Blockchains and Decentralized Applications

Thierry Sans

Forbes

LEADERSHIP

All Hype? What Every Business Leader Should Know About Web3

Sally Percy Contributor ⓘ

Follow

Aug 17, 2022, 03:00am EDT

1

Web3 could restore power to the people GETTY

Web3 is a decentralized, blockchain-based version of the internet. Experts predict that it will undermine the dominance of big tech by giving users more control over which services they access. As such, it could completely revolutionize how businesses operate.

So, what should leaders know about Web3?

From the article:

1. Web3 could restore "Power to the People"
2. Web3 could change the way we communicate with each other (perhaps)
3. Web3 could transform the infrastructure that all business runs on
4. Web3's impact could go beyond 'tech' companies
5. Web3 could be just another disruption for leaders to navigate

Forbes

FORBES DIGITAL ASSETS • EDITORS' PICK

Web3 Growth Stymied By Scarcity Of Programmers

Nina Bambysheva Forbes Staff

I cover cryptocurrencies and other applications of blockchain

Follow

Aug 29, 2022, 07:30pm EDT

Developer shortage stands in the way of decentralized, blockchain-centric internet bliss. GETTY

Advocates of Web3, a catch-all term widely used to incorporate concepts of decentralized networks, cryptocurrencies and other blockchain-powered applications, have a grand vision for the future of the Internet and global finance.

One thing that stands in the way: a lack of people to make it happen.

Cryptocurrencies

Total cryptocurrency market cap

The original Bitcoin paper (2008)

Bitcoin: A Peer-to-Peer Electronic Cash System

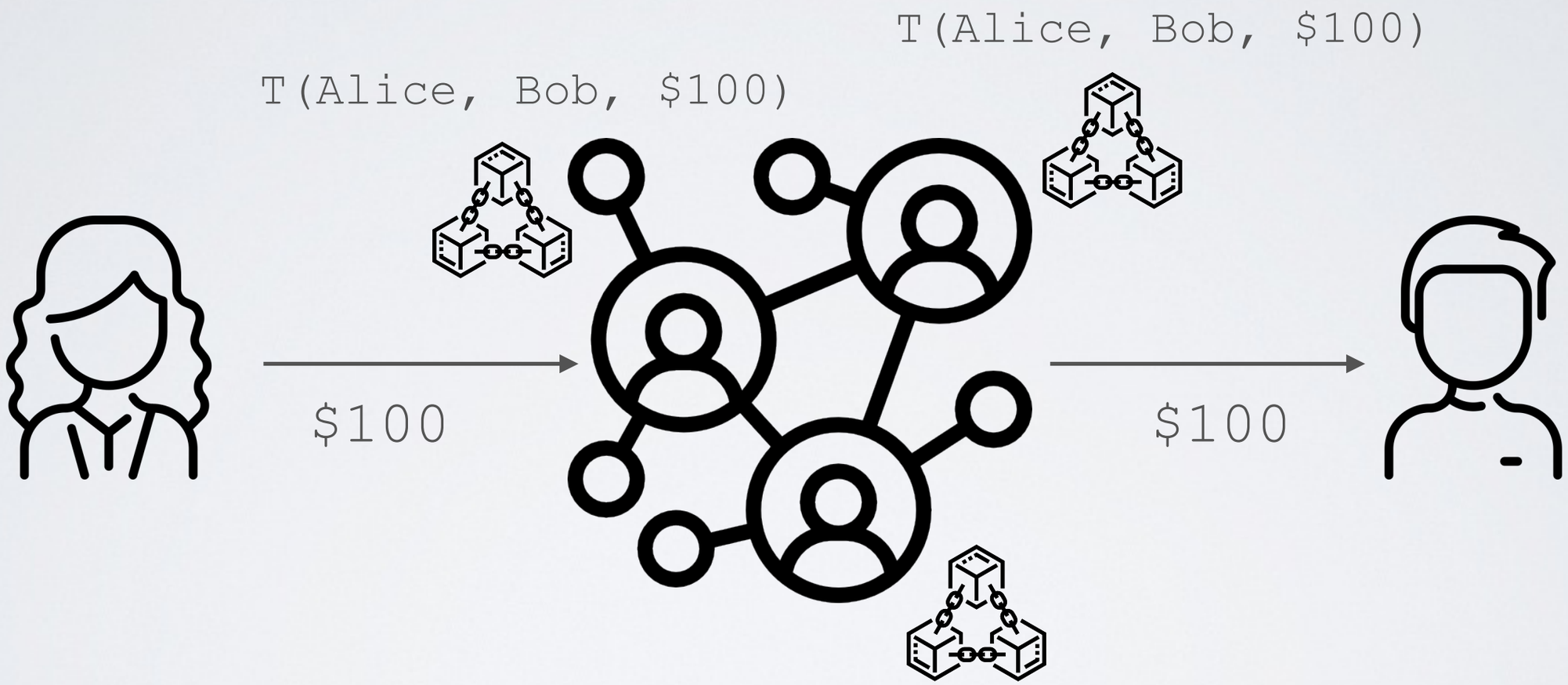
Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

