of the day
- Centralized Blockchains add an extra level of security: create a transaction history that can't be arbitrarily tampered with – only rolled back entirely or stopped pre-emptively.
- That's still a lot of power in the admin's hands!

A Centralized Blockchain

Centralized Blockchains are in fact widely-used:

- Binance Smart Chain
- Coinbase Base
- Ripple XRP

When are these better than just a spreadsheet?

- Often they aren't: Chase, VISA, Zelle, and Federal Reserves are all just spreadsheets at the end of the day
- Centralized Blockchains add an extra level of security: create a transaction history that can't be arbitrarily tampered with – only rolled back entirely or stopped pre-emptively.
- That's still a lot of power in the admin's hands!

Can we take away even more power from the admin? Can we make a set of rules to agree on a shared history of transactions without needing to trust anybody?

A Decentralized Blockchain

- A natural, "democratic" rule would be to have multiple admins taking turns adding transactions to the chain

A Decentralized Blockchain

- A natural, "democratic" rule would be to have multiple admins taking turns adding transactions to the chain
- How do we prevent one admin from just pretending to be multiple admins?

A Decentralized Blockchain

- A natural, "democratic" rule would be to have multiple admins taking turns adding transactions to the chain
- How do we prevent one admin from just pretending to be multiple admins?
- Needs to be some "cost"/"barrier" to becoming an admin

A Decentralized Blockchain

- A natural, "democratic" rule would be to have multiple admins taking turns adding transactions to the chain
- How do we prevent one admin from just pretending to be multiple admins?
- Needs to be some "cost"/"barrier" to becoming an admin
- Bitcoin solves this through **proof of work**

Hash Puzzles

- The amount of computational power in the world is finite!
- Let's try to find a number whose SHA256 hash has a 0 at the beginning...

Hash Puzzles

- The amount of computational power in the world is finite!
- Let's try to find a number whose SHA256 hash has a 0 at the beginning...
- Why is this a nice puzzle? Like a PhD
  1. Hard to do
  2. Proves that you did something hard to do

Hash Puzzles

- The amount of computational power in the world is finite!
- Let's try to find a number whose SHA256 hash has a 0 at the beginning...
- Why is this a nice puzzle? Like a PhD
  1. Hard to do
  2. Proves that you did something hard to do
- With puzzle solution in hand, you become the admin!
- **Incentives:** solving the puzzle gives you some BTC as a block reward
- **Consensus:** everyone works on the longest chain

How to Prevent Doublespending?

- I may want to spend some money to buy a coffee, then "unspend" it (but keep my coffee!)
- Can't just selectively delete the transaction from history (why?)

How to Prevent Doublespending?

- I may want to spend some money to buy a coffee, then "unspend" it (but keep my coffee!)
- Can't just selectively delete the transaction from history (why?)
  - Hashes verify integrity of history!
- But...could go back in time, then create a parallel universe (a fork in the chain) where I never spent the money
  - This would be called a doublespend

Turning Back Time...

- Anyone can try, since anyone can try to mine and become admin
- When would I succeed?

Turning Back Time...

- Anyone can try, since anyone can try to mine and become admin
- When would I succeed?
  - Because everyone uses the **longest chain**, I must be able to hash faster than **everyone else combined**
  - Control 51% of the computing power in the Bitcoin ecosystem!
- For more details, see Stanford CS251 slides

In the News: Google's Agent Payment Protocol

In the News: Google's Agent Payment Protocol

How it works: Establishing trust via mandates and verifiable credentials

AP2 builds trust by using Mandates—tamper-proof, cryptographically-signed digital contracts that serve as verifiable proof of a user's instructions. These mandates are signed by verifiable credentials (VCs) and act as the foundational evidence for every transaction.

