Blockchain 101

Assoc. Prof. Dr. **Cihangir Tezcan**

MIDDLE EAST TECHNICAL UNIVERSITY

Informatics Institute, Department of Cyber Security
MIDDLE EAST TECHNICAL UNIVERSITY, ANKARA, TURKEY

ODTÜ BLOCKCHAIN
18 October 2022

You are doing it wrong

Cryptocurrencies

Cryptographers are not happy with how you're using the word 'crypto'

The renamed Crypto.com Arena is a win for cryptocurrency fans but strikes a blow against the word's original meaning

Matthew Cantor
@CantorMatthew

Thu 18 Nov 2021 16.14 GMT

Construction workers put the finishing touches on the Staples Center sign outside the arena in downtown Los Angeles on 16 September 1999. Photograph: Xerro Ryan Covarrubias/AP

The stadium that is home to the Los Angeles Lakers is getting a new name: the Crypto.com Arena. The name reflects the arena's new sponsorship agreement with a Singapore-based cryptocurrency trading platform. That may be good news for cryptocurrency fanatics - but perhaps not so much for another faction within the digital landscape: cryptographers.

Look up the word "crypto" in Webster's dictionary, and you'll see it refers to cryptography, which in turn is defined as "the computerized encoding and decoding of information". Search "crypto" on Google, however, and you'll see a host of top results pointing to cryptocurrencies like bitcoin and ethereum.

https://www.theguardian.com/technology/2021/nov/18/crypto-cryptocurrency-cryptographers

Short History

Short History

- The internet and cryptography made electronic money possible

Short History

Short History

- The internet and cryptography made electronic money possible
- Some of the desired properties
  1. No need for a central authority
  2. Prevent double spending
  3. Provide anonymity
  4. Allow offline payment
  5. ...

Short History

Short History

- The internet and cryptography made electronic money possible
- Some of the desired properties
  1. No need for a central authority
  2. Prevent double spending
  3. Provide anonymity
  4. Allow offline payment
  5. ...
- Almost every e-cash company failed until the invention of Bitcoin

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public
- Bitcoin has no **encryption** (Surprise!)

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public
- Bitcoin has no **encryption** (Surprise!)
- Integrity of ledgers: **cryptographic hash functions**

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public
- Bitcoin has no **encryption** (Surprise!)
- Integrity of ledgers: **cryptographic hash functions**
- Prevention of double spending: **hash puzzles**

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public
- Bitcoin has no **encryption** (Surprise!)
- Integrity of ledgers: **cryptographic hash functions**
- Prevention of double spending: **hash puzzles**
- To send bitcoins a transaction is authorized: **digital signatures**

Bitcoin

Bitcoin (Peer-to-peer electronic cash)

- Proposed by Satoshi Nakamoto in 2008 *(bitcoin.org/bitcoin.pdf)*
- Decentralized: Every node in the network keeps a copy of the ledger
- All accounts and transactions are public
- Bitcoin has no **encryption** (Surprise!)
- Integrity of ledgers: **cryptographic hash functions**
- Prevention of double spending: **hash puzzles**
- To send bitcoins a transaction is authorized: **digital signatures**
- Pseudo-anonymity: Account numbers are cryptographically obtained values and do no contain personal information

Cryptographic Hash Functions

Definition

A hash function $H(\cdot)$ is a keyless algorithm that takes a variable length input x and returns an output y of fixed length n (*message digest length*)

Cryptographic Hash Functions

Definition

A hash function $H(\cdot)$ is a keyless algorithm that takes a variable length input x and returns an output y of fixed length n (*message digest length*)

Cryptographic Hash Functions

We expect a cryptographic hash function satisfy the following properties

1. Preimage Resistance
2. Second Preimage Resistance
3. Collision Resistance

Cryptographic Hash Functions

Definition

A hash function $H(\cdot)$ is a keyless algorithm that takes a variable length input x and returns an output y of fixed length n (*message digest length*)

Cryptographic Hash Functions

We expect a cryptographic hash function satisfy the following properties

1. Preimage Resistance
2. Second Preimage Resistance
3. Collision Resistance

Actually we want a hash function to behave like a **random** function

Resistance

1. Preimage Resistance

Given an output y it should be hard to find any input x such that $H(x) = y$

Resistance

1. Preimage Resistance

Given an output y it should be hard to find any input x such that $H(x) = y$

2. Second Preimage Resistance

Given an output y and input x_1 such that $H(x_1) = y$, it should be hard to find any other input x_2 such that $H(x_2) = y$

Resistance

1. Preimage Resistance

Given an output y it should be hard to find any input x such that $H(x) = y$

2. Second Preimage Resistance

Given an output y and input x_1 such that $H(x_1) = y$, it should be hard to find any other input x_2 such that $H(x_2) = y$

3. Collision Resistance

It should be hard to find two inputs x_1 and x_2 such that $H(x_1) = H(x_2)$

Hash Functions

Secure Hash Functions

Algorithm	**n**	**Security**
MD4	128	*Broken*
MD5	128	*Broken*
SHA-1	160	*Broken*

Hash Functions

Secure Hash Functions

Algorithm	**n**	**Security**
MD4	128	*Broken*
MD5	128	*Broken*
SHA-1	160	*Broken*
RIPEMD-160	160	Secure
SHA-2	224,256,384,512	Secure
SHA-3	224,256,384,512	Secure

Hash Functions

Secure Hash Functions

Algorithm	**n**	**Security**
MD4	128	*Broken*
MD5	128	*Broken*
SHA-1	160	*Broken*
RIPEMD-160	160	Secure
SHA-2	224,256,384,512	Secure
SHA-3	224,256,384,512	Secure

Bitcoin script allows the use of **SHA-1**, **RIPEMD-160**, and **SHA-2**

Discrete Logarithm Problem (DLP)

Discrete Logarithm Problem (DLP)

Let G be a group and g its generator

- Given a, it is *feasible* to compute g^a

Discrete Logarithm Problem (DLP)

Discrete Logarithm Problem (DLP)

Let G be a group and g its generator

- Given a, it is *feasible* to compute g^a
- Given $b = g^a$, it is **not** *feasible* to find a

Discrete Logarithm Problem (DLP)

Discrete Logarithm Problem (DLP)

Let G be a group and g its generator

- Given a, it is *feasible* to compute g^a
- Given $b = g^a$, it is **not** *feasible* to find a

Example

- $3^x = 73768848753614705 34 \pmod{18446744073709551615}$

Discrete Logarithm Problem (DLP)

Discrete Logarithm Problem (DLP)

Let G be a group and g its generator

- Given a, it is *feasible* to compute g^a
- Given $b = g^a$, it is **not** *feasible* to find a

Example

- $3^x = 737688487536147053 4 \pmod{18446744073709551615}$
- **Answer:** $x = 127002$

Discrete Logarithm Problem (DLP)

Discrete Logarithm Problem (DLP)

Let G be a group and g its generator

- Given a, it is *feasible* to compute g^a
- Given $b = g^a$, it is **not** *feasible* to find a

Example

- $3^x = 737688487536147053 4 \pmod{18446744073709551615}$
- **Answer:** $x = 127002$

Warning

Hardness (intractability) of the discrete logarithm problem depends on the group! (e.g. DLP on an elliptic curve $y^2 = x^3 + ax + b$)

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q
4. Choose L-bit prime modulus p s.t. $p-1$ is a multiple of q

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q
4. Choose L-bit prime modulus p s.t. $p - 1$ is a multiple of q
5. Choose g (its multiplicative group order modulo p must be q)

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q
4. Choose L-bit prime modulus p s.t. $p-1$ is a multiple of q
5. Choose g (its multiplicative group order modulo p must be q)

Parameters (p, q, g) may be shared between different users of the system

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q
4. Choose L-bit prime modulus p s.t. $p-1$ is a multiple of q
5. Choose g (its multiplicative group order modulo p must be q)

Parameters (p, q, g) may be shared between different users of the system

User key Generation

1. **Randomly** choose a secret key x where $0 < x < q$

Digital Signature Algorithm/Standard (DSA/DSS)

Parameter Generation

1. Choose a good hash function $H()$ (e.g. SHA-2, SHA-3)
2. Choose key lengths L and N (e.g. (3072,256))
3. Choose N-bit prime q
4. Choose L-bit prime modulus p s.t. $p - 1$ is a multiple of q
5. Choose g (its multiplicative group order modulo p must be q)

Parameters (p, q, g) may be shared between different users of the system

User key Generation

1. **Randomly** choose a secret key x where $0 < x < q$
2. Calculate the public key $y = g^x \pmod p$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message $0 < k < q$**

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q \ )$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q \ )$ (m is the message)

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

The signature is (r, s)

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

The signature is (r, s)

Verifying

1. Calculate $w = s^{-1} \ (\mod q)$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

The signature is (r, s)

Verifying

1. Calculate $w = s^{-1} \ (\mod q)$
2. Calculate $u_1 = H(m) \cdot w \ (\mod q)$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

The signature is (r, s)

Verifying

1. Calculate $w = s^{-1} \ (\mod q)$
2. Calculate $u_1 = H(m) \cdot w \ (\mod q)$
3. Calculate $u_2 = r \cdot w \ (\mod q)$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

The signature is (r, s)

Verifying

1. Calculate $w = s^{-1} \ (\mod q)$
2. Calculate $u_1 = H(m) \cdot w \ (\mod q)$
3. Calculate $u_2 = r \cdot w \ (\mod q)$
4. Calculate $v = (g^{u_1} y^{u_2} \mod p) \ (\mod q)$

Digital Signature Algorithm/Standard (DSA/DSS)

Signing

1. **Randomly choose k per message** $0 < k < q$
2. Calculate $r = (g^k \mod p) \ (\mod q)$
3. Calculate $s = k^{-1}(H(m) + xr) \ (\mod q)$ (m is the message)