A centralized ledger (Trusted Third Party)

T(Alice, Bob, $100)

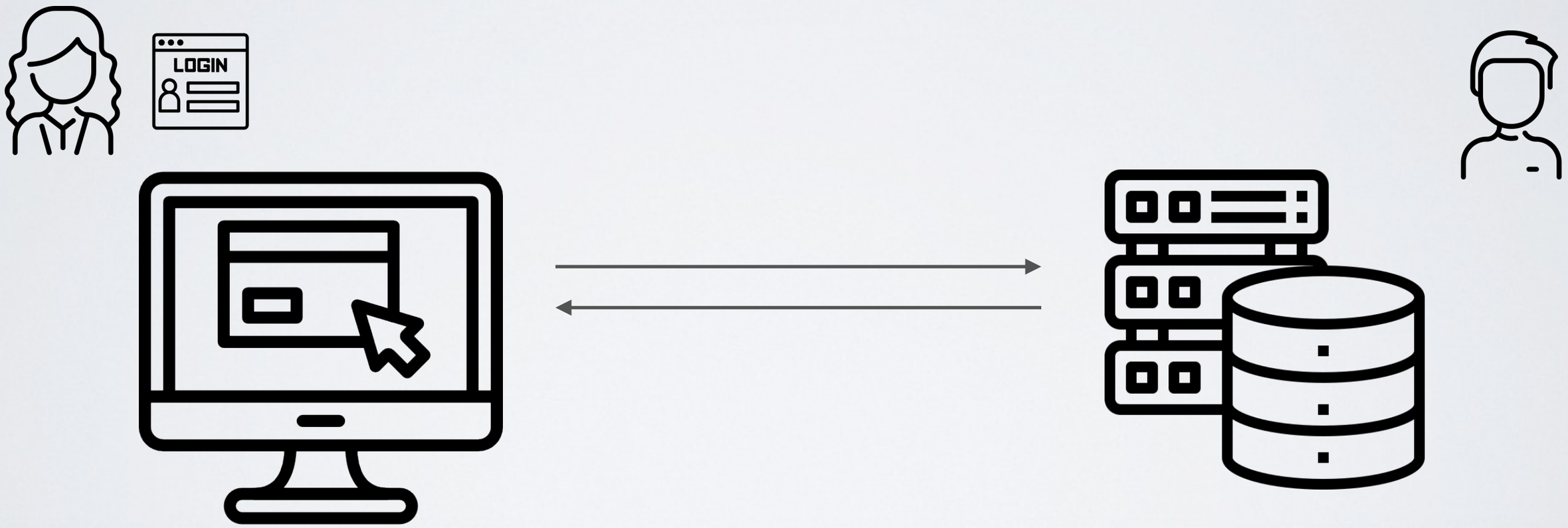
A decentralized ledger (Trustless)