Mandates address the two primary ways a user will shop with an agent:

- **Real-time purchases (human present)**: When you ask an agent, "Find me new white running shoes," your request is captured in an initial **Intent Mandate**. This provides the auditable context for the entire interaction in a transaction process. After the agent presents a cart with the shoes you want, your approval signs a **Cart Mandate**. This is a critical step that creates a secure, unchangeable record of the exact items and price, ensuring what you see is what you pay for.
- **Delegated tasks (human not present)**: When you delegate a task like, "Buy concert tickets the moment they go on sale," you sign a detailed **Intent Mandate** upfront. This mandate specifies the rules of engagement—price limits, timing, and other conditions. It serves as verifiable, pre-authorized proof that can allow the agent to automatically generate a **Cart Mandate** on your behalf once your precise conditions are met.

In both scenarios, this chain of evidence culminates in securely linking your payment method to the verified contents of the Cart Mandate. This complete sequence—from intent, to cart, to payment—creates a non-repudiable audit trail that answers the critical questions of authorization and authenticity, providing a clear foundation for accountability.

The Money Pot Game

- Two options:
  1. Get $1 for sure
  2. Fight for a $10 pot, which is equally divided between everyone who picks this option

The Money Pot Game

- Two options:
  1. Get $1 for sure
  2. Fight for a $10 pot, which is equally divided between everyone who picks this option
- Which do you pick?

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"
- Similar settings: what's the equilibrium condition?
  - Firms competing for profits?

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"
- Similar settings: what's the equilibrium condition?
  - Firms competing for profits?
  - Going to the beach on a sunny day?

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"
- Similar settings: what's the equilibrium condition?
  - Firms competing for profits?
  - Going to the beach on a sunny day?
  - Getting a table for Brunch on Sunday at 12pm?

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"
- Similar settings: what's the equilibrium condition?
  - Firms competing for profits?
  - Going to the beach on a sunny day?
  - Getting a table for Brunch on Sunday at 12pm?
- "Equilibrium" behavior is easy to predict in such settings
  - In equilibrium, the "good" deal is so crowded that it becomes just as bad as the "bad" deal

The Money Pot Game

- What is going on?
  - Everyone wants the money pot
  - But the more people take the money pot, the less money there is for everyone
  - In "Nash equilibrium", the money pot becomes no better than the "outside option"
- Similar settings: what's the equilibrium condition?
  - Firms competing for profits?
  - Going to the beach on a sunny day?
  - Getting a table for Brunch on Sunday at 12pm?
- "Equilibrium" behavior is easy to predict in such settings
  - In equilibrium, the "good" deal is so crowded that it becomes just as bad as the "bad" deal
- Game theory language: call these congestion games, or wars of attrition, or tragedy of the commons

BTC mining

- Suppose mining a block pays 0.001 BTC
- Suppose mining block costs $60 in electricity
  - If BTC price $> \$60k$, everyone mines; blocks produced quickly
  - If BTC price $< \$60k$, nobody mines; tx's aren't included and the blockchain stops working!
- The fact that mining rate depends on BTC vs electricity price is not desirable...
- Solution: use difficulty adjustment to maintain constant block production rate

BTC Difficulty Adjustment

- Blocks should be produced every 10 minutes
- Every 2016 blocks (approx 14 days), Bitcoin considers average block production rate, and...
  - Increases/decreases hash problem difficulty, if block production is too fast/slow
- What determines how fast blocks are produced, fixing hash rate?

BTC Difficulty Adjustment

- Blocks should be produced every 10 minutes
- Every 2016 blocks (approx 14 days), Bitcoin considers average block production rate, and...
  - Increases/decreases hash problem difficulty, if block production is too fast/slow
- What determines how fast blocks are produced, fixing hash rate?
  - How many people (that is, how much electricity) is devoted to mining

BTC Difficulty Adjustment

- Blocks should be produced every 10 minutes
- Every 2016 blocks (approx 14 days), Bitcoin considers average block production rate, and...
  - Increases/decreases hash problem difficulty, if block production is too fast/slow
- What determines how fast blocks are produced, fixing hash rate?
  - How many people (that is, how much electricity) is devoted to mining
- What determines how many people mine?
  - How profitable mining is, that is, how high BTC price is, vs electricity price

BTC: Equilibrium Condition

What happens to mining when BTC price goes up?