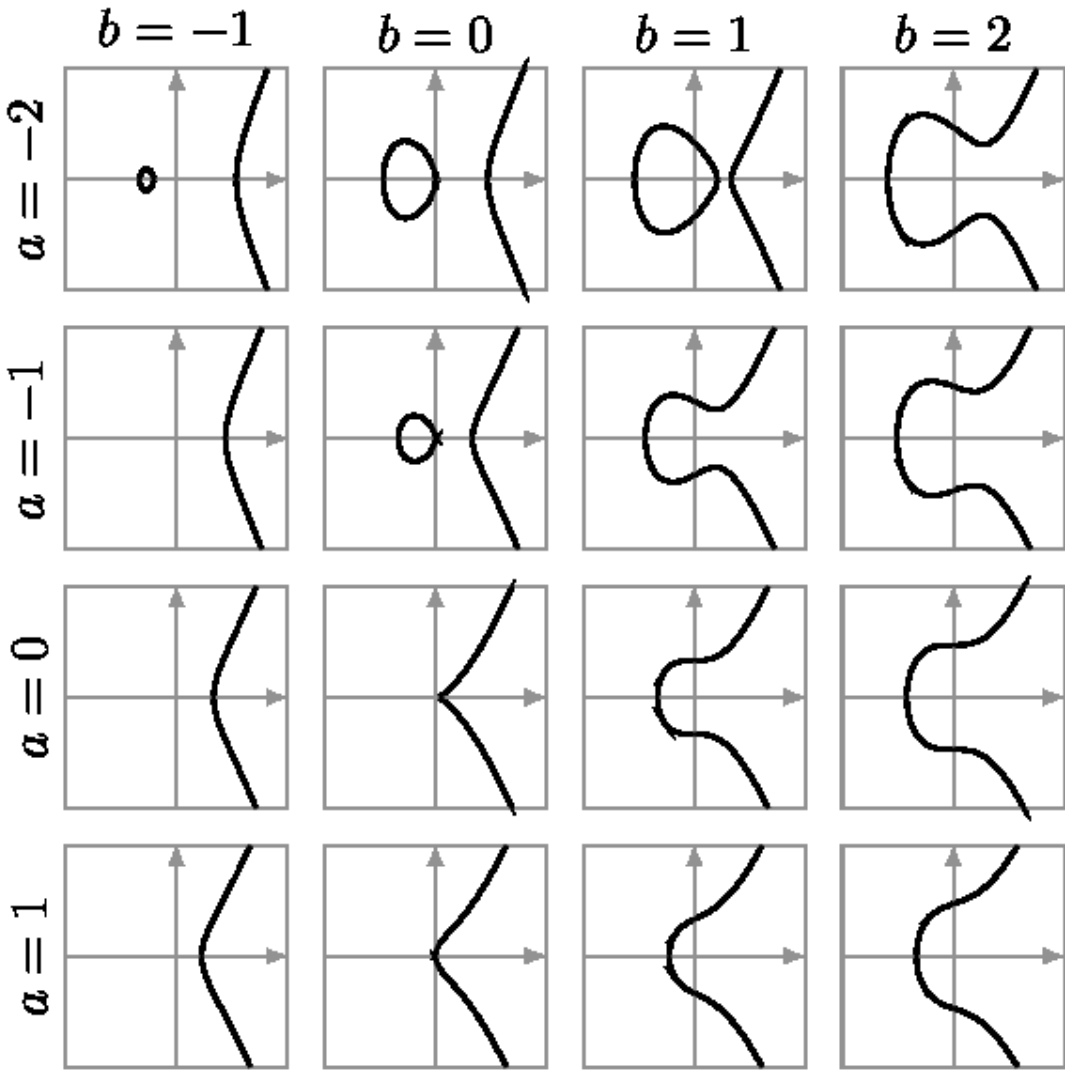
The signature is (r, s)

Verifying

1. Calculate $w = s^{-1} \ (\mod q)$
2. Calculate $u_1 = H(m) \cdot w \ (\mod q)$
3. Calculate $u_2 = r \cdot w \ (\mod q)$
4. Calculate $v = (g^{u_1} y^{u_2} \mod p) \ (\mod q)$

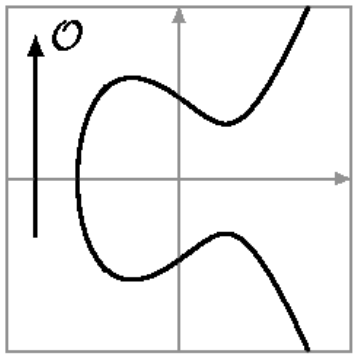
The signature is valid if $v = r$

Elliptic Curve Cryptography

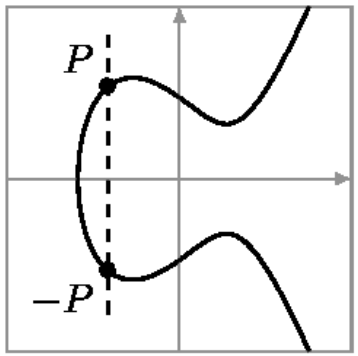


$$y^2 = x^3 + a \cdot x + b$$

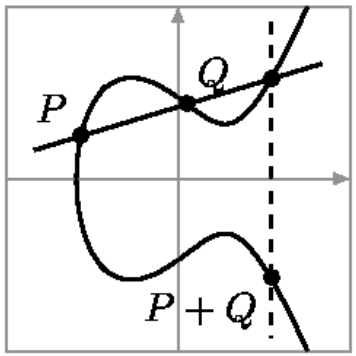
Elliptic Curve Cryptography



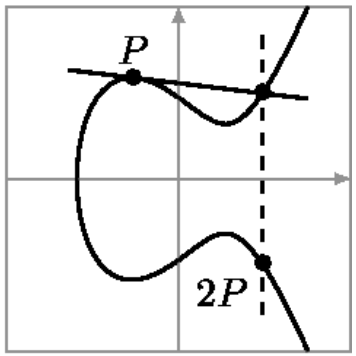
Neutral element $\mathcal{O}$



Inverse element $-P$

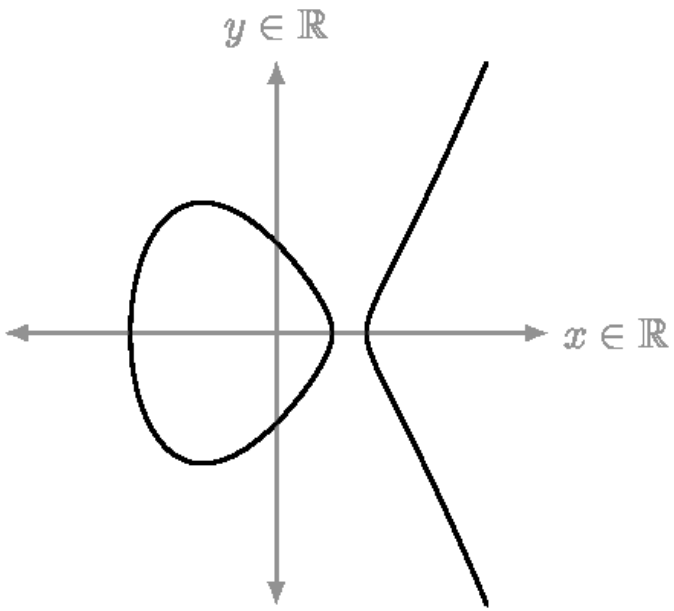


Addition $P + Q$
"Chord rule"

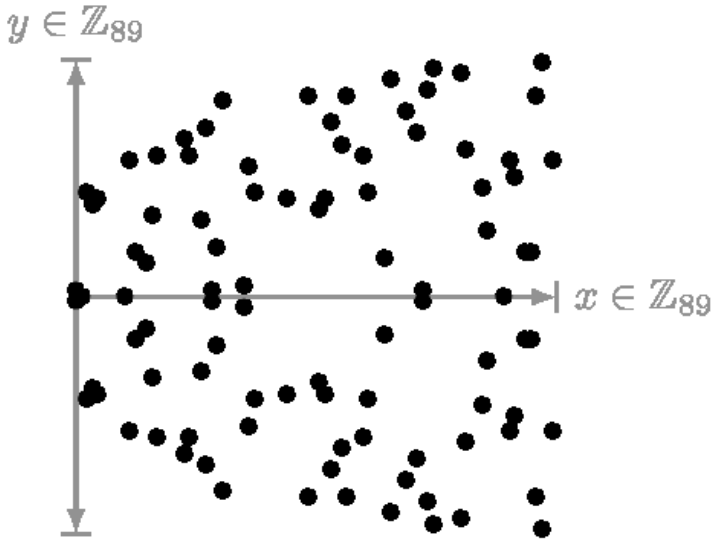


Doubling $P + P$
"Tangent rule"

Elliptic Curve Cryptography

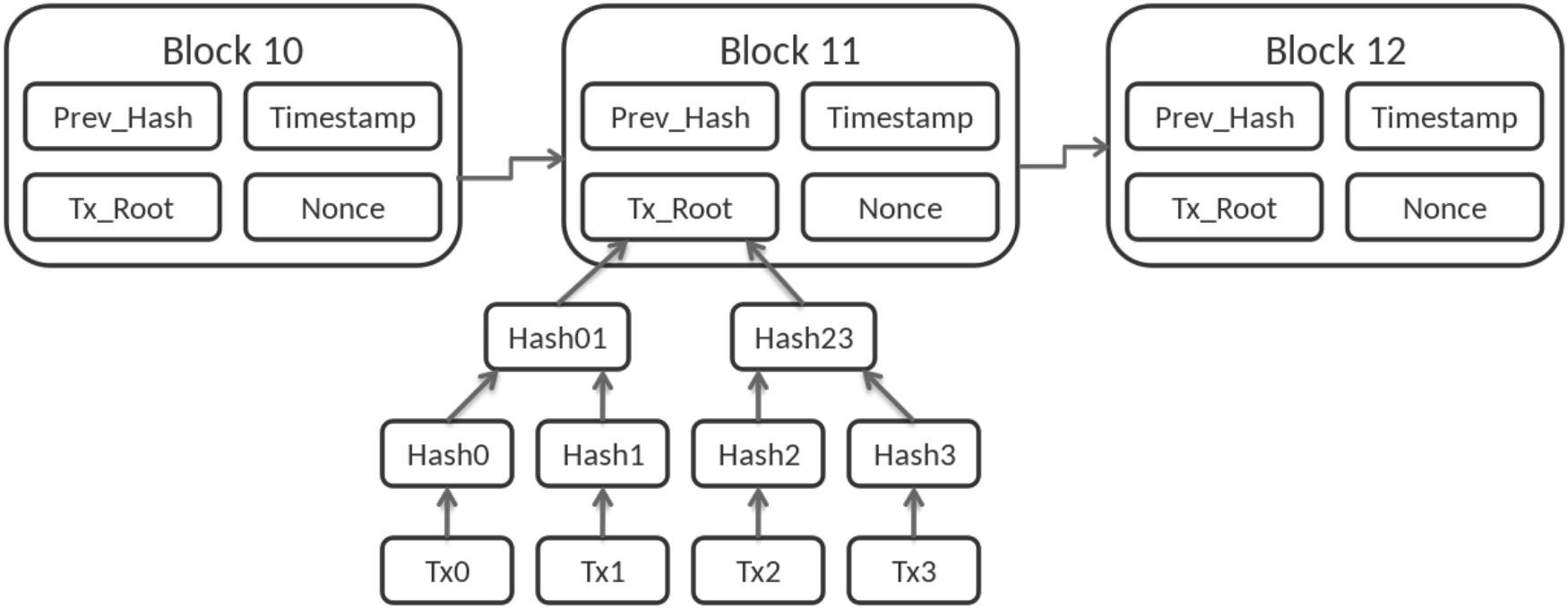


$y^2 = x^3 - 2x + 1$ over $\mathbb{R}$



$y^2 = x^3 - 2x + 1$ over $\mathbb{Z}_{89}$

Blockchain



Bitcoin blockchain structure by Matthäus Wander (CC BY-SA 3.0)

