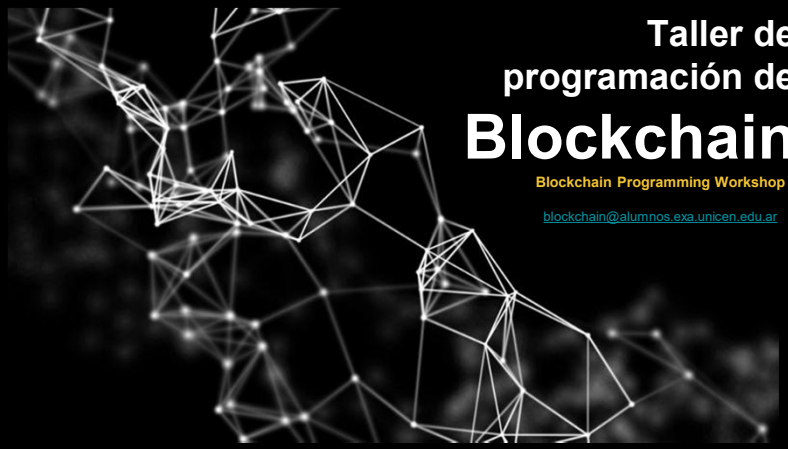


Taller de programación de Blockchain

Blockchain Programming Workshop

blockchain@alumnos.exa.unicen.edu.ar

1

:Schedule

1:
Introduction to Blockchain technology
2:
Peer-to-Peer Value Transfer System
3:
Blockchain as an application platform
4:
Smart contracts, Solidity and Web3
5:
Tools for the safe development of Dapps
6:
Blockchain as a coordination platform

2

3

Conflict

SQL (1970)

Documents & Internet

Bureaucracy & Connection

Data Philosophy

4

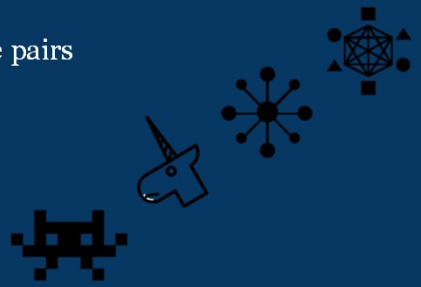
Solutions

Model of diverse pairs

Hub and Spoke

Protocols

Blockchain



5

Blockchain

*Incorruptible digital and **decentralized** ledger*

Can be programmed to record virtually everything of value

6

Why Decentralization Matters

Internet services on open protocols (1980-2000)

(2000-2010)

"Web 3"—the third era of the internet (2010-...)

7

Products - Processes

Microsoft® Encarta®

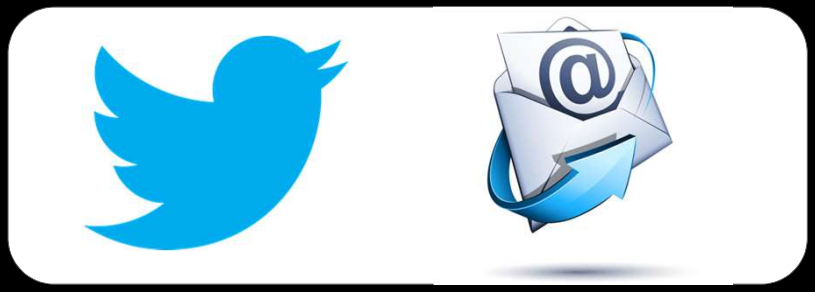
1993-2009

WIKIPEDIA

2000-

8

Rules changes



Spam filters

9

Information Governance

10

Cryptonetworks

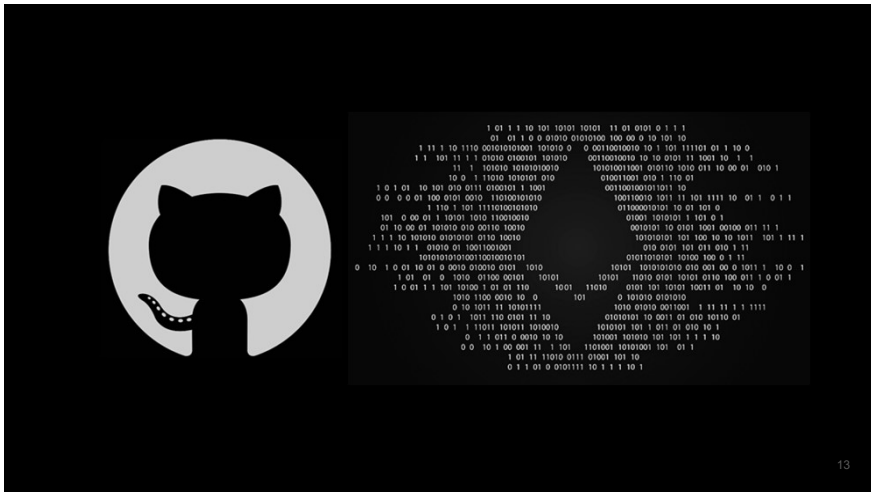
- Consensus Mechanisms
- Cryptocurrencies
- Open Source (public)
- Community Governance
- All together toward a common goal

11

Platform for Dapps

- Crowdfunding as a test bench
- Smart Contracts
- Global Computer
- Democratized access to "state of the art"

12



13

Minimum Viable Blockchain

The blockchain is agnostic to any "currency".