BTC: Equilibrium Condition

What happens to mining when BTC price goes up?

- More people try to mine
- Difficulty goes up; so electricity to mine a BTC increases
- $\implies$ profit per \$ of electricity spent doesn't change

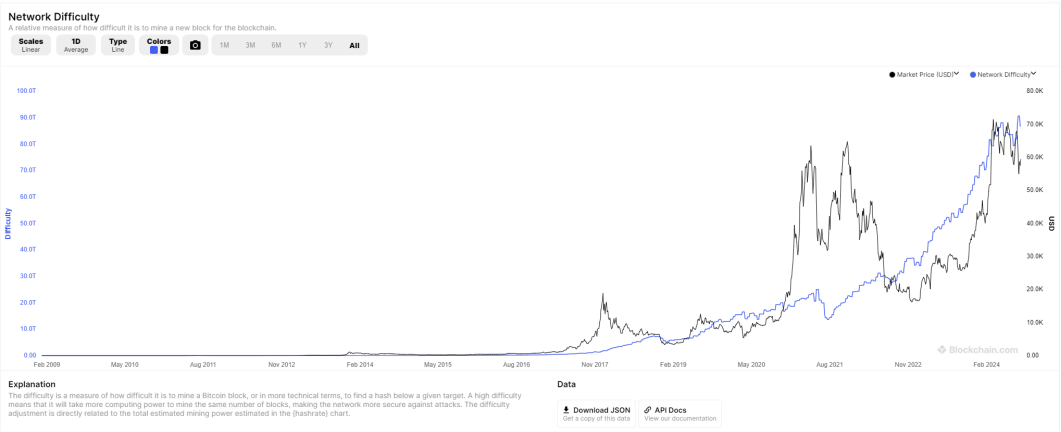
BTC: Equilibrium Condition

What happens to mining when BTC price goes up?

- More people try to mine
- Difficulty goes up; so electricity to mine a BTC increases
- $\implies$ profit per \$ of electricity spent doesn't change

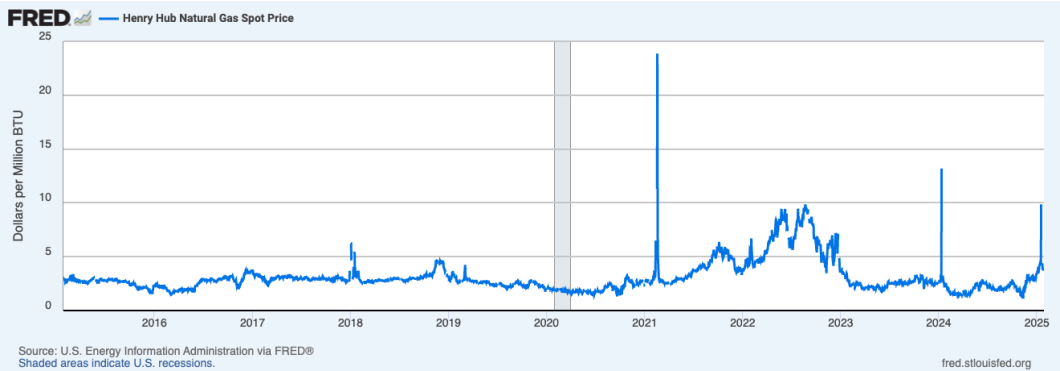
Prediction: difficulty should increase with BTC price