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million
- **Bitcoin in Circulation:** 19.2 million

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million
- **Bitcoin in Circulation:** 19.2 million
- **Mining Award:** 6.25 BTC

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million
- **Bitcoin in Circulation:** 19.2 million
- **Mining Award:** 6.25 BTC
- **Award Halving:** 4 years

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million
- **Bitcoin in Circulation:** 19.2 million
- **Mining Award:** 6.25 BTC
- **Award Halving:** 4 years

Forks

- Any change in these specs causes a **hard fork**

Bitcoin

Bitcoin (Specs)

- **Block Size:** 1 MB (virtually 4 MB since 2017, generally does not exceed 1.5 MB)
- **Blockchain Size:** 421+ GBs
- **Block Addition Time:** 10 minutes
- **Total Bitcoin Amount:** 21 million
- **Bitcoin in Circulation:** 19.2 million
- **Mining Award:** 6.25 BTC
- **Award Halving:** 4 years

Forks

- Any change in these specs causes a **hard fork**
- Adding a cryptographic algorithm causes a **soft fork**

P2SH Example: SHA1 Collision Bounty

SHA1 Collision Bitcoin Bounty (September 2013)

- Peter Todd offered rewards for SHA-1, RIPEMD-160, and SHA-256 collisions by transferring bitcoins to P2SH accounts which can only be claimed by providing collisions

P2SH Example: SHA1 Collision Bounty

SHA1 Collision Bitcoin Bounty (September 2013)

- Peter Todd offered rewards for SHA-1, RIPEMD-160, and SHA-256 collisions by transferring bitcoins to P2SH accounts which can only be claimed by providing collisions
- The SHA1 bounty has address 37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP

P2SH Example: SHA1 Collision Bounty

SHA1 Collision Bitcoin Bounty (September 2013)

- Peter Todd offered rewards for SHA-1, RIPEMD-160, and SHA-256 collisions by transferring bitcoins to P2SH accounts which can only be claimed by providing collisions
- The SHA1 bounty has address 37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP
- `https://blockchain.info/address/37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP`

P2SH Example: SHA1 Collision Bounty

SHA1 Collision Bitcoin Bounty (September 2013)

- Peter Todd offered rewards for SHA-1, RIPEMD-160, and SHA-256 collisions by transferring bitcoins to P2SH accounts which can only be claimed by providing collisions
- The SHA1 bounty has address 37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP
- `https://blockchain.info/address/37k7toV1Nv4DfmQbmZ8KuZDQCYK9x5KpzP`
- A disassembly of its script is
  1. *OP_2DUP*
  2. *OP_EQUAL*
  3. *OP_NOT*
  4. *OP_VERIFY*
  5. *OP_SHA*1
  6. *OP_SWAP*
  7. *OP_SHA*1
  8. *OP_EQUAL*

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate
4. More transactions per second

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate
4. More transactions per second
5. Smaller transactions fees

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate
4. More transactions per second
5. Smaller transactions fees
6. GPU-friendly mining

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate
4. More transactions per second
5. Smaller transactions fees
6. GPU-friendly mining
7. Scripting language with more capabilities

Alternative Coins

Alternative Coins

Reasons for introducing Bitcoin alternatives

1. Higher coin cap
2. Faster transactions
3. Coin introduction rate
4. More transactions per second
5. Smaller transactions fees
6. GPU-friendly mining
7. Scripting language with more capabilities
8. Full anonymity

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker
- The event sparked a debate in the crypto-community about whether Ethereum should perform a contentious *hard fork* to reappropriate the affected funds

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker
- The event sparked a debate in the crypto-community about whether Ethereum should perform a contentious *hard fork* to reappropriate the affected funds
- This resulted in the network splitting into two blockchains

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker
- The event sparked a debate in the crypto-community about whether Ethereum should perform a contentious *hard fork* to reappropriate the affected funds
- This resulted in the network splitting into two blockchains
  1. *Ethereum* with the theft reversed

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker
- The event sparked a debate in the crypto-community about whether Ethereum should perform a contentious *hard fork* to reappropriate the affected funds
- This resulted in the network splitting into two blockchains
  1. *Ethereum* with the theft reversed
  2. *Ethereum Classic* which continued on the original chain

Is a Blockchain Temper-Proof?

Ethereum Classic

- In 2016, a decentralized autonomous organization called The DAO - a set of smart contracts developed on the platform - raised a record $150 USD million in a crowd sale to fund the project
- The DAO was exploited in June 2016 when $50 USD million of DAO tokens were stolen by an unknown hacker
- The event sparked a debate in the crypto-community about whether Ethereum should perform a contentious *hard fork* to reappropriate the affected funds
- This resulted in the network splitting into two blockchains
  1. *Ethereum* with the theft reversed
  2. *Ethereum Classic* which continued on the original chain
- After the hard fork, Ethereum subsequently forked twice in the fourth quarter of 2016 to deal with other attacks

Hash Puzzle to Prevent Double Spending

Proof of Work

Find a hash value that is smaller than a target value

Hash Puzzle to Prevent Double Spending

Proof of Work

Find a hash value that is smaller than a target value

Shares

Miners join together to form *pools*. They prove that they are working by providing hash values slightly larger than the target value.

Hash Puzzle to Prevent Double Spending

Proof of Work

Find a hash value that is smaller than a target value

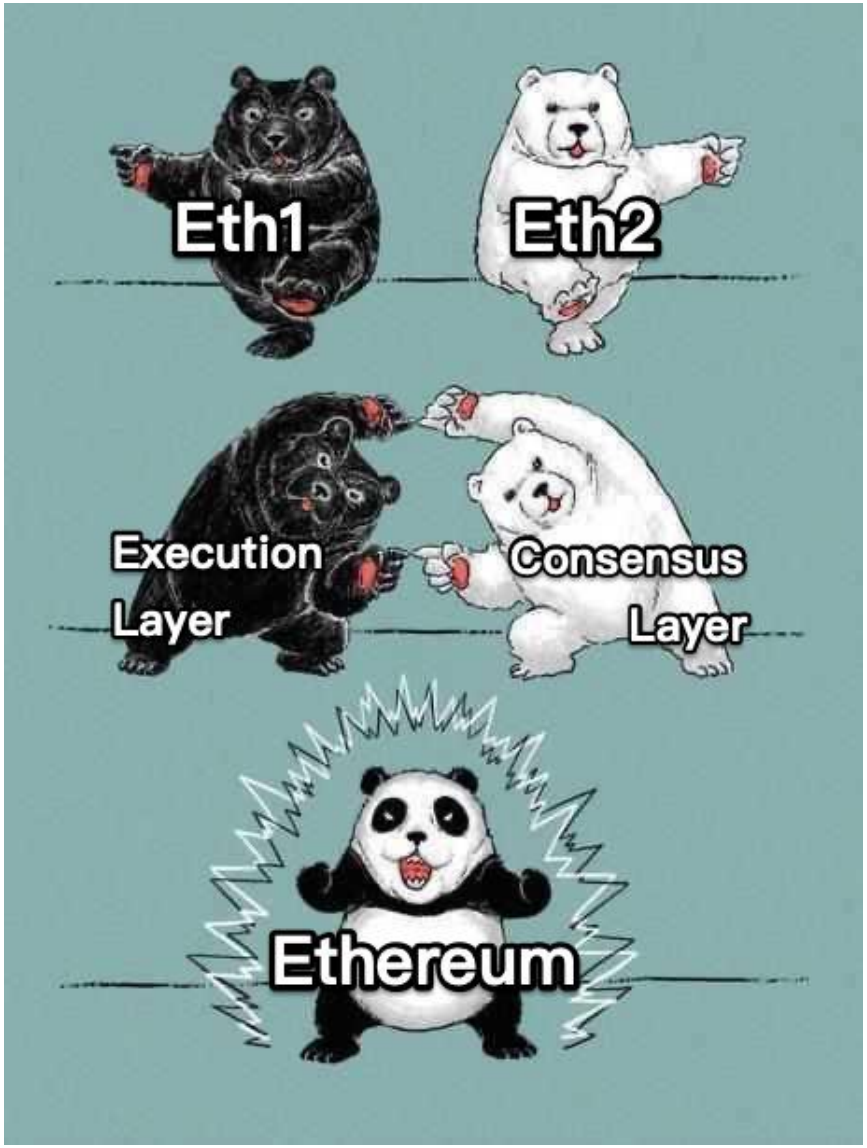
Shares

Miners join together to form *pools*. They prove that they are working by providing hash values slightly larger than the target value.

Electricity Consumption

- Bitcoin mining's yearly electricity consumption is around 200 TWh
- Norway's yearly electricity consumption is around 120 TWh
- Ethereum mining's yearly electricity consumption is around 100 TWh

Ethereum Merge



https://blog.ethereum.org/2022/01/24/the-great-eth2-renaming/

A Bitcoin Block

Block 714108

USD BTC

This block was mined on December 14, 2021 at 2:49 PM GMT+3 by AntPool. It currently has 1 confirmations on the Bitcoin blockchain.

The miner(s) of this block earned a total reward of 6.25000000 BTC ($299,190.81). The reward consisted of a base reward of 6.25000000 BTC ($299,190.81) with an additional 0.07192409 BTC ($3,443.04) reward paid as fees of the 3561 transactions which were included in the block. The Block rewards, also known as the Coinbase reward, were sent to this address.