In fact, it will be adapted to power many other use cases.

14

15

	₿	$
User-facing		
Underlying system	Bitcoin protocol	Banking system

16

Protocol

- Anyone can add lines to the Ledger
- Settle up with real money each month

Ledger

Alice pays Bob $20
Bob pays Charlie $40
Charlie pays You $30
You pay Alice $10

} Tally up

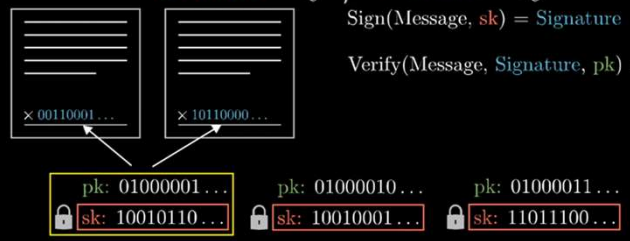
17

Signatures

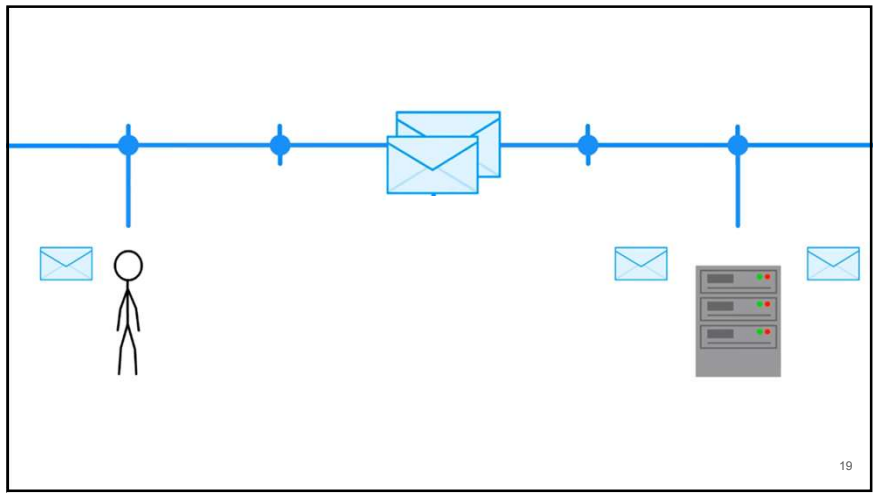
Secret key / Public key

Sign(Message, sk) = Signature

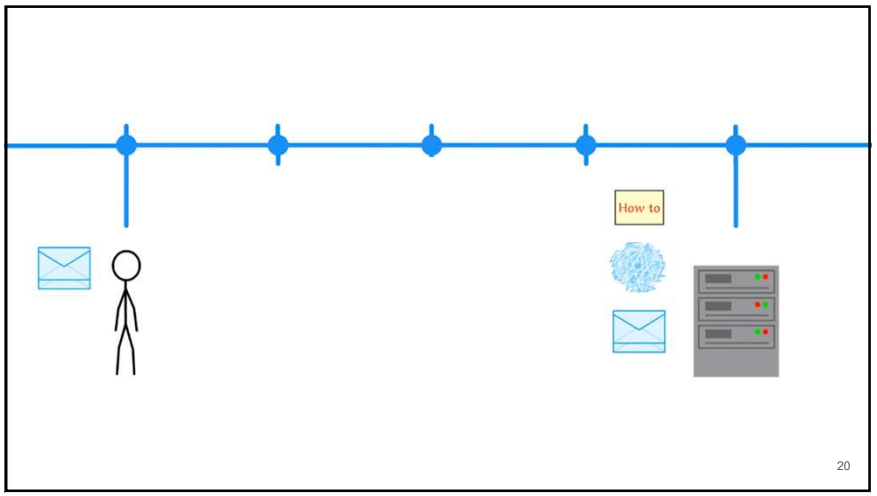
Verify(Message, Signature, pk) = T/F



18



19



20

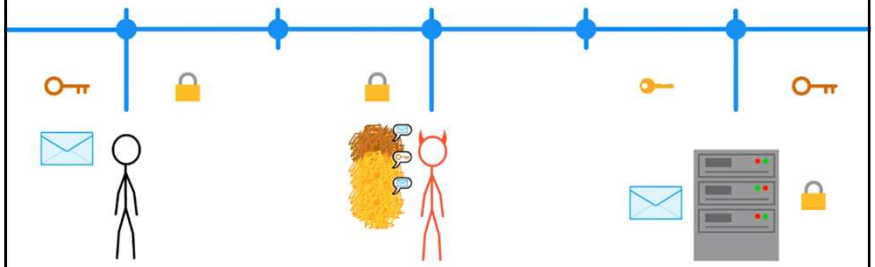
RSA

21

RSA

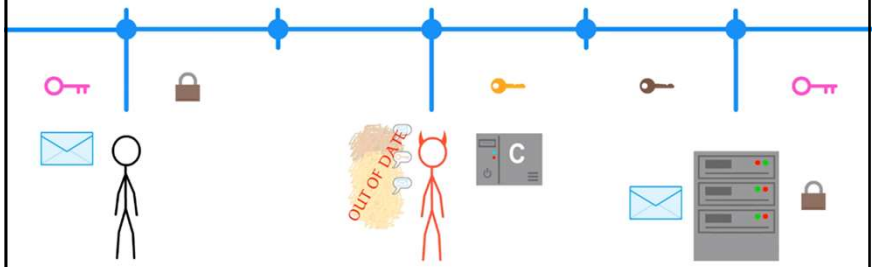
22

RSA

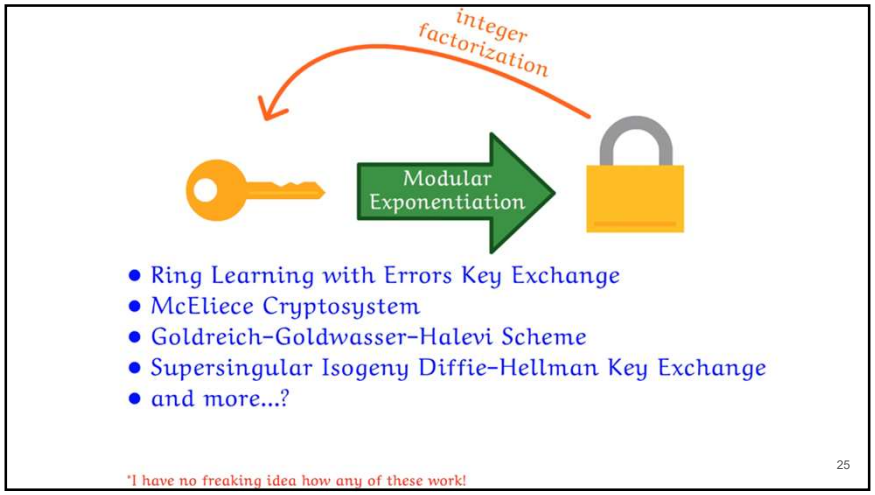


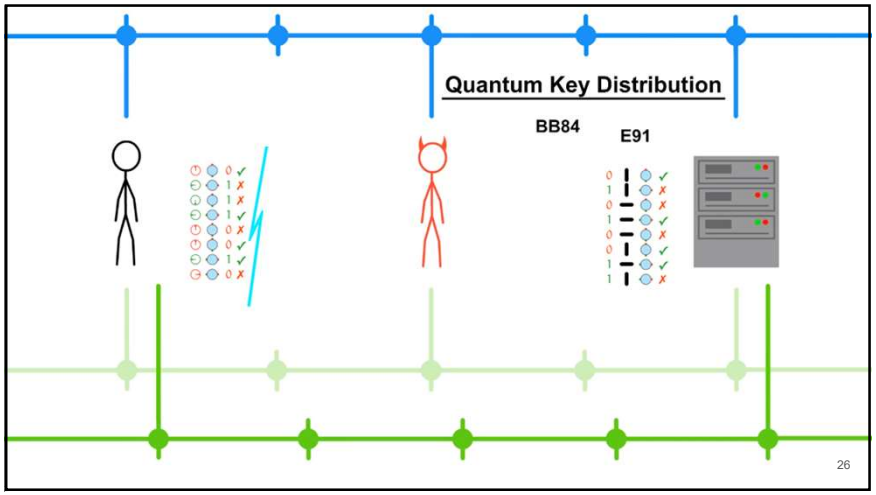
23

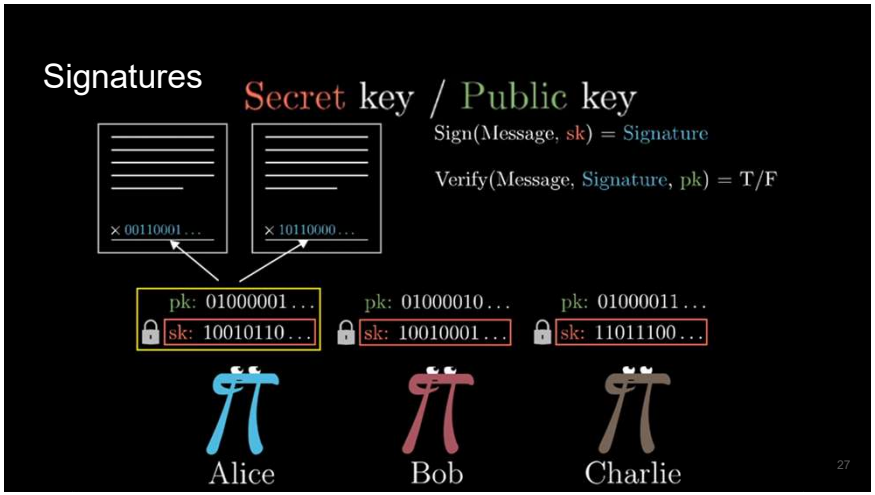
RSA

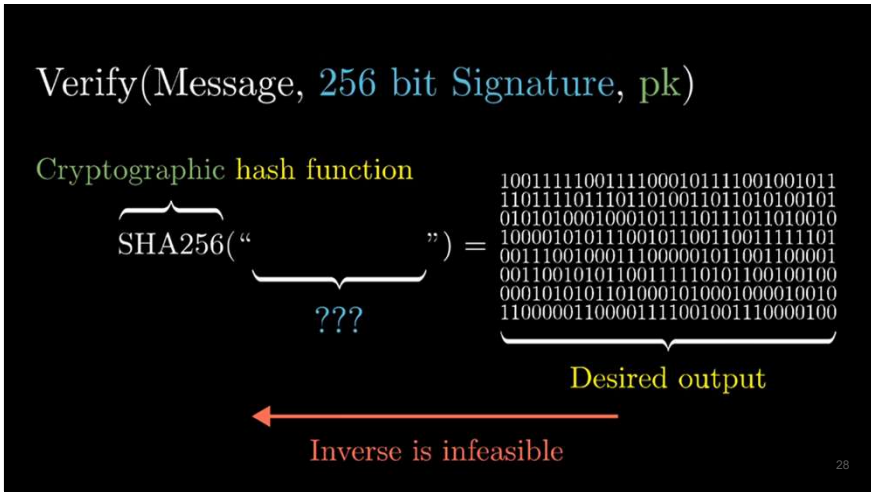


24


integer factorization
Modular Exponentiation
• Ring Learning with Errors Key Exchange
• McEliece Cryptosystem
• Goldreich-Goldwasser-Halevi Scheme
• Supersingular Isogeny Diffie-Hellman Key Exchange
• and more...?
*I have no freaking idea how any of these work!
25


Quantum Key Distribution
BB84
E91
26


Signatures
Secret key / Public key
Sign(Message, sk) = Signature
Verify(Message, Signature, pk) = T/F
× 00110001...
× 10110000...