Difficulty vs BTC price



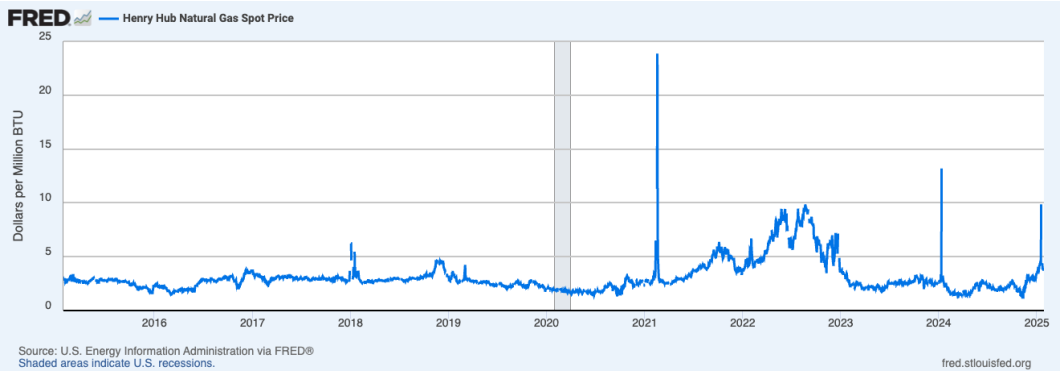
Source: Blockchain.com

Energy Prices...



Source: FRED

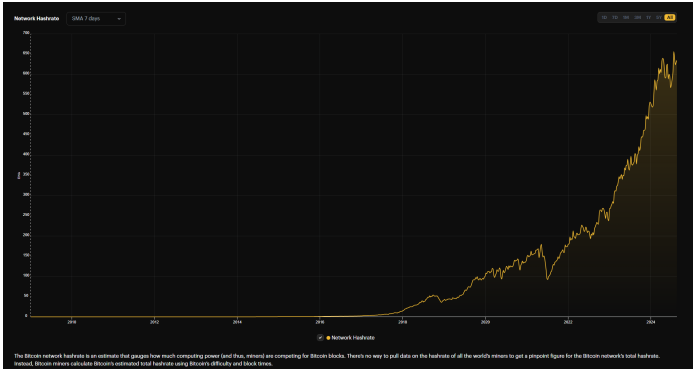
Energy Prices...



Source: FRED

A **BTU** is a "British Thermal Unit", the energy required to heat a pound of water by 1 degree Fahrenheit (about one match)

Technological Improvement



- 842 Exahashes/s = 842 billion billion hashes a second!
- Each living person in the world could shake each other living person's hand 10 times, every second!
- $\sim 1\%$ of global energy consumption...

Source: Hashrate Index

Back-of-Envelope Calculation

As of 2024,

- 164,250 BTC mined per year, worth around $8bil
  - In theory, what should total mining costs be?

Back-of-Envelope Calculation

As of 2024,

- 164,250 BTC mined per year, worth around $8bil
  - In theory, what should total mining costs be?
- Back of envelope calculation...
  - World electricity consumption approx 25,000TWh; BTC mining around 0.5%, so 125TWh
  - At US price $0.12/kWh, this is approx $15bil!
  - (a kWh (kilo-Watt-hour) is 3,412 BTUs)
- Electricity is probably somewhat cheaper, getting us close to $8bil

Back-of-Envelope Calculation

As of 2024,

- 164,250 BTC mined per year, worth around $8bil
  - In theory, what should total mining costs be?
- Back of envelope calculation...
  - World electricity consumption approx 25,000TWh; BTC mining around 0.5%, so 125TWh
  - At US price $0.12/kWh, this is approx $15bil!
  - (a kWh (kilo-Watt-hour) is 3,412 BTUs)
- Electricity is probably somewhat cheaper, getting us close to $8bil
- Will this decrease as technology improves?
  - No! Problem difficulty will just adjust...
- What if BTC price falls?