Beyond Cryptocurrencies

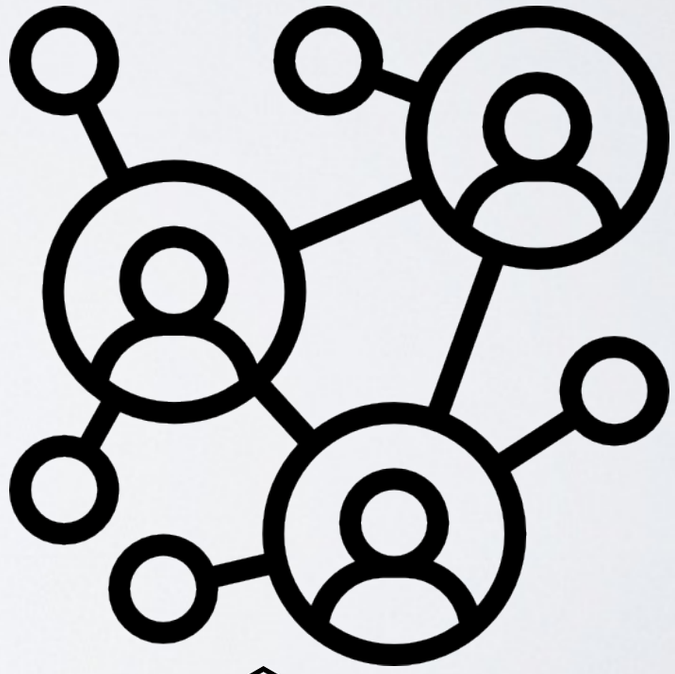
Towards Decentralized Applications

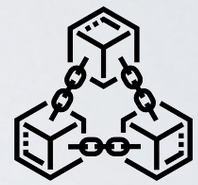
Web 2 - The centralized Web

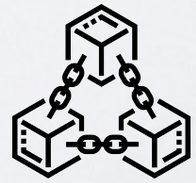


Web 3 - The **decentralized** web

smart contracts







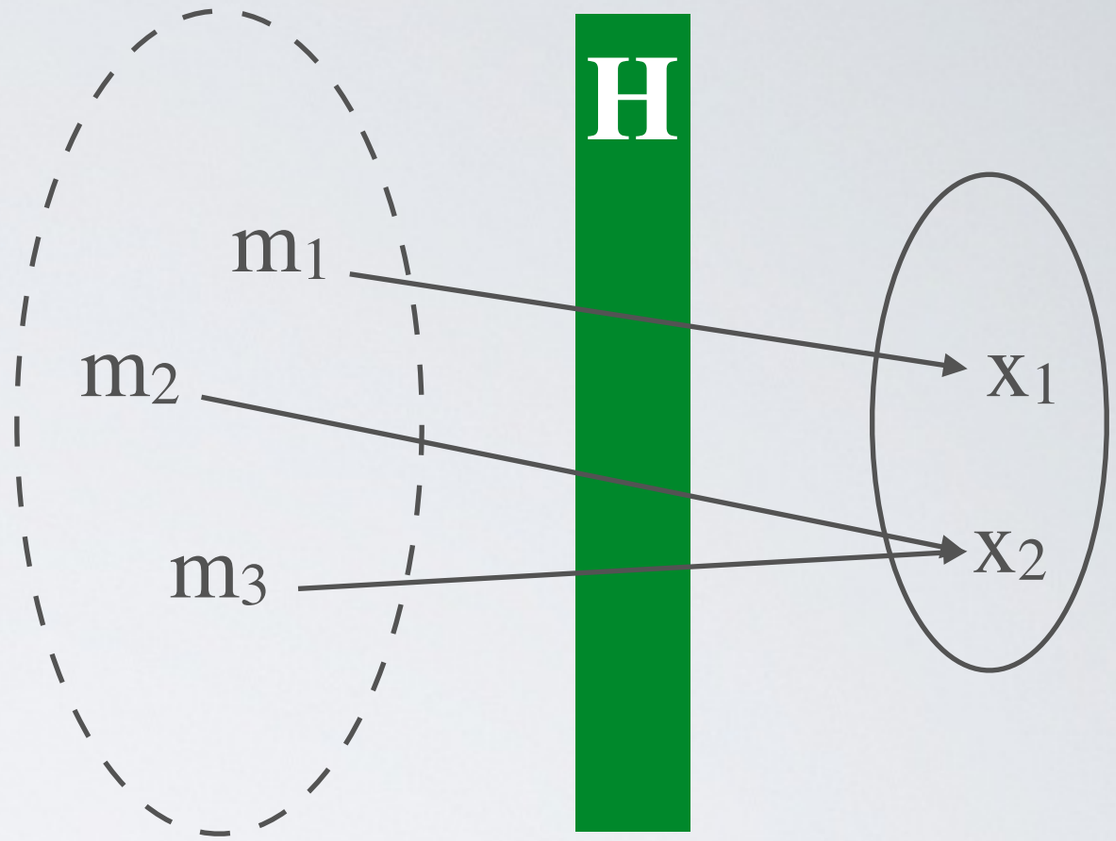
Cryptography Toolbox

The cryptography toolbox has many building blocks …

… but here we only need:

- Hashing
- Digital Signature

Cryptographic Hashing



$\mathrm{H}(\mathrm{m}) = \mathrm{x}$ is a hash function if

- m is a message of any length
- x is a message digest of a fixed length
- H is a non invertible function

➡ H is a lossy compression function
necessarily there exists x, m_1 and m_2 | $\mathrm{H}(\mathrm{m}_1) = \mathrm{H}(\mathrm{m}_2) = \mathrm{x}$

Computational Properties

✓ Given $\mathbf{H}$ and $\mathbf{m}$, computing $\mathbf{x}$ is **easy** (polynomial or linear)

- Given $\mathbf{H}$ and $\mathbf{x}$, computing $\mathbf{m}$ is **hard** (exponential)
- Given $\mathbf{H}$, $\mathbf{m}$ and $\mathbf{x}$, it is hard (exponential) to find $\mathbf{m'}$ such that $\mathbf{H(m) = H(m') = x}$
- Given $\mathbf{H}$, it is hard (exponential) to find $\mathbf{m}$ and $\mathbf{m'}$ such that $\mathbf{H(m) = H(m') = x}$

Digital Signatures

$(pk_A, sk_A) = generateKeyPair()$

$sig = sign(m, sk_A)$

pk_A

pk_A

m, sig, pk_A

$verify(m, sig, pk_A)$

Only Alice can sign a message $\mathbf{m}$ with her secret key $\mathbf{sk_A}$

➡ Everybody can verify $\mathbf{m}$ using Alice's public key $\mathbf{pk_A}$

Computational Properties

- ✓ $(pk, sk) = generateKeyPair()$
  is easy to compute (polynomial)

- ✓ $sig = sign(m, sk_A)$
  is easy to compute (either polynomial or linear)

- ✓ $verify(m, sig, pk_A)$
  is easy to compute (either polynomial or linear)

- ◉ Finding a matching key sk, given pk is hard (exponential)

- ◉ Forging a valid signature without knowing sk is hard (exponential)

Naive Cryptocurrency

Alice's Blockchain

oldH: null
from: null to: aGks5 amount: 100
newH: uUiN1

Genesis Block

Alice's Keypair

pk_B: aGks5

sk_B: ...

Alice's Tracker

aGks5: 100

...

Bob's Keypair

pk_B: bJR5H

sk_B: ...

Bob's Tracker

aGks5: 100

Bob's Blockchain

...

Alice's Blockchain

oldH: null
from: null to: aGks5 amount: 100
newH: uUiN1

oldH: uUiN1
from: aGks5 to: bJR5H amount: 20
newH: dSm3LJ

Genesis Block

Alice's Key pair

pk_B: aGks5
sk_B: ...

Alice's Tracker

aGks5: 80
bJR5H: 20

...

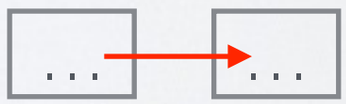
Bob's Keypair

pk_B: bJR5H
sk_B: ...

Bob's Tracker

aGks5: 80
bJR5H: 20

Bob's Blockchain



Alice's Blockchain

oldH: null
from: null to: aGks5 amount: 100
newH: uUiN1

Genesis Block

oldH: uUiN1
from: aGks5 to: bJR5H amount: 20
newH: dSm3LJ

oldH: dSm3LJ
from: bJR5H to: m406Z amount: 10
newH: 7fLvX

Alice's Key pair

pk_B: aGks5
sk_B: ...

Alice's Tracker

aGks5: 80
bJR5H: 10
m406Z: 10

...

...

Malory's Keypair

pk_M: m406Z
sk_M: ...

Malory's Tracker

aGks5: 80
bJR5H: 10
m406Z: 10

Mallory's Blockchain

Bob's Keypair

pk_B: bJR5H
sk_B: ...

Bob's Tracker

aGks5: 80
bJR5H: 10
m406Z: 10

Bob's Blockchain

Concurrency issues

Alice's Key pair

pk_B: aGks5
sk_B: ...

oldH: 7fLvX
from: aGks5 to: m406Z amount: 10
newH:P+q2C

Solution: consensus algorithm

Malory's Keypair

pk_M: m406Z
sk_M: ...