pk: 01000001...
sk: 10010110...
pk: 01000010...
sk: 10010001...
pk: 01000011...
sk: 11011100...
Alice
Bob
Charlie
27


Verify(Message, 256 bit Signature, pk)
Cryptographic hash function
SHA256(" ??? ") =
1001111100111100010111100100101
1101111011101101001101101010010
0101010001000101111011011010010
1000010101110010110011001111101
0011100100011100000101100110000
0011001010110011111010110010010
0001010101101000101000100001001
1100000110000111100100111000010
Desired output
Inverse is infeasible
28

2^{256} possible signatures

00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000000000000000 2^{32} 4 Billion possibilities
00000000000000000000001010111000 2^{32} 4 Billion possibilities

29

( )(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)

Graphics Processing Unit
< 1 Billion Hashes/sec

4 Billion Hashes/sec

hash
hash
hash
hash
hash
hash
hash
hash
hash
hash

30

(4 Billion)( )(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)

H/s

Google ~ Millions of servers

KiloGoogle++

4 Billion

31

(4 Billion)(4 Billion)( )(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)

H/s

KG++

7.3 Billion people

4 Billion

32

(4 Billion)(4 Billion)(4 Billion)( )(4 Billion)(4 Billion)(4 Billion)(4 Billion)

$\frac{\text{H/s}}{\square}$ $\frac{\square}{\text{KG++}}$ $\frac{\text{KG++}}{\oplus}$