BTC Halving

- Approximately every 4 years, issuance of BTC drops by half
- Last halving: April 2024, now 3.125 BTC per block
- Does this affect block production rate?
  - No, difficulty adjustment

BTC Halving

- Approximately every 4 years, issuance of BTC drops by half
- Last halving: April 2024, now 3.125 BTC per block
- Does this affect block production rate?
  - No, difficulty adjustment
- Also implies there will only ever be 21 million BTC in existence!
  - By approximately 2140, the block reward will drop to less than the smallest unit of bitcoin ($1/10^8$) and become equal to zero.
  - Difficulty adjustment ensures that block production rate is approximately constant

Why 21 Million?

Satoshi: "My choice for the number of coins and distribution schedule was an educated guess. It was a difficult choice, because once the network is going it's locked in and we're stuck with it. I wanted to pick something that would make prices similar to existing currencies, but without knowing the future, that's very hard. I ended up picking something in the middle. If Bitcoin remains a small niche, it'll be worth less per unit than existing currencies. If you imagine it being used for some fraction of world commerce, then there's only going to be 21 million coins for the whole world, so it would be worth much more per unit. Values are 64-bit integers with 8 decimal places, so 1 coin is represented internally as 100000000. There's plenty of granularity if typical prices become small. For example, if 0.001 is worth 1 Euro, then it might be easier to change where the decimal point is displayed, so if you had 1 Bitcoin it's now displayed as 1000, and 0.001 is displayed as 1."

Transaction Fees

- Block rewards are not the only incentive to process transactions: miners also charge transaction fees to include a transaction in their blocks.

Transaction Fees

- Block rewards are not the only incentive to process transactions: miners also charge transaction fees to include a transaction in their blocks.
- Transaction fees are quoted in "sat/vB", or "Satoshis per virtual Byte"
  - A Satoshi is $1/10^8$ BTC, the smallest unit that can be transacted

Transaction Fees

- Block rewards are not the only incentive to process transactions: miners also charge transaction fees to include a transaction in their blocks.
- Transaction fees are quoted in "sat/vB", or "Satoshis per virtual Byte"
  - A Satoshi is $1/10^8$ BTC, the smallest unit that can be transacted
  - The Size of the transaction depends mostly on its format. The current most efficient format for submitting transactions was created in 2017.

Transaction Fees

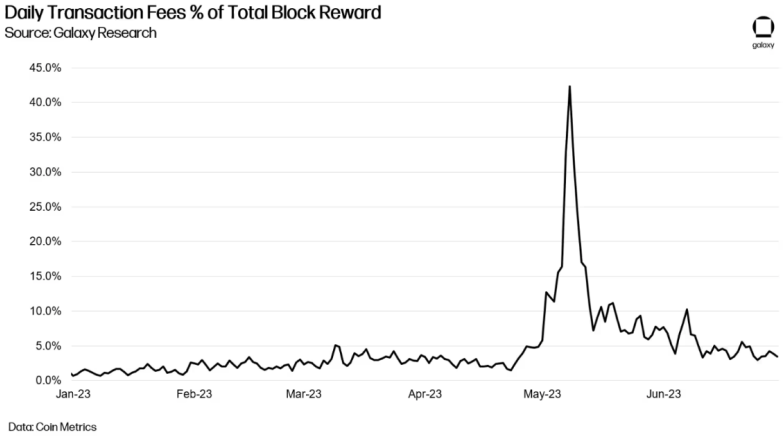
- Block rewards are not the only incentive to process transactions: miners also charge transaction fees to include a transaction in their blocks.
- Transaction fees are quoted in "sat/vB", or "Satoshis per virtual Byte"
  - A Satoshi is $1/10^8$ BTC, the smallest unit that can be transacted
  - The Size of the transaction depends mostly on its format. The current most efficient format for submitting transactions was created in 2017.
- Miners prioritize including transactions that offer higher fee rates
- As of 2/4/25, the going rate to complete your transaction within 30 minutes is 3 sat/vB, or $\$0.42$ for an average transaction