A total of 27,890.66393872 BTC ($1,335,140,864.80) were sent in the block with the average transaction being 7.83225609 BTC ($374,934.25). Learn more about how blocks work.

Hash	00000000000000000008247d658dbb5502b4ad00087adc0823ddd34f4dce5b79
Confirmations	1
Timestamp	2021-12-14 14:49
Height	714108
Miner	AntPool
Number of Transactions	3,561
Difficulty	24,195,286,980,613.62
Merkle root	b716d73ff930a5fe5ca70a573734bbe6fc9506d4c1953438543f22224dcbdcbf
Version	0x20400004
Bits	386,638,367
Weight	3,993,549 WU
Size	1,538,685 bytes
Nonce	4,170,003,327
Transaction Volume	27890.66393872 BTC
Block Reward	6.25000000 BTC
Fee Reward	0.07192409 BTC

From www.blockchain.com

Digital Signatures

Randomness

- Randomness plays a very important role in DSA, ECDSA, and Schnorr Signature algorithms
- We require random numbers for key generation and every signature

Digital Signatures

Randomness

- Randomness plays a very important role in DSA, ECDSA, and Schnorr Signature algorithms
- We require random numbers for key generation and every signature
- Thus, a strong random number generator (RNG) is a must for security of these algorithms
- Using the same **k** for signing two different transactions allows an attacker to capture this **k** and consequently obtain the secret key **x**

Digital Signatures

Randomness

- Randomness plays a very important role in DSA, ECDSA, and Schnorr Signature algorithms
- We require random numbers for key generation and every signature
- Thus, a strong random number generator (RNG) is a must for security of these algorithms
- Using the same **k** for signing two different transactions allows an attacker to capture this **k** and consequently obtain the secret key **x**
- Some Bitcoin users used the same **k** for signing two different transactions due to a Java bug in the Java SecureRandom class on Android and attackers stole their bitcoins by obtaining secret keys from these transactions

Elliptic Curve DSA (ECDSA)

How to Choose an Elliptic Curve?

- Bitcoin uses the elliptic curve $y^2 = x^3 + 7$ (secp256k1) over F_p where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$

Elliptic Curve DSA (ECDSA)

How to Choose an Elliptic Curve?

- Bitcoin uses the elliptic curve $y^2 = x^3 + 7$ (secp256k1) over F_p where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- To avoid known attacks, we need ordinary, non-anomalous elliptic curves with group orders divisible by a large prime number of at least 224 bits

Elliptic Curve DSA (ECDSA)

How to Choose an Elliptic Curve?

- Bitcoin uses the elliptic curve $y^2 = x^3 + 7$ (secp256k1) over F_p where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- To avoid known attacks, we need ordinary, non-anomalous elliptic curves with group orders divisible by a large prime number of at least 224 bits
- Unlike DSA, ECDSA allows us to choose many elliptic curves with the same number of points on them. Thus, a natural question arises: Does the DLP equally hard for elliptic curves that have the same number of points?

Elliptic Curve DSA (ECDSA)

How to Choose an Elliptic Curve?

- Bitcoin uses the elliptic curve $y^2 = x^3 + 7$ (secp256k1) over F_p where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- To avoid known attacks, we need ordinary, non-anomalous elliptic curves with group orders divisible by a large prime number of at least 224 bits
- Unlike DSA, ECDSA allows us to choose many elliptic curves with the same number of points on them. Thus, a natural question arises: Does the DLP equally hard for elliptic curves that have the same number of points?
- ECDLP is random self-reducible among curves with the same number of points

Elliptic Curve DSA (ECDSA)

How to Choose an Elliptic Curve?

- Bitcoin uses the elliptic curve $y^2 = x^3 + 7$ (secp256k1) over F_p where $p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$
- To avoid known attacks, we need ordinary, non-anomalous elliptic curves with group orders divisible by a large prime number of at least 224 bits
- Unlike DSA, ECDSA allows us to choose many elliptic curves with the same number of points on them. Thus, a natural question arises: Does the DLP equally hard for elliptic curves that have the same number of points?
- ECDLP is random self-reducible among curves with the same number of points
- *Inside Information:* NIST is planning to add *secp256k1* to its standards for interoperability

Post-Quantum Cryptography

Post-Quantum Cryptography

- Development of a large quantum computer would allow to solve ECDLP, DLP, and IFP on a quantum computer in polynomial-time

Post-Quantum Cryptography

Post-Quantum Cryptography

- Development of a large quantum computer would allow to solve ECDLP, DLP, and IFP on a quantum computer in polynomial-time
- IBM achieved 127 qubits in November 2021

Post-Quantum Cryptography

Post-Quantum Cryptography

- Development of a large quantum computer would allow to solve ECDLP, DLP, and IFP on a quantum computer in polynomial-time
- IBM achieved 127 qubits in November 2021
- To break 80-bit security (e.g RSA-1024) one needs 2048 qubits (or 1.5m real qubits)

Post-Quantum Cryptography

Post-Quantum Cryptography

- Development of a large quantum computer would allow to solve ECDLP, DLP, and IFP on a quantum computer in polynomial-time
- IBM achieved 127 qubits in November 2021
- To break 80-bit security (e.g RSA-1024) one needs 2048 qubits (or 1.5m real qubits)
- NIST is organizing Post-Quantum Cryptography Competition (2016-2024)

Post-Quantum Cryptography

Post-Quantum Cryptography

- Development of a large quantum computer would allow to solve ECDLP, DLP, and IFP on a quantum computer in polynomial-time
- IBM achieved 127 qubits in November 2021
- To break 80-bit security (e.g RSA-1024) one needs 2048 qubits (or 1.5m real qubits)
- NIST is organizing Post-Quantum Cryptography Competition (2016-2024)
- Recall that adding new cryptographic algorithms to a blockchain requires soft forks

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing
- Two protocols, Zerocoin and Zerocash, provide anonymity at the protocol level by using zero-knowledge protocols

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing
- Two protocols, Zerocoin and Zerocash, provide anonymity at the protocol level by using zero-knowledge protocols
- Main idea behind zero-knowledge protocols is that one party can prove that they know a value x without revealing it

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing
- Two protocols, Zerocoin and Zerocash, provide anonymity at the protocol level by using zero-knowledge protocols
- Main idea behind zero-knowledge protocols is that one party can prove that they know a value x without revealing it
- Zerocoin protocol uses two coins A and B to obtain anonymity: A user obtains anonymity by destroying their A coins and minting same amount of B coins

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing
- Two protocols, Zerocoin and Zerocash, provide anonymity at the protocol level by using zero-knowledge protocols
- Main idea behind zero-knowledge protocols is that one party can prove that they know a value x without revealing it
- Zerocoin protocol uses two coins A and B to obtain anonymity: A user obtains anonymity by destroying their A coins and minting same amount of B coins
- Zerocash uses zk-SNARKs which make zero-knowledge proofs much more compact and efficient to verify (288 bytes)
- SNARKS are not post-quantum secure

Zero-Knowledge Proofs

Zero-Knowledge Proofs

- Cryptocurrencies like Bitcoin provide pseudo-anonymity
- To obtain anonymity, users use some non-cryptographic techniques like mixing
- Two protocols, Zerocoin and Zerocash, provide anonymity at the protocol level by using zero-knowledge protocols
- Main idea behind zero-knowledge protocols is that one party can prove that they know a value x without revealing it
- Zerocoin protocol uses two coins A and B to obtain anonymity: A user obtains anonymity by destroying their A coins and minting same amount of B coins
- Zerocash uses zk-SNARKs which make zero-knowledge proofs much more compact and efficient to verify (288 bytes)
- SNARKS are not post-quantum secure
- STARKS are quantum secure but proof sizes are larger (45-200 KB)

Lightweight Cryptography

Blockchains for IoT Devices

- Lightweight cryptography focuses on resource-constrained devices and tries to provide solutions that are tailored for them
  - require less energy
  - require less power
  - have small latency
  - provide better throughput
  - have side-channel resistance

Lightweight Cryptography

Blockchains for IoT Devices

- NIST is organizing Lightweight Authenticated Encryption Competition (2019-2023)

Lightweight Cryptography

Blockchains for IoT Devices

- NIST is organizing Lightweight Authenticated Encryption Competition (2019-2023)
- ISO/IEC 29192-5:2016 Lightweight Hash Functions standards
  - Photon (80, 128, 160, 224, and 256 bits)
  - Spongent (88, 128, 160, 224, and 256 bits)
  - Lesamnta-LW (256 bits)

Lightweight Cryptography

Blockchains for IoT Devices

- NIST is organizing Lightweight Authenticated Encryption Competition (2019-2023)
- ISO/IEC 29192-5:2016 Lightweight Hash Functions standards
  - Photon (80, 128, 160, 224, and 256 bits)
  - Spongent (88, 128, 160, 224, and 256 bits)
  - Lesamnta-LW (256 bits)
- ISO/IEC 29192-3:2012 Lightweight Stream Cipher standards
  - Trivium (80-bit key)
  - Enocoro (80 and 128-bit keys)

Lightweight Cryptography

Blockchains for IoT Devices

- NIST is organizing Lightweight Authenticated Encryption Competition (2019-2023)
- ISO/IEC 29192-5:2016 Lightweight Hash Functions standards
  - Photon (80, 128, 160, 224, and 256 bits)
  - Spongent (88, 128, 160, 224, and 256 bits)
  - Lesamnta-LW (256 bits)
- ISO/IEC 29192-3:2012 Lightweight Stream Cipher standards
  - Trivium (80-bit key)
  - Enocoro (80 and 128-bit keys)
- ISO/IEC 29192-2:2012 Lightweight Block Cipher standards
  - PRESENT (80 and 128-bit keys)
  - CLEFIA (128, 192, and 256-bit keys)
  - LEA (128, 192, and 256-bit keys)

Misconceptions

Misconceptions

1. Cryptocurrencies do NOT use encryption

Misconceptions

Misconceptions

1. Cryptocurrencies do NOT use encryption
2. Not every blockchain requires high electricity consumption

Misconceptions

Misconceptions

1. Cryptocurrencies do NOT use encryption
2. Not every blockchain requires high electricity consumption
3. NOT every blockchain is permissionless