100 to 400 billion stars

4 Billion { ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⊕ ⋯ ⊕ ⊕ }

33

(4 Billion)(4 Billion)(4 Billion)(4 Billion)( )(4 Billion)(4 Billion)(4 Billion)

$\frac{\text{H/s}}{\square}$ $\frac{\square}{\text{KG++}}$ $\frac{\text{KG++}}{\oplus}$ $\frac{\oplus}{@}$

GigaGalactic Super Computer

2^{160} Hashes/sec

4 Billion { @ @ @ @ @ @ @ @ ⋯ @ @ }

34

(4 Billion)(4 Billion)(4 Billion)(4 Billion)(4 Billion)( )( )(4 Billion)

$\frac{\text{H/s}}{\square}$ $\frac{\square}{\text{KG++}}$ $\frac{\text{KG++}}{\oplus}$ $\frac{\oplus}{@}$ $\frac{@}{\text{GGSC}}$

4 Billion seconds $\approx$ 126.8 years

4 Billion $\times$ 126.8 years $\approx$ 507 Billion years

$\approx 37\times$ Age of universe

1 in 4 Billion chance of success

35

Total ₿ mining

5 Billion Billion $\frac{\text{Hashes}}{\text{Second}}$

$\frac{1}{3}$ KiloGoogle Application Specific Integrated Circuit

Trillion hashes/sec

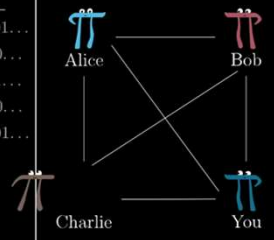
36

Protocol

- Anyone can add lines to the Ledger
- Settle up with real money each month
- Only signed transactions are valid

Ledger

1. Charlie pays Alice $100 00110001...
2. Charlie pays Bob $200 10110000...
3. Charlie pays You $800 00110011...
4. Charlie pays Bob $600 10110010...
5. Charlie pays Alice $900 00110101...