Transaction Fees

- Block rewards are not the only incentive to process transactions: miners also charge transaction fees to include a transaction in their blocks.
- Transaction fees are quoted in "sat/vB", or "Satoshis per virtual Byte"
  - A Satoshi is $1/10^8$ BTC, the smallest unit that can be transacted
  - The Size of the transaction depends mostly on its format. The current most efficient format for submitting transactions was created in 2017.
- Miners prioritize including transactions that offer higher fee rates
- As of 2/4/25, the going rate to complete your transaction within 30 minutes is 3 sat/vB, or $\$0.42$ for an average transaction
- A **"mempool"** is a pool of transactions waiting to be added to the bitcoin blockchain

Explore mempools at https://mempool.space/

Transaction Fees as % of Block Revenue



Source: Galaxy

Incentives

- A centralized authority can "censor", i.e. not include tx's
- Decentralization and competition "ensure" performance
  - If you don't include my tx, the next miner will
  - Even if the US bans US miners from including tx's, someone else will include them
- More or less works for just transacting

Incentives

- A centralized authority can "censor", i.e. not include tx's
- Decentralization and competition "ensure" performance
  - If you don't include my tx, the next miner will
  - Even if the US bans US miners from including tx's, someone else will include them
- More or less works for just transacting
- Incentives not that strong: tx fees $< 10\%$ of mining revenues
  - But, it's free $$!

Further Reading on Mining Industry

- Galaxy digital 2023 and 2024
- Cool paper on mining pools

Roadmap

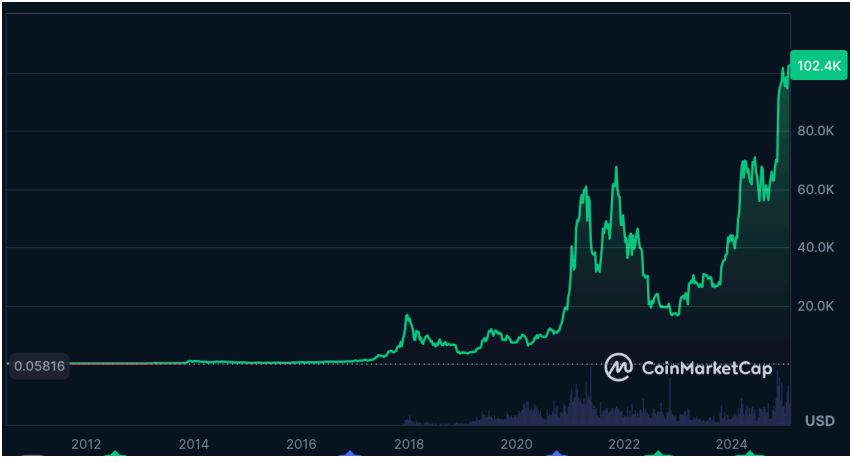
Satoshi Nakamoto

- Started sending emails about Bitcoin August 2008
- Oct 2008: Bitcoin white paper, code released
- January 9: first block mined!
  - "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"
- May 22, 2010: Laszlo Hanyecz bought two Papa John's pizzas for 10,000 BTC, now worth $600mil USD!
- Quickly became popular among black markets, Silk Road...
- Founder Ross Ulbricht was recently pardoned by Trump

Prizes for a Starcraft Tournament in 2011

Prizes:
1st Place: *$500*
2nd Place: *$250*
3rd Place: *$150*
4th Place: *$100*
5th-8th Place: *25 BitCoins*

Bitcoin Prices



Source: CoinMarketCap

Each BTC Block is a Merkle Tree of Transactions!