Malory's Tracker

aGks5: ?
bJR5H: ?
m406Z:?

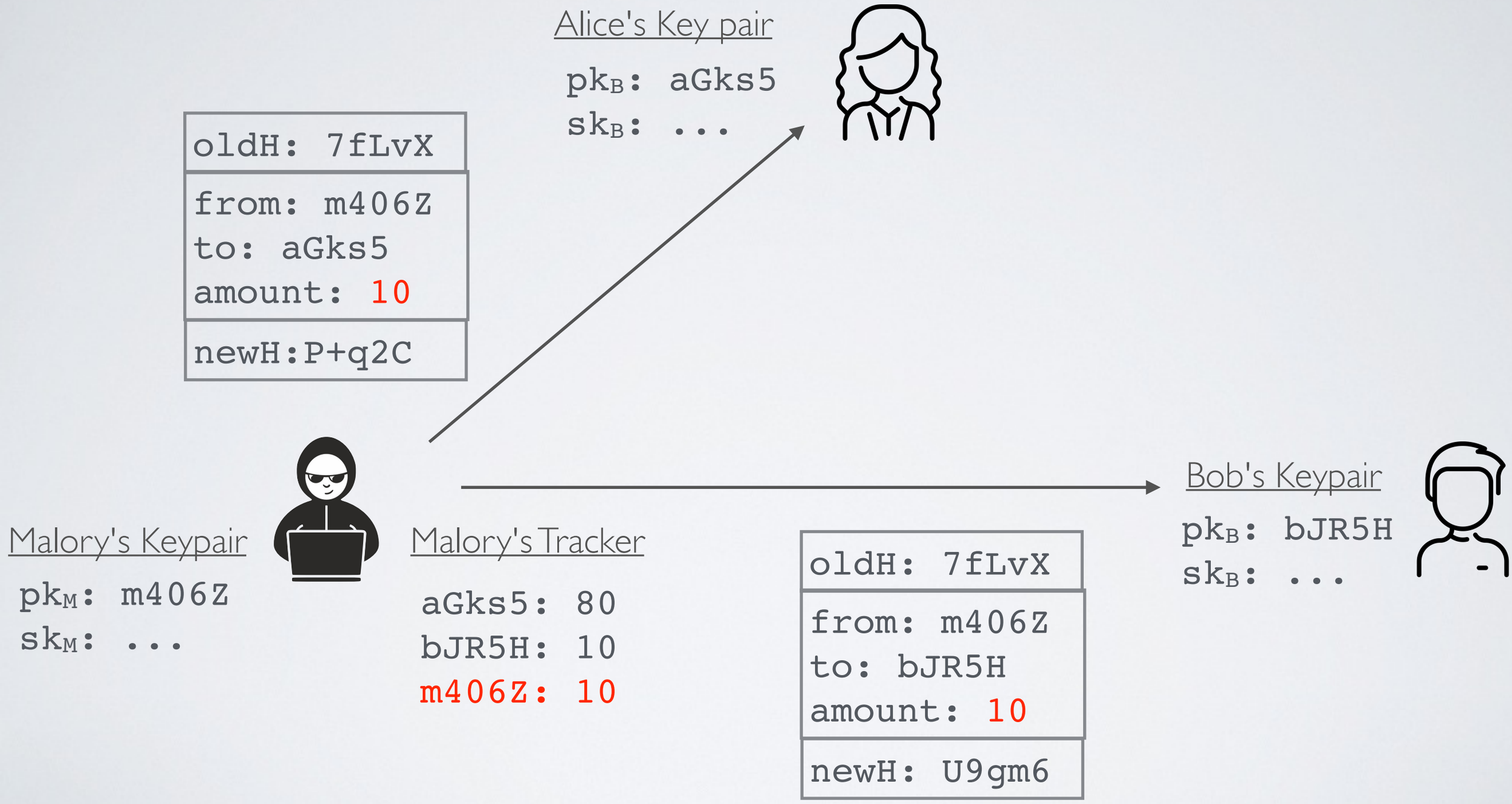
Mallory's Blockchain

Bob's Keypair

pk_B: bJR5H
sk_B: ...

oldH: 7fLvX
from: bJR5H to: m406Z amount: 10
newH: vVxL6

Double Spending Attack



The original Bitcoin paper (2008)

Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.