Misconceptions

Misconceptions

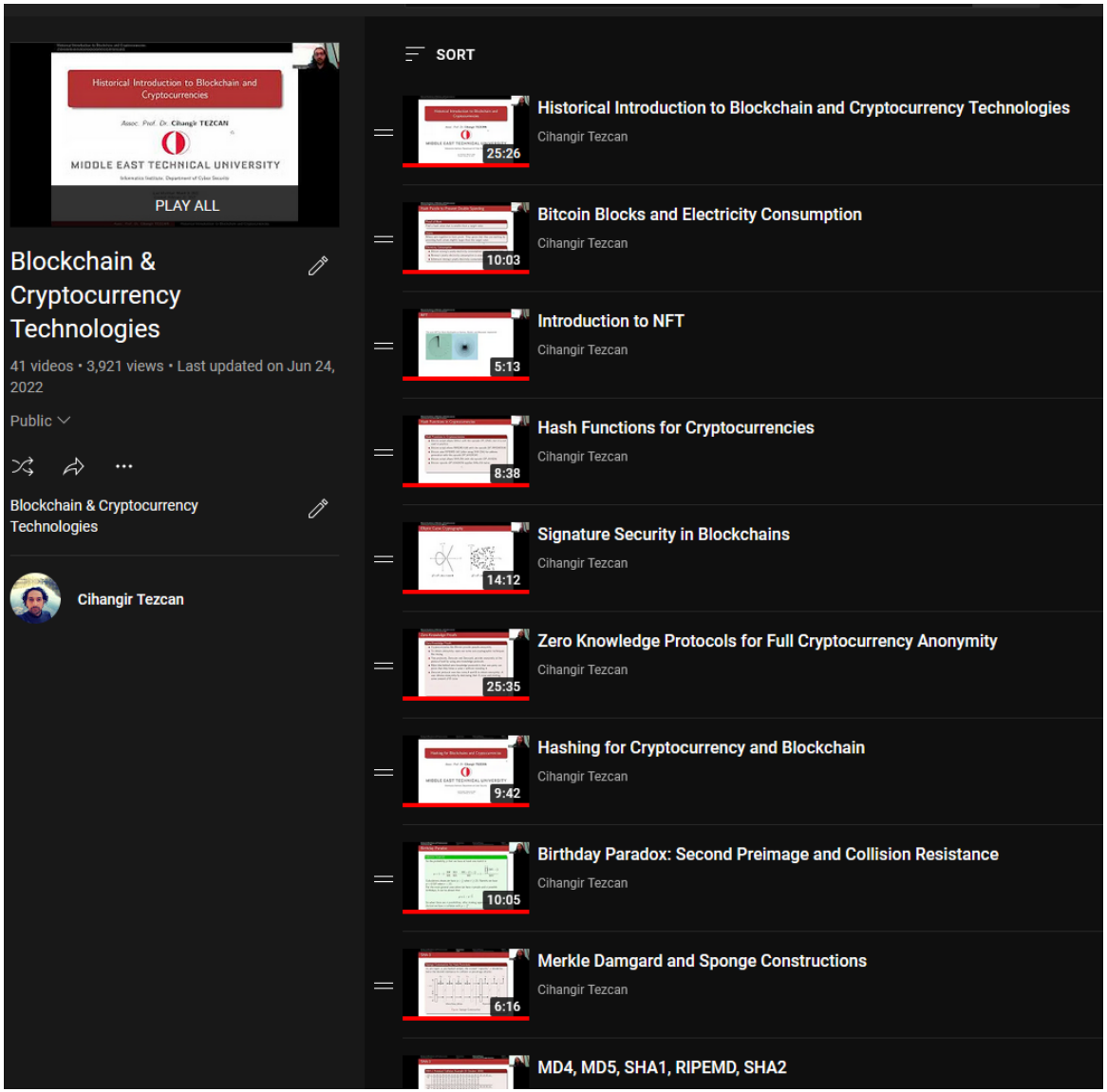
1. Cryptocurrencies do NOT use encryption
2. Not every blockchain requires high electricity consumption
3. NOT every blockchain is permissionless
4. Cryptocurrencies / cryptoassets are NOT sent or received, only the ownership info is updated in the blockchain

Misconceptions

Misconceptions

1. Cryptocurrencies do NOT use encryption
2. Not every blockchain requires high electricity consumption
3. NOT every blockchain is permissionless
4. Cryptocurrencies / cryptoassets are NOT sent or received, only the ownership info is updated in the blockchain
5. Cryptocurrencies / cryptoassets are NOT stored in digital wallets

CSEC 519



youtube.com/CihangirTezcan

Flu TV

youtube.com/c/ilkercanikligilflutv

Tokens

Tokens

Tokens are digital assets built on top of the blockchain via specialized smart contracts

Tokens

Tokens

Tokens are digital assets built on top of the blockchain via specialized smart contracts

1. Fungible tokens are identical and interchangeable (ERC-20)

Tokens

Tokens

Tokens are digital assets built on top of the blockchain via specialized smart contracts

1. Fungible tokens are identical and interchangeable (ERC-20)
2. Non-fungible tokens are unique and each token represents someone's ownership of a specific digital asset (ERC-721)

NFT

Non-Fungible Token (NFT)

- An NFT is an ownership record stored on a blockchain

NFT

Non-Fungible Token (NFT)

- An NFT is an ownership record stored on a blockchain
- Digital items such as pictures and videos are the most common assets traded as NFTs

NFT

Non-Fungible Token (NFT)

- An NFT is an ownership record stored on a blockchain
- Digital items such as pictures and videos are the most common assets traded as NFTs
- An NFT is the equivalent of a conventional proof-of-purchase, such as a paper invoice or an electronic receipt

NFT

Non-Fungible Token (NFT)

- An NFT is an ownership record stored on a blockchain
- Digital items such as pictures and videos are the most common assets traded as NFTs
- An NFT is the equivalent of a conventional proof-of-purchase, such as a paper invoice or an electronic receipt
- The NFT concept allows for the trading of digital assets between two mutually distrusting parties, as both the cryptocurrency payment and the asset transfer happen atomically in a single transaction

NFT

Non-Fungible Token (NFT)

- An NFT is an ownership record stored on a blockchain
- Digital items such as pictures and videos are the most common assets traded as NFTs
- An NFT is the equivalent of a conventional proof-of-purchase, such as a paper invoice or an electronic receipt
- The NFT concept allows for the trading of digital assets between two mutually distrusting parties, as both the cryptocurrency payment and the asset transfer happen atomically in a single transaction
- In Bitcoin like blockchains, this generally requires 2 transactions

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens
- When an NFT is created (minted), the creator can optionally associate a URL with the NFT

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens
- When an NFT is created (minted), the creator can optionally associate a URL with the NFT
- That URL (metadata_url) should point to a JSON file that conforms to the ERC-721 Metadata JSON Schema

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens
- When an NFT is created (minted), the creator can optionally associate a URL with the NFT
- That URL (metadata_url) should point to a JSON file that conforms to the ERC-721 Metadata JSON Schema
- The JSON file stores the details of the asset, e.g., its name and description, and also contains an image field storing a URL (image_url) that points to the asset

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens
- When an NFT is created (minted), the creator can optionally associate a URL with the NFT
- That URL (metadata_url) should point to a JSON file that conforms to the ERC-721 Metadata JSON Schema
- The JSON file stores the details of the asset, e.g., its name and description, and also contains an image field storing a URL (image_url) that points to the asset
- NFT essentially connects an asset with the record of its ownership

NFT

Non-Fungible Token (NFT)

- ERC-721 is the most popular standard for implementing NFTs
- Each NFT has its own *(tokenId)* to keep track of these unique tokens
- When an NFT is created (minted), the creator can optionally associate a URL with the NFT
- That URL (metadata_url) should point to a JSON file that conforms to the ERC-721 Metadata JSON Schema
- The JSON file stores the details of the asset, e.g., its name and description, and also contains an image field storing a URL (image_url) that points to the asset
- NFT essentially connects an asset with the record of its ownership
- ERC-721 standard does NOT record the hash of the file!!!

NFT Misconceptions

NFT Misconceptions

1. NFT images/videos are NOT stored in the blockchain

NFT Misconceptions

NFT Misconceptions

1. NFT images/videos are NOT stored in the blockchain
2. NFT is NOT copyright

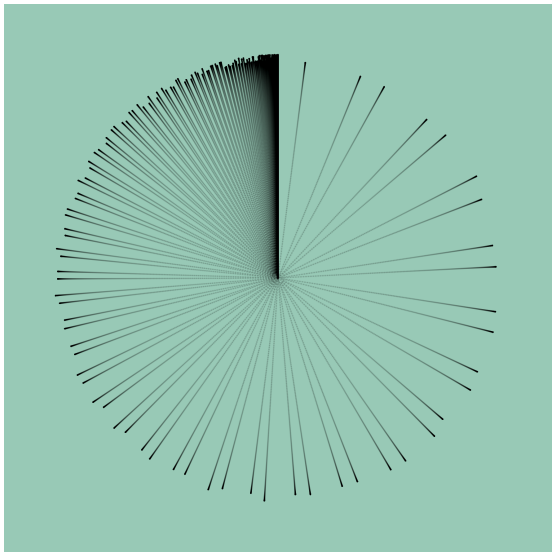
NFT Misconceptions

NFT Misconceptions

1. NFT images/videos are NOT stored in the blockchain
2. NFT is NOT copyright
3. NFT files may be lost or changed in the future