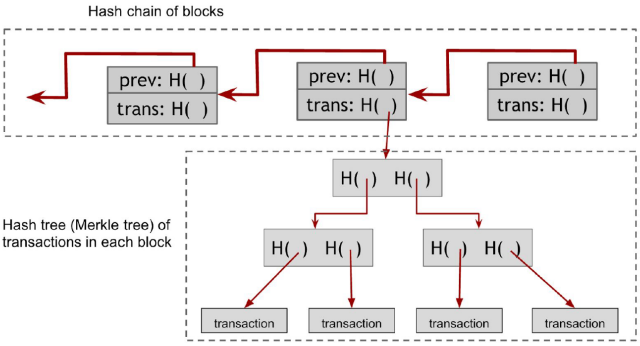


Figure 3.8. The Bitcoin block chain contains two different hash structures. The first is a hash chain of blocks that links the different blocks to one another. The second is internal to each block and is a Merkle Tree of transactions within the blocks.

- Valid block headers' hashes start with a big number of 0's (mining puzzle)

Source: NBFMG, fig. 3.8

Mining, in Practice

- Miners connected to each other in a network
- If you find a valid block, tell all other miners about it
  - They'll then mine the next block, based on yours
- If you solved the hash puzzle, why can't I just steal your solution?

Mining, in Practice

- Miners connected to each other in a network
- If you find a valid block, tell all other miners about it
  - They'll then mine the next block, based on yours
- If you solved the hash puzzle, why can't I just steal your solution?
  - My address is hashed into the block header

Mining, in Practice

- Miners connected to each other in a network
- If you find a valid block, tell all other miners about it
  - They'll then mine the next block, based on yours
- If you solved the hash puzzle, why can't I just steal your solution?
  - My address is hashed into the block header
- Why wouldn't I try to re-mine your block?

Mining, in Practice

- Miners connected to each other in a network
- If you find a valid block, tell all other miners about it
  - They'll then mine the next block, based on yours
- If you solved the hash puzzle, why can't I just steal your solution?
  - My address is hashed into the block header
- Why wouldn't I try to re-mine your block?
  - Consensus: if everyone else uses "longest chain", I do better by accepting your win, and trying to mine the next block

The Power of Consensus

- Notice that all properties depend on everyone "following the rules"...
  - If your block:
    - Doesn't hash to enough 0's, or
    - Steals $$, or
    - Includes invalid tx's, etc.
  - Others won't build on it, so others won't build on others not building on it,
  - So you won't get mining reward
- At some level, maybe the simplicity of the rules, plus the decentralization of mining, provides security?
  - By default, miners will pick the simplest possible path

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**
- Why can't miner steal my $?

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**
- Why can't miner steal my $?
  - Tx specifies signature and receiver; miner can only include tx
- What if miner doesn't want to include me?

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**
- Why can't miner steal my $?
  - Tx specifies signature and receiver; miner can only include tx
- What if miner doesn't want to include me?
  - Someone else will, and take the tx fee...
  - ...And miner eventually has to accept my tx as valid, or "fork" whole network (since hash pointers break)

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**
- Why can't miner steal my $?
  - Tx specifies signature and receiver; miner can only include tx
- What if miner doesn't want to include me?
  - Someone else will, and take the tx fee...
  - ...And miner eventually has to accept my tx as valid, or "fork" whole network (since hash pointers break)
- What if I try to spend twice?

Transacting, in Practice

- If I want to send, sign a tx, and tell all the miners about it
  - Pile of valid signed, but not-yet in chain, tx's floating around waiting to be included by miners called the **mempool**
- Why can't miner steal my $?
  - Tx specifies signature and receiver; miner can only include tx
- What if miner doesn't want to include me?
  - Someone else will, and take the tx fee...
  - ...And miner eventually has to accept my tx as valid, or "fork" whole network (since hash pointers break)
- What if I try to spend twice?
  - One tx is "invalid": doesn't obey the rules, since old tx output already spent
  - Remember, though; everything is ultimately due to consensus among miners, that "longest block of valid chains" is the legitimate blockchain

Fees and Times

- Fees: a few cents to a few USD
  - Doesn't depend on tx size
- Time: roughly 1hr
  - Most services wait a few blocks (why?)