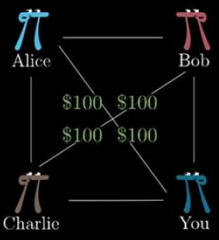
37

Protocol

- Anyone can add lines to the Ledger
- Only signed transactions are valid
- No overspending

Ledger

1. Alice gets $100
2. Bob gets $100
3. Charlie gets $100
4. You get $100



38

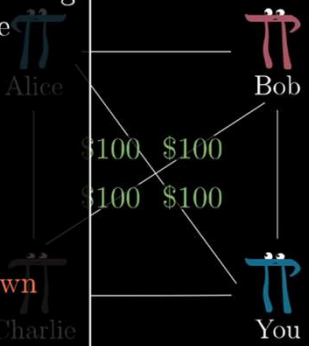
Ledger

1. Alice gets $100
2. Bob gets $100
3. Charlie gets $100
4. You get $100
5. Charlie pays Alice $50 00
6. Charlie pays Bob $50 10
7. Charlie pays You $20 00

Invalid

Charlie's running balance

$100
$50
$0
Overdrawn



39

Exchange LD for $$$

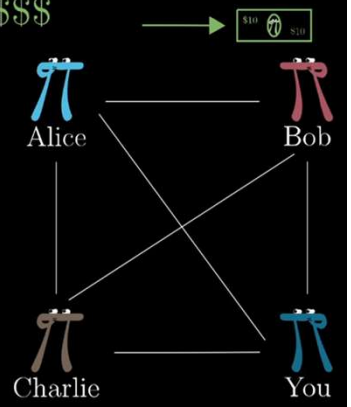
Ledger

⋮

Bob pays Alice 10 LD 00110001...

Currency = Transaction history

Ledger

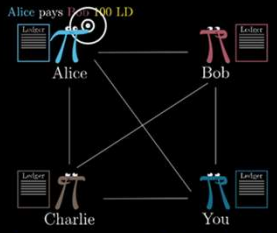


40

Protocol

- Broadcast transactions
- Only accept signed transactions
- No overspending

What to add here?



Main tool: Cryptographic hash functions

41

Main tool: Cryptographic hash functions

Ledger

Alice pays Bob 20 LD
Alice pays You 30 LD
Charlie pays You 100 LD
1073765433

"Proof of work"

SHA256

Probability: $\frac{1}{2^{30}} \approx \frac{1}{1,000,000,000}$

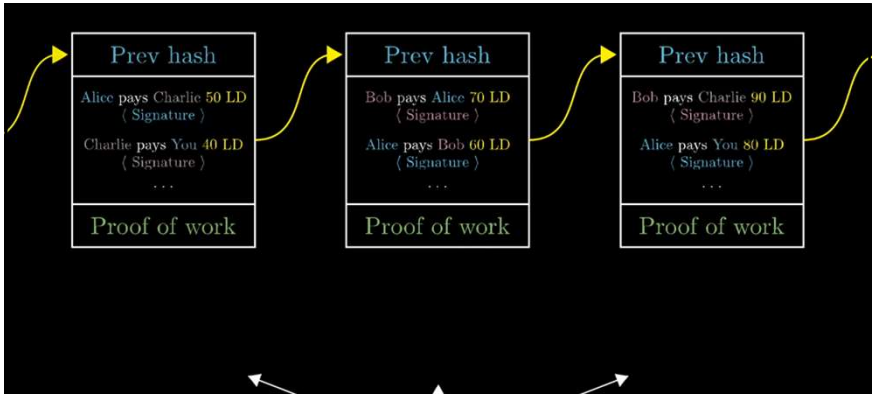
30 zeros

```
00000000000000000000000000000011
00110001011011101100100100110110
10000000010001100101101110100011
10111111100111000110010010111000
11011011101110010101101101000111
00011110001000001000100110000110
11100111000110100001100010010001
10000101100010011010000101000000
```