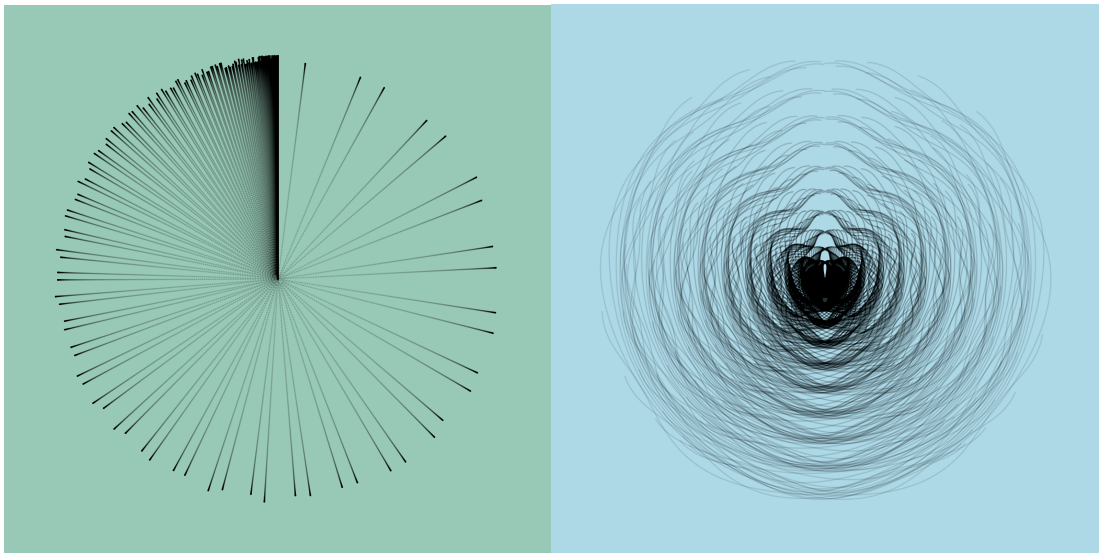
NFT Example 1

The same NFT by Moxie Marlinspike on Opensea, Rarible, and Metamask, respectively



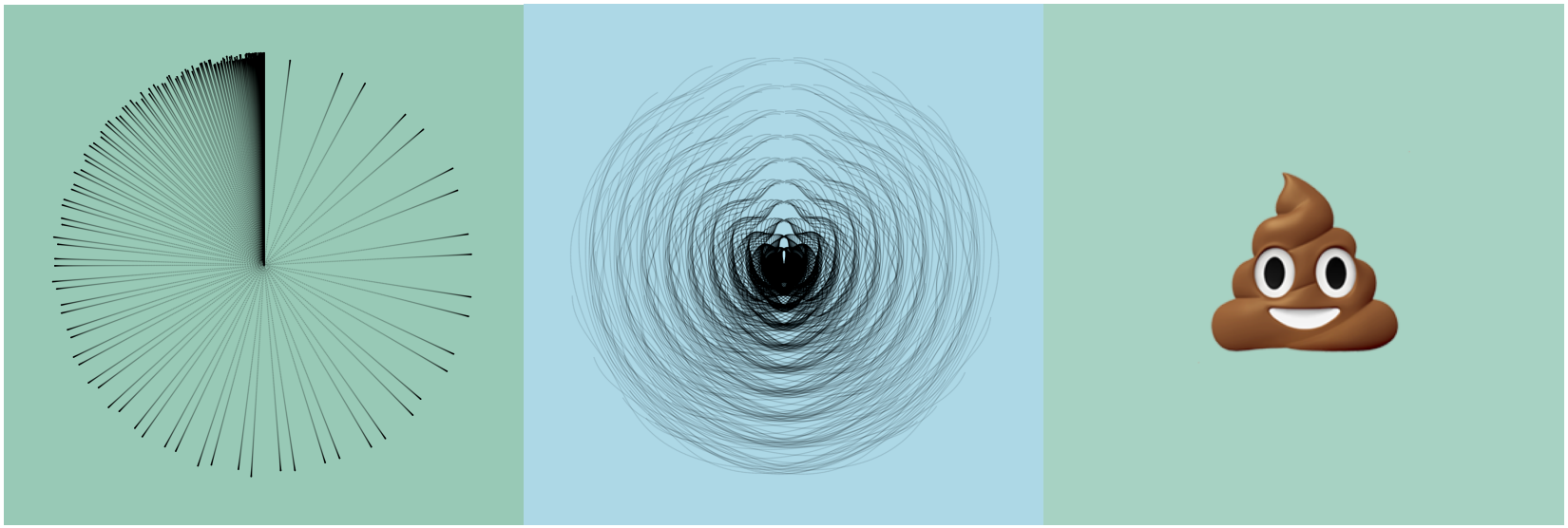
NFT Example 1

The same NFT by Moxie Marlinspike on Opensea, Rarible, and Metamask, respectively



NFT Example 1

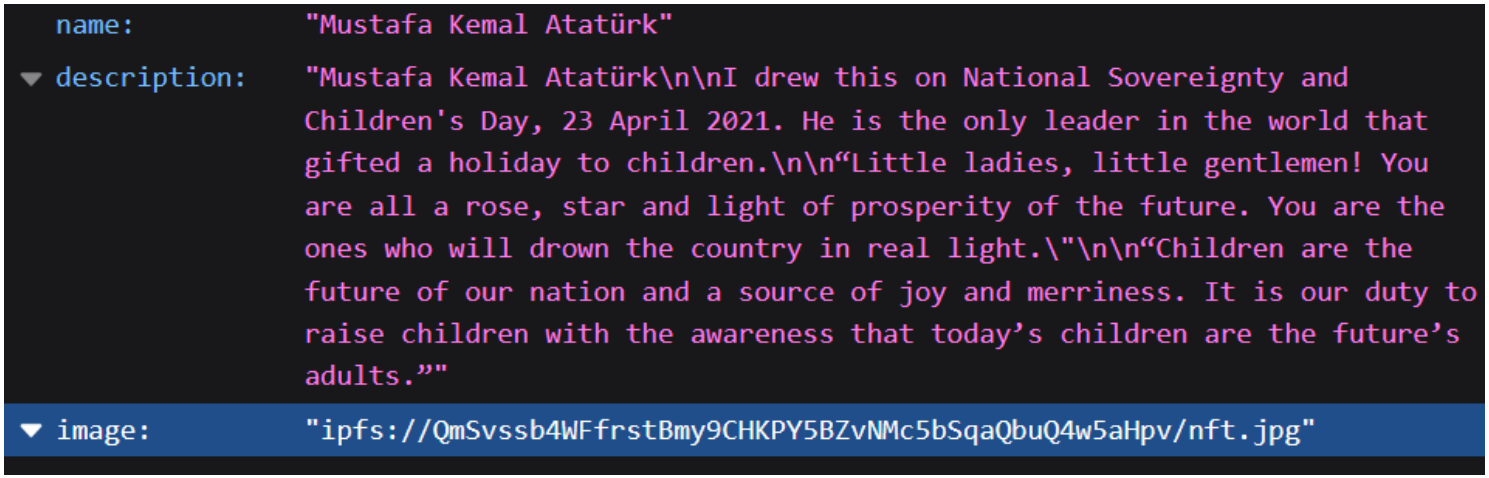
The same NFT by Moxie Marlinspike on Opensea, Rarible, and Metamask, respectively



Storing an NFT at a regular webpage (HTTP) is NOT a good idea

NFT Example 2

NFT Example 2



ipfs://QmbGNG3wes9tk7ori7Gu6mRcV9fwWvrzyJutqVQ3sCje99/metadata.json

NFT Example 2

IPFS link: ipfs://QmSvssb4WFfrstBmy9CHKPY5BZvNMc5bSqaQbuQ4w5aHpv/nft.jpg

NFT Example 2

IPFS link: ipfs://QmSvssb4WFfrstBmy9CHKPY5BZvNMc5bSqaQbuQ4w5aHpv/nft.jpg
IPFS SHA-256 (CID): 9772F58B3EA7BD44DB28A93DED464D6A46027B602B8C298F597FE7A9CC5B594B

NFT Example 2

IPFS link: ipfs://QmSvssb4WFfrstBmy9CHKPY5BZvNMc5bSqaQbuQ4w5aHpv/nft.jpg
IPFS SHA-256 (CID): 9772F58B3EA7BD44DB28A93DED464D6A46027B602B8C298F597FE7A9CC5B594B
SHA-256 of the file: 0BBB197E13566163C678EB2A05CE3703E89C82D517B609BC6A848E61DF9CD0FA

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD
3. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD
3. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata
   - Researchers observed that more than 100,000 out of 3 million metadata_urls were changed in 6 months

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD
3. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata
   - Researchers observed that more than 100,000 out of 3 million metadata_urls were changed in 6 months
4. 4 out of 12 million NFTs on OpenSea are actually inaccessible, but 68% of these images are still cached

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD
3. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata
   - Researchers observed that more than 100,000 out of 3 million metadata_urls were changed in 6 months
4. 4 out of 12 million NFTs on OpenSea are actually inaccessible, but 68% of these images are still cached
5. OpenSea and Rarible allow the creator to modify royalty after the sale

NFT Issues

NFT Issues

1. NFT Marketplaces (NFTMs) currently do not enforce
   - Know-Your-Customer rules
   - Anti-Money Laundering/Combating the Financing of Terrorism
2. Around 28% of the token contracts are closed-source (not verifiable)
   - Between June and December 2021, OpenSea took down 1765 closed-source tokens which account for $328.8M USD
3. The ERC-721 standard for NFTs actually allows for the possibility to change a token's metadata
   - Researchers observed that more than 100,000 out of 3 million metadata_urls were changed in 6 months
4. 4 out of 12 million NFTs on OpenSea are actually inaccessible, but 68% of these images are still cached
5. OpenSea and Rarible allow the creator to modify royalty after the sale
   - Around 9% of OpenSea royalty fees were modified after the 1st sale

Blockchain Use Cases

Blockchain Use Cases

- Since its introduction with Bitcoin in 2008, blockchain technology created a massive hype

Blockchain Use Cases

Blockchain Use Cases

- Since its introduction with Bitcoin in 2008, blockchain technology created a massive hype
- Yet after 14 years we have seen only a few meaningful use cases

Blockchain Use Cases

Blockchain Use Cases

- Since its introduction with Bitcoin in 2008, blockchain technology created a massive hype
- Yet after 14 years we have seen only a few meaningful use cases
- So far we have seen
  1. cryptocurrencies
  2. financial applications
  3. NFT

Blockchain Use Cases

Blockchain Use Cases

- Since its introduction with Bitcoin in 2008, blockchain technology created a massive hype
- Yet after 14 years we have seen only a few meaningful use cases
- So far we have seen
  1. cryptocurrencies
  2. financial applications
  3. NFT
- Many academic papers that propose blockchain solutions are wrong or redundant

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed
2. Data is contributed by more than one entity or auditing is required

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed
2. Data is contributed by more than one entity or auditing is required
3. Records are never updated or deleted after they are written

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed
2. Data is contributed by more than one entity or auditing is required
3. Records are never updated or deleted after they are written
4. Sensitive data will not be stored as plaintext

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed
2. Data is contributed by more than one entity or auditing is required
3. Records are never updated or deleted after they are written
4. Sensitive data will not be stored as plaintext
5. Control of the data store cannot be assigned to a single entity