Fees and Times

- Fees: a few cents to a few USD
  - Doesn't depend on tx size
- Time: roughly 1hr
  - Most services wait a few blocks (why?)
  - Tx's deeper down the blockchain are more "final" (harder to "doublespend")
- Long confirmation times a little awkward for "merchant" tx's

User Experience

- In practice, private keys held using "wallet" software or hardware
  - Software: Exodus, Electrum, MetaMask...
  - Hardware: Ledger, Trezor...
  - Some mobile app wallets, which I'm not familiar with
  - I use MetaMask

User Experience

- In practice, private keys held using "wallet" software or hardware
  - Software: Exodus, Electrum, MetaMask...
  - Hardware: Ledger, Trezor...
  - Some mobile app wallets, which I'm not familiar with
  - I use MetaMask
- In theory, no credit risk – you're holding your own BTC!
  - As long as internet + Bitcoin exists, you can use your BTC!

Don't Lose Your Keys...

As It Happens

This man owns $321M in bitcoin — but he can't access it because he lost his password

Stefan Thomas has 2 guesses left before he's locked out of his fortune forever

CBC Radio · Posted: Jan 15, 2021 5:02 PM CST | Last Updated: January 15, 2021

Stefan Thomas, the founder and CEO of micropayment streaming service Coil, has $321 million worth of bitcoins he can't access. (Submitted by Stefan Thomas)

284 comments

Read Story Transcript

Custodial Ownership

- Many don't care about decentralization, censorship-proofness...
  - But want to buy in case prices go up
- More and more ways to buy BTC
  - Crypto exchanges: Coinbase, Binance...
  - Classic institutions: Paypal, Robinhood...
- Recent ETF approvals makes it easier to buy BTC
- New SEC commissioner Paul Atkins has signaled positivity towards legitimizing crypto investments

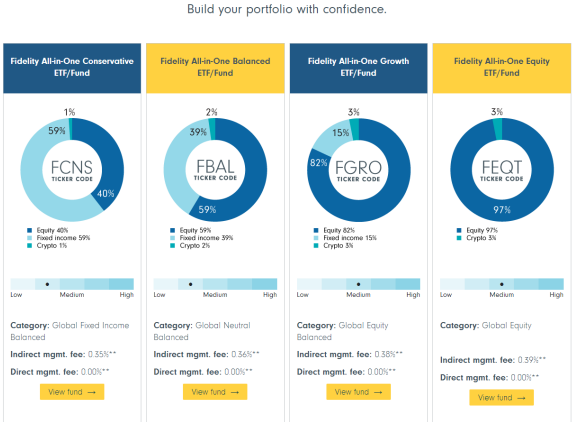
Mentions are not endorsements!

"Buy the Market"

The CAPM says buy the market – and BTC's market cap is now approx 2% S&P 500!

"Buy the Market"

The CAPM says buy the market – and BTC's market cap is now approx 2% S&P 500!



Source: Fidelity Canada

The "Market Cap" Illusion

Market Capitalization is just units outstanding times unit price

- Say everyone tried to sell their BTC tomorrow. Would they get that price for it?

The "Market Cap" Illusion

Market Capitalization is just units outstanding times unit price

- Say everyone tried to sell their BTC tomorrow. Would they get that price for it?
- Traditional assets (stocks, bonds) generate **dividends** or **coupons**
  - If a major owner of a stock worth 2% of the S&P500 sold, the supply would be absorbed relatively quickly
  - Investors are willing to substitute frictionlessly between stocks if one is temporarily undervalued

The "Market Cap" Illusion

Market Capitalization is just units outstanding times unit price

- Say everyone tried to sell their BTC tomorrow. Would they get that price for it?
- Traditional assets (stocks, bonds) generate **dividends** or **coupons**
  - If a major owner of a stock worth 2% of the S&P500 sold, the supply would be absorbed relatively quickly
  - Investors are willing to substitute frictionlessly between stocks if one is temporarily undervalued
- Reports of $TRUMP being worth $10 billion+ assume unit price would survive a large selloff (a bad assumption in my opinion)