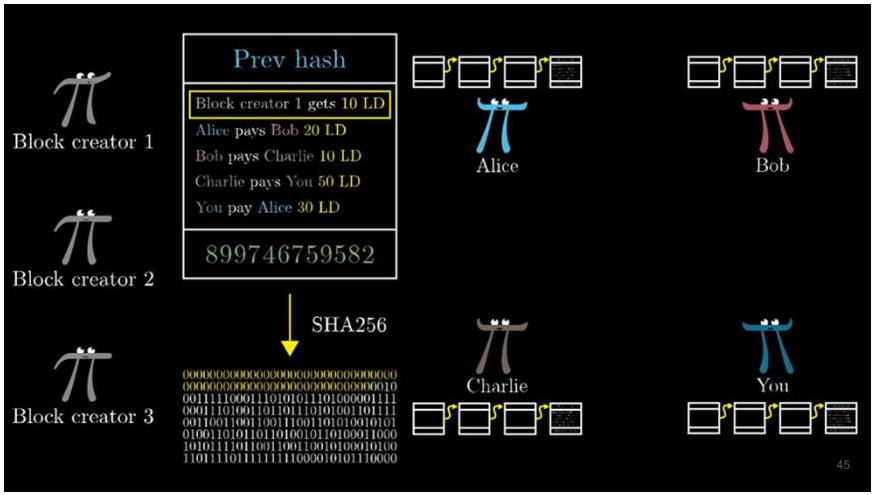
42

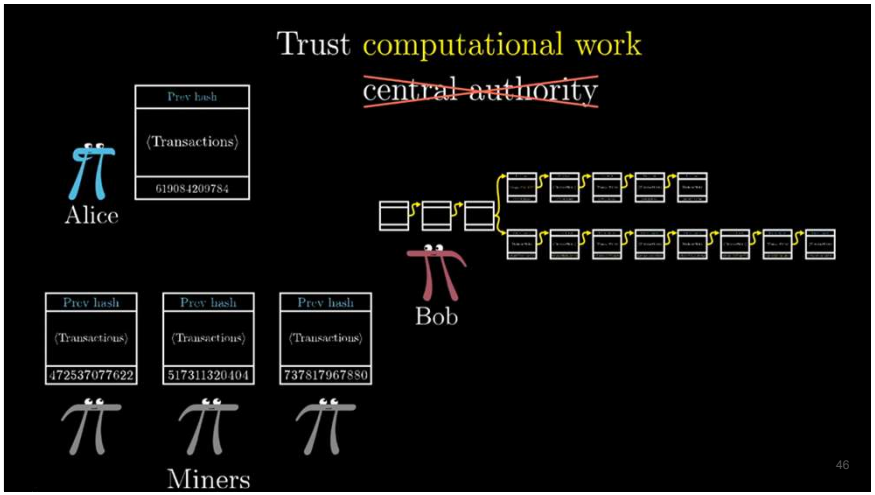
Main tool: Cryptographic hash functions

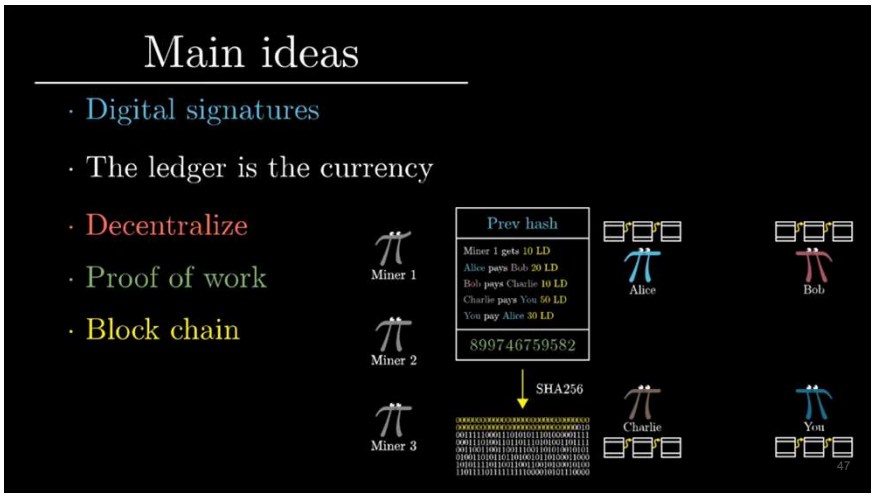
43



44

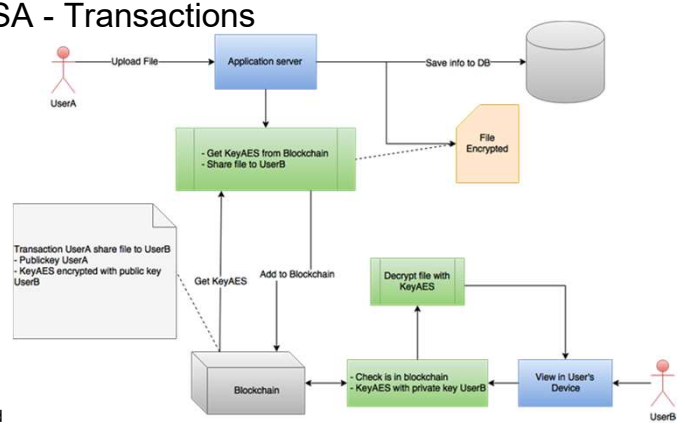

Prev hash
Block creator 1 gets 10 LD
Alice pays Bob 20 LD
Bob pays Charlie 10 LD
Charlie pays You 50 LD
You pay Alice 30 LD
899746759582
SHA256
Block creator 1
Block creator 2
Block creator 3
Alice
Bob
Charlie
You
45


Trust computational work
central authority
Prev hash
(Transactions)
619084209784
Alice
Bob
Prev hash
(Transactions)
472537077622
Prev hash
(Transactions)
517311320404
Prev hash
(Transactions)
737817967880
Miners
46


Main ideas
· Digital signatures
· The ledger is the currency
· Decentralize
· Proof of work
· Block chain
Miner 1
Miner 2
Miner 3
Prev hash
Miner 1 gets 10 LD
Alice pays Bob 20 LD
Bob pays Charlie 10 LD
Charlie pays You 50 LD
You pay Alice 30 LD
899746759582
SHA256
Alice
Bob
Charlie
You
47


Review
99B80F93F32MD2
Data:
" Hi im the first blocks data."
Previous Hash:
0
V890FDSUF8D9290
Data:
" Satoshi smells "
Previous Hash:
99B80F93F32MD2
3XSE9KASS9R0CK5
Data:
"...So this is a blockchain"
Previous Hash:
V890FDSUF8D9290
index
timestamp
data hash
previous hash
{
"from": "71238uqirbfh894-random-public-key-a-alkjdflakjfewn204ij",
"to": "93j4ivnqiopvh43-random-public-key-b-qjrgvnoeirbnferinfo",
"amount": Xcoins
}
48

RSA - Transactions



Advanced

49

Signature and verification

Asymmetric cryptography algorithm → keys mathematically linked.

public key : to encrypt / to verify digital signature

private key : to decrypt / to create a digital signature.

50

Balance = Σ(receipts)

Your wallets balance is the sum of all the unspent transaction outputs addressed to you.