Why You Do NOT Need a Blockchain

Why You Do NOT Need a Blockchain

A Blockchain might be useful if:

1. A shared and consistent data store is needed
2. Data is contributed by more than one entity or auditing is required
3. Records are never updated or deleted after they are written
4. Sensitive data will not be stored as plaintext
5. Control of the data store cannot be assigned to a single entity
6. Tamper-proof log of all data store writes is wanted

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain
- dApps allow anyone to participate without monetizing their data

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain
- dApps allow anyone to participate without monetizing their data

Advantages

1. Decentralized, peer-to-peer, no single point of failure

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain
- dApps allow anyone to participate without monetizing their data

Advantages

1. Decentralized, peer-to-peer, no single point of failure
2. Users own their data and content

WEB2 vs WEB3

WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain
- dApps allow anyone to participate without monetizing their data

Advantages

1. Decentralized, peer-to-peer, no single point of failure
2. Users own their data and content
3. Users can monetize their data and content

WEB2 vs WEB3

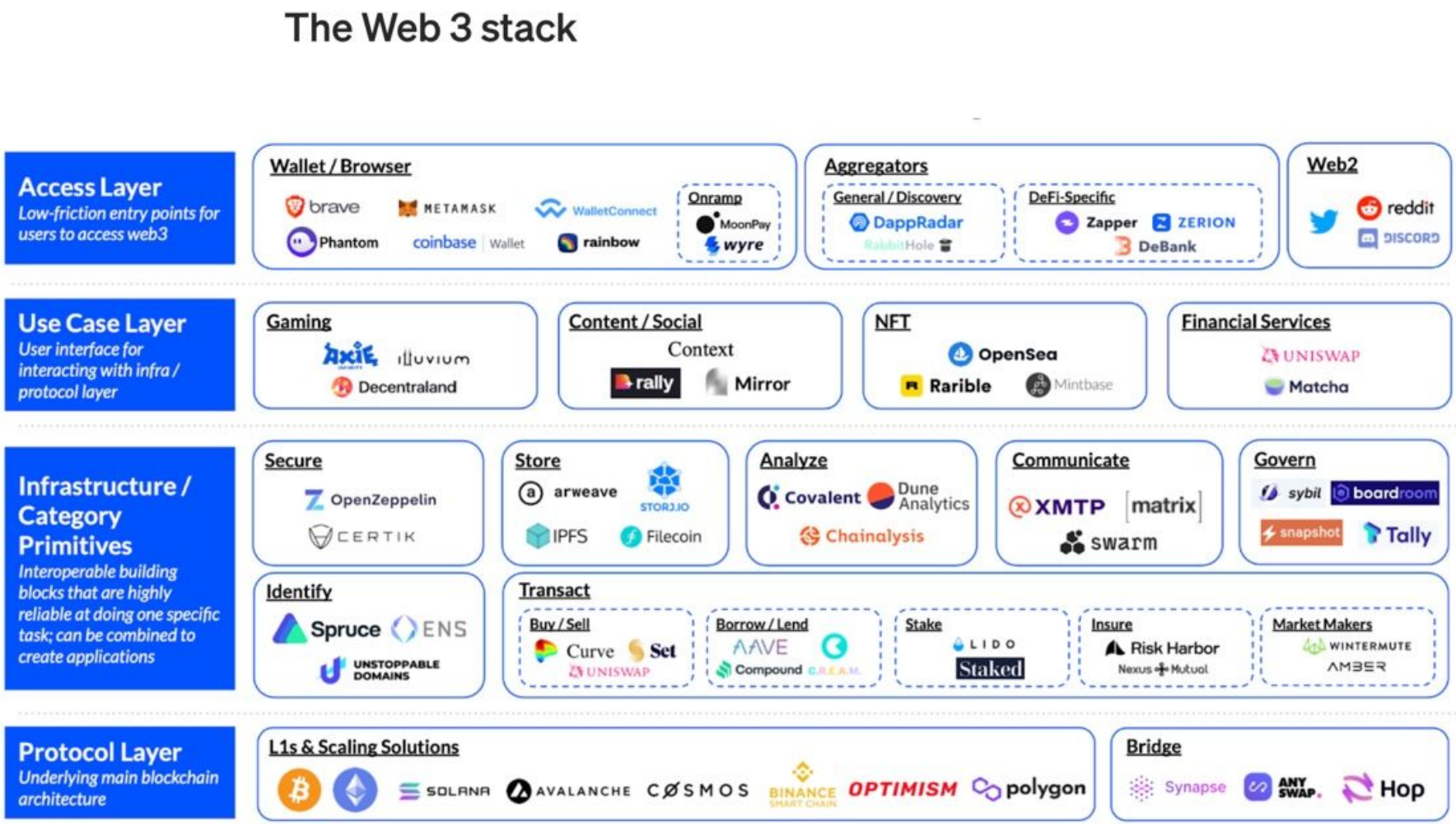
WEB3

- Current internet (Web2) is dominated by companies that provide *free* services in exchange for your personal data
- Because 'data is the new oil'
- Web3 refers to *decentralized apps* that run on the blockchain
- dApps allow anyone to participate without monetizing their data

Advantages

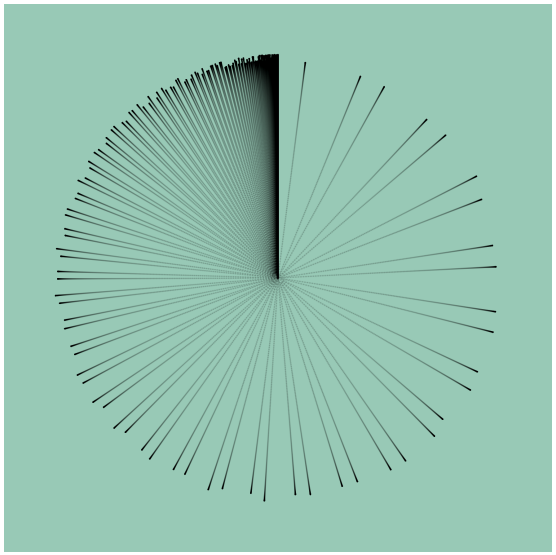
1. Decentralized, peer-to-peer, no single point of failure
2. Users own their data and content
3. Users can monetize their data and content
4. Private key can replace every password

WEB3


The Web 3 stack
Access Layer
Low-friction entry points for users to access web3
Wallet / Browser
brave
METAMASK
WalletConnect
Phantom
coinbase Wallet
rainbow
Onramp
MoonPay
wyre
Aggregators
General / Discovery
DappRadar
RabbitHole
DeFi-Specific
Zapper
ZERION
DeBank
Web2
reddit
DISCORD
Use Case Layer
User interface for interacting with infra / protocol layer
Gaming
Axie
illuvium
Decentraland
Content / Social
Context
rally
Mirror
NFT
OpenSea
Rarible
Mintbase
Financial Services
UNISWAP
Matcha
Infrastructure / Category Primitives
Interoperable building blocks that are highly reliable at doing one specific task; can be combined to create applications
Secure
OpenZeppelin
CERTIK
Store
arweave
STORJ.IO
IPFS
Filecoin
Analyze
Covalent
Dune Analytics
Chainalysis
Communicate
XMTP
matrix
swarm
Govern
sybil
boardroom
snapshot
Tally
Identify
Spruce
ENS
UNSTOPPABLE DOMAINS
Transact
Buy / Sell
Curve
Set
UNISWAP
Borrow / Lend
AAVE
Compound
Stake
LIDO
Staked
Insure
Risk Harbor
Nexus Mutual
Market Makers
WINTERMUTE
AMBER
Protocol Layer
Underlying main blockchain architecture
L1s & Scaling Solutions
SOLANA
AVALANCHE
COSMOS
BINANCE SMART CHAIN
OPTIMISM
polygon
Bridge
Synapse
ANY SWAP
Hop

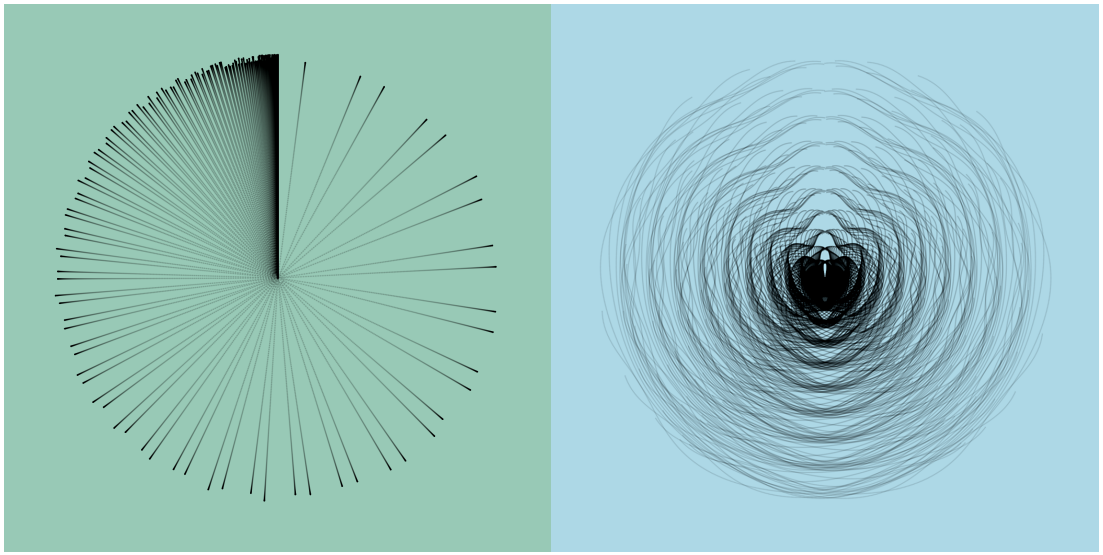
Back to NFT Example

The same NFT by Moxie Marlinspike on OpenSea, Rarible, and Metamask, respectively



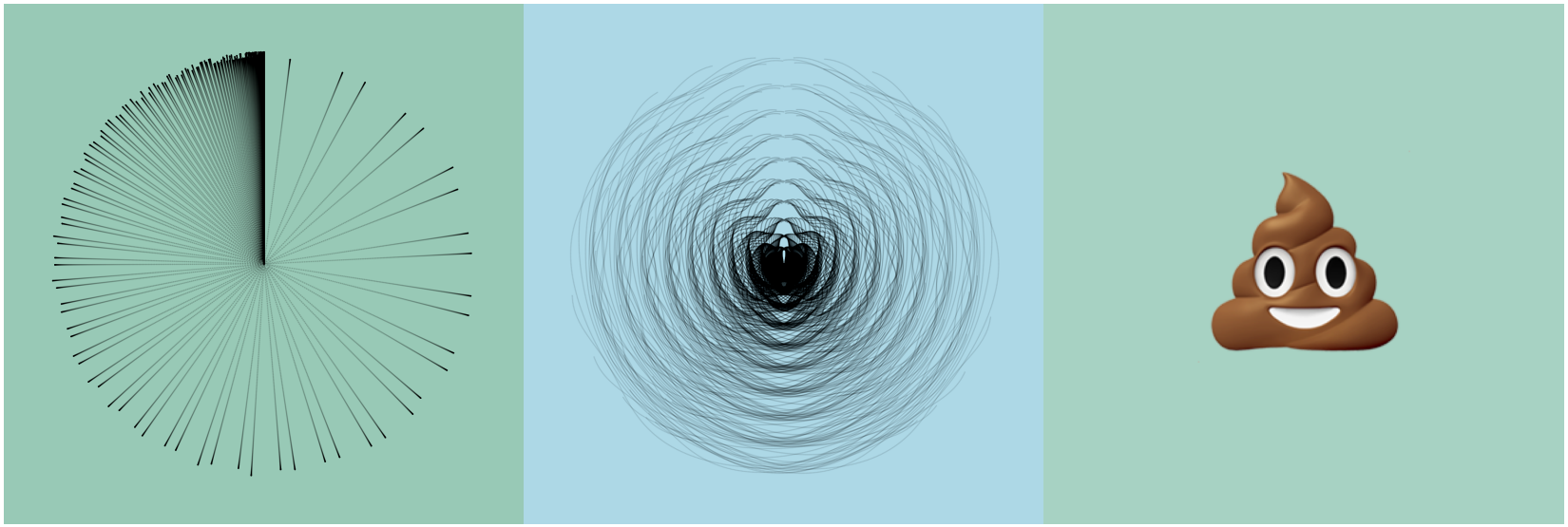
Back to NFT Example

The same NFT by Moxie Marlinspike on OpenSea, Rarible, and Metamask, respectively



Back to NFT Example

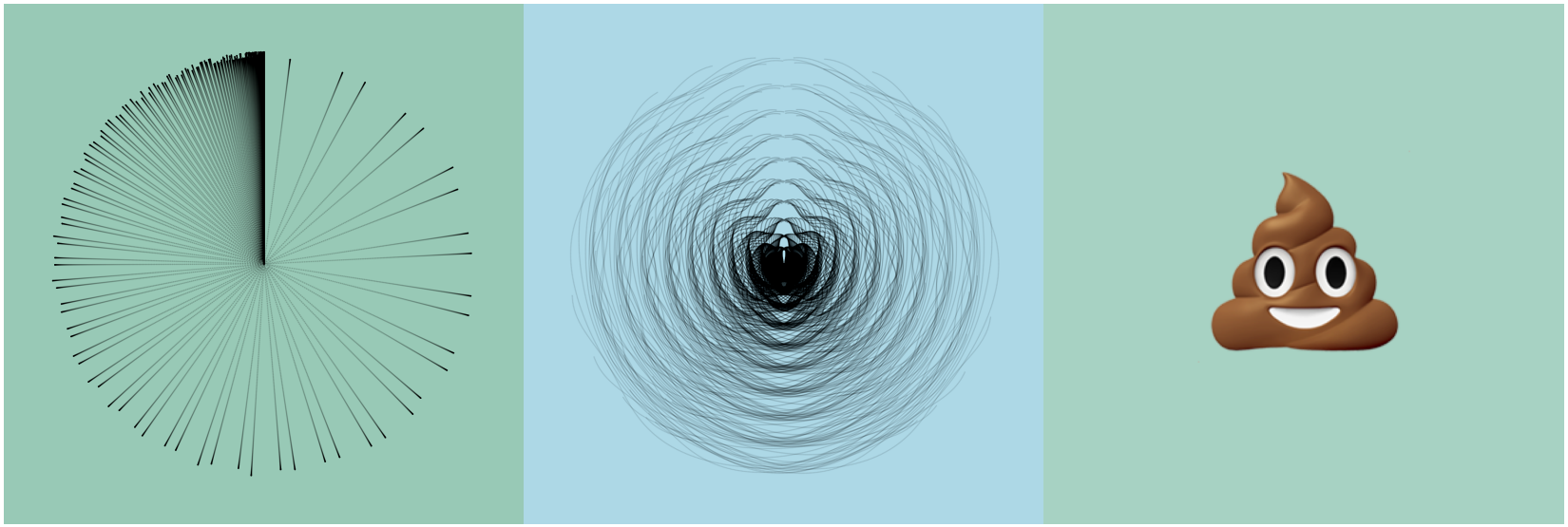
The same NFT by Moxie Marlinspike on OpenSea, Rarible, and Metamask, respectively



- For a short time OpenSea removed this NFT from their webpage

Back to NFT Example

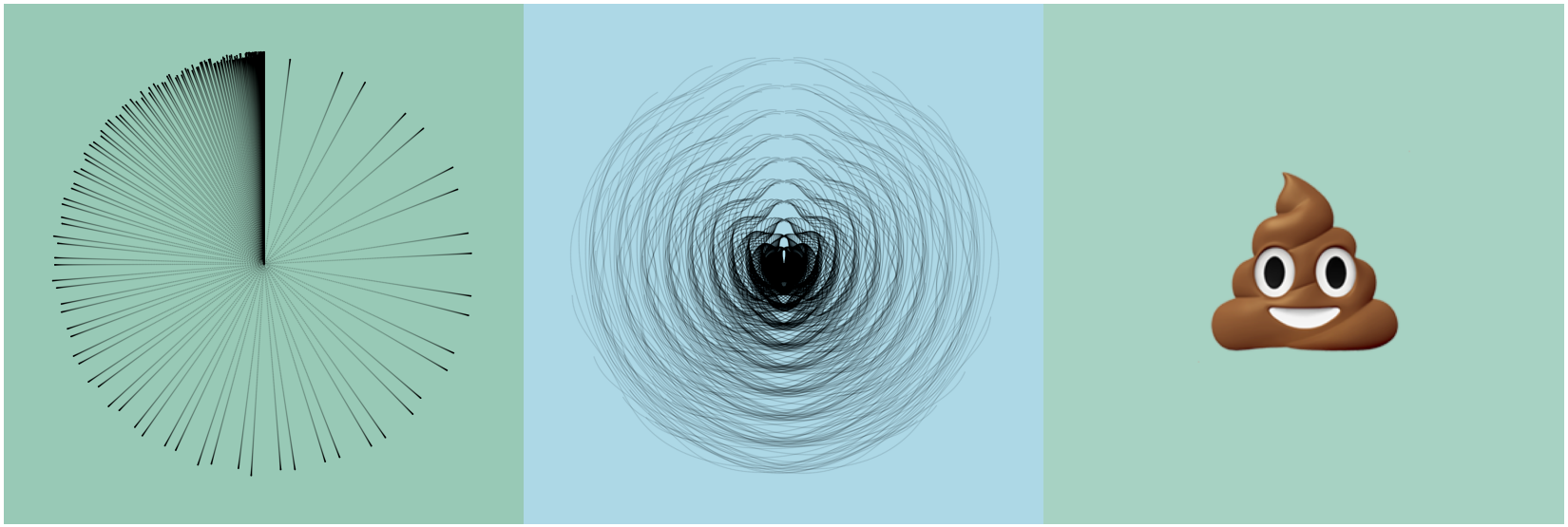
The same NFT by Moxie Marlinspike on OpenSea, Rarible, and Metamask, respectively



- For a short time OpenSea removed this NFT from their webpage
- This made NFT to disappear also from Metamask

Back to NFT Example

The same NFT by Moxie Marlinspike on OpenSea, Rarible, and Metamask, respectively



- For a short time OpenSea removed this NFT from their webpage
- This made NFT to disappear also from Metamask
- Instead of looking at the blockchain, using a library/API of others result in centralized systems

WEB3 Challenges

WEB3 Challenges

1. People do NOT like managing their own servers

WEB3 Challenges

WEB3 Challenges

1. People do NOT like managing their own servers
2. Decentralized apps turn into centralized apps when they share the same library/API

WEB3 Challenges

WEB3 Challenges

1. People do NOT like managing their own servers
2. Decentralized apps turn into centralized apps when they share the same library/API
3. Considering that hundreds of cryptocurrencies died in the last 10 years, assuming that a blockchain will live forever might be a huge assumption

Metaverse

Metaverse

The metaverse refers to digital worlds in which people will gather to work, play and hang out where you can own digital assets like NFTs and transact using cryptocurrencies

Metaverse

Metaverse

The metaverse refers to digital worlds in which people will gather to work, play and hang out where you can own digital assets like NFTs and transact using cryptocurrencies

Metaverse Misconceptions

Metaverse Misconceptions

1. Metaverse does NOT exist yet

Metaverse Misconceptions

Metaverse Misconceptions

1. Metaverse does NOT exist yet
2. There may be more than one Metaverse in the future

Metaverse Misconceptions

Metaverse Misconceptions

1. Metaverse does NOT exist yet
2. There may be more than one Metaverse in the future
3. Metaverse will NOT replace internet

Metaverse Misconceptions

Metaverse Misconceptions

1. Metaverse does NOT exist yet
2. There may be more than one Metaverse in the future
3. Metaverse will NOT replace internet
4. Metaverse does NOT mean VR/AR

Metaverse Challenges

Metaverse Challenges

1. Software development takes too much time

Metaverse Challenges

Metaverse Challenges

1. Software development takes too much time
2. Legal system is not ready for Metaverse

Metaverse Challenges

Metaverse Challenges

1. Software development takes too much time
2. Legal system is not ready for Metaverse
3. Services that do not provide convenience will not be preferred

Metaverse Challenges

Metaverse Challenges

1. Software development takes too much time
2. Legal system is not ready for Metaverse
3. Services that do not provide convenience will not be preferred
4. Some technologies emerge untimely and leave the scene

Thanks

Thank You for Your Attention