Anyone can compute everyone's balance

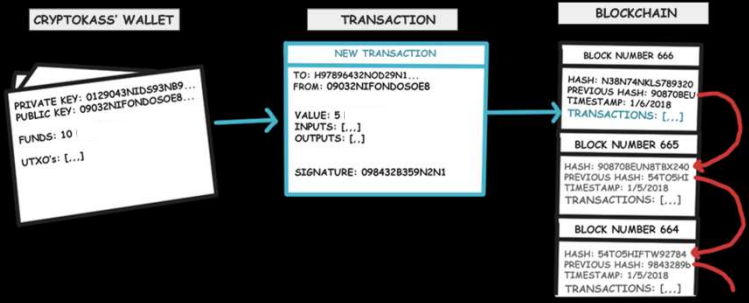
Anyone can traverse the full ledger, verify the trades

The system is trivial to audit

Transaction = TX

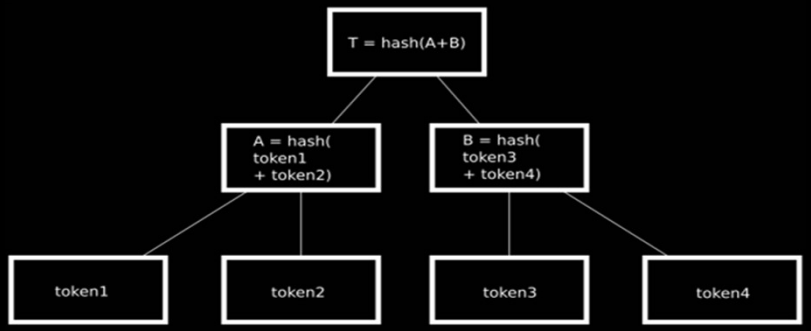
51

Transaction



52

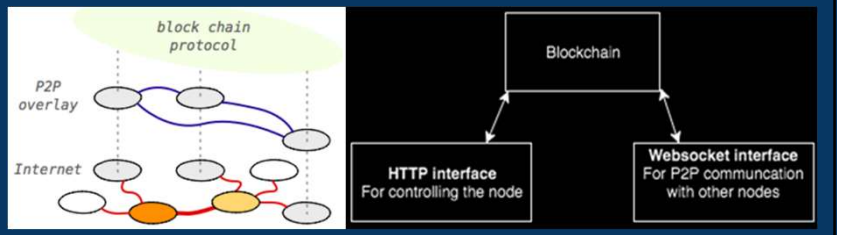
Merkle Hash Tree



¿ hash(hash(hash(contrato) + token2) + B) == T?

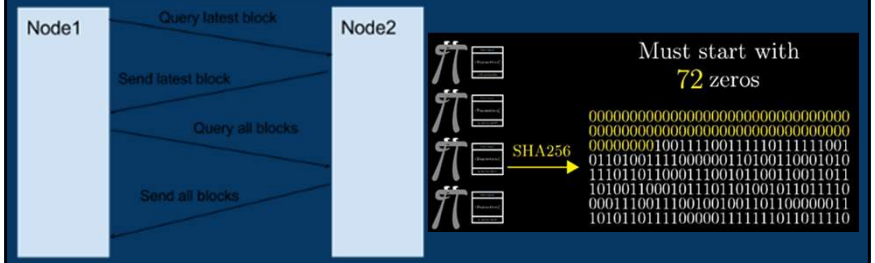
53

Net Arch.



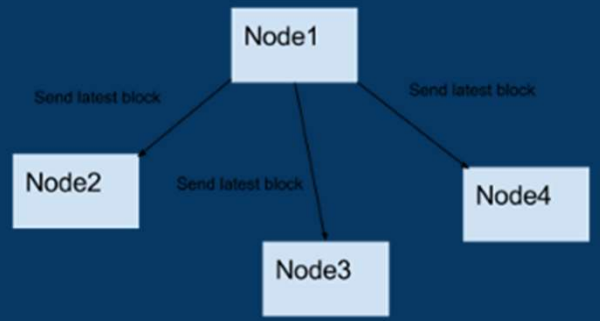
54

Node1 connects and syncs with Node2



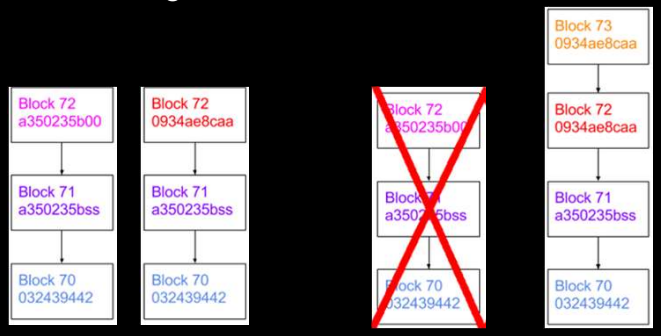
55

Node (Miner) generates a block and broadcasts it



56

Conflict - Finding New Chains



57

Don't trust yet

58

Still don't trust

59

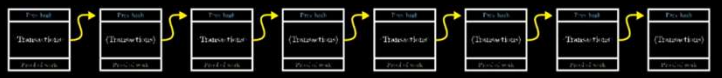
...a little more...

60

Maybe trust

61

Probably safe



62

Alright, you're good.

63

Proof-of-Work (PoW)

The OG Consensus

Popular implementations:
Bitcoin, Ethereum, Litecoin, Dogecoin, (Most of them)

Pros: We know it works

Cons: Slow throughput; killing the planet

64

Proof-of-Stake (PoS)

New kid on the block(chain)

Popular implementations:
Decred, Ethereum (soon), Peercoin

Pros:
Attacks more expensive; More decentralized; Energy efficient

Cons: Nothing at Stake

65

Delegated Proof-of-Stake (DPoS)

Elect your Validators

Popular Implementations:
Steemit, EOS, BitShares

Pros: Cheap transactions; scalable; energy efficient

Cons: Partially centralized

66

Proof-of-Authority (PoA)

Trust the know it all

Popular Implementations:
POA.Network, Ethereum Kovan testnet

Pros: High throughput; scalable

Cons: Centralized system

67

Proof-of-Weight (PoWeight)

Bigger is better

Popular Implementations:
Algorand, Filecoin, Chia

Pros: Customizable; scalable

Cons: Incentivization can be a challenge

68

Byzantine Fault Tolerance (BFT)

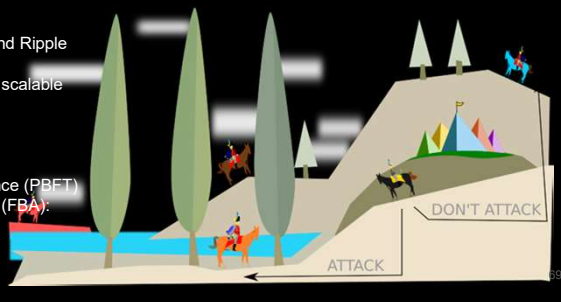
Siege the blockchain!

Popular Implementations:
Hyperledger, Stellar, Dispatch, and Ripple

Pros: High throughput; low cost; scalable

Cons: Semi-trusted

Variants:
Practical Byzantine Fault Tolerance (PBFT)
Federated Byzantine Agreement (FBA):



Directed Acyclic Graphs (DAGs)

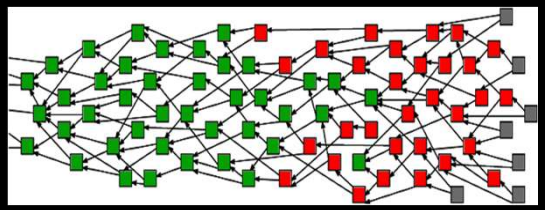
aka the Blockchain Killers!

Popular Implementations:
Iota, Hashgraph, Raiblocks/Nano

Pros: Network Scalability; low cost

Cons: Depends on implementation

Variants:
Tangle, Hashgraph,
Block-lattice, SPECTRE



Proof of Location Protocol (PoLP)

In development..

Field of application:
- Internet of Things
- Geospatial Data
- Supply Chain
- Mobility
- Insurance
- Location Intelligence

Proof of Curation Markets (PCM)

In development..

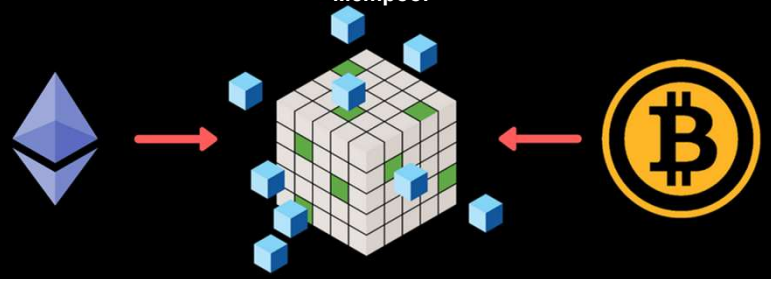
Field of application:
- IA Data
- Services

Blockchain Scalability

Transactions format

Block limits

Mempool



73

Average block time

- BTC: 10 minutes
- ETH: 15 Seconds
- XRP: 3.5 Seconds
- LTC: 2.5 Minutes

74

Block

Prev hash

Alice pays Bob 0.42 BTC
You pay Charlie 3.14 BTC
Bob pays You 2.72 BTC
Alice pays Charlie 4.67 BTC
⋮

Proof of work

Limited to
~ 2,400 transactions

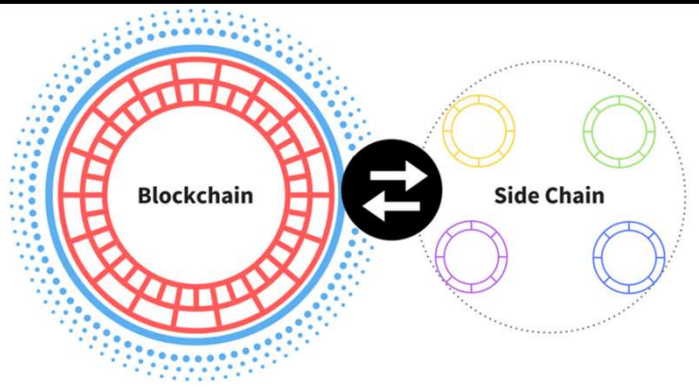
VISA Avg: 1,700/second
Max: > 24,000/second

Alice pays Bob 0.42 BTC
And leaves 0.001 BTC to the miner
⟨Alice's digital signature⟩

Incentivizes miner
to include

75

Blockchain Scalability



76

Trabajo Práctico [1] 💾

a) Implementar en Java la estructura básica de una blockchain: transacciones, bloques, firmas digitales, nodos.
b) Programar en Java los mecanismos básicos de validación de firma digital en transacciones y bloques.
c) Programar el proceso de seguimiento de balance de transacciones, no permitir transacciones en descubierto (sobre gasto).

77