Dionysis Zindros
National Technical University of Athens 2012

What is bitcoin?

- **Digital currency**
- For very real **online payments**
- **Replacement** (?) for **€** and **$**

History

- **Wei Dai**, 1998: "Bmoney" (cypherpunks)
- **Satoshi Nakamoto**, 2009: "Bitcoin: A Peer-to-Peer Electronic Cash System"
- 2009: bitcoind **open source** in C++

Problem: Online payments

- A trusted authority is required
- Payments with **credit cards**
- **e.g. Visa, MasterCard**
- Or services such as **PayPal**
- **No anonymity**
- **Cost** for the services
- Inability for small amounts

Problem

- We could use **gold** – objective value
- Hard to use
- **Slow**
- Inconvenient
- Dangerous

Problem

- **People dislike central control**
- **€** and **$** are **centrally controlled**
- Government control of the economy may be undesired
- **Centrally controlled inflation**

Many people do not trust their government for managing the economy.

Solution

- A digital currency **bitcoin**
- **Peer-to-peer** network

Advantages

- **Fast** payments (< 10’)
- **No** central authority
- **Free market** exchange rates
- **Secure** transactions
- **Anonymity**

Disadvantages? From a government perspective…

- People are going to use **bitcoin** anyway
  - Because bitcoin is a fundamentally **good** idea
  - Its technology makes it hard to illegalize
- It’s hard to track
  - People don’t want to be tracked by the government
- But bad things can happen
  - Fraud
  - Money laundering
- How can a government
  - Ensure safety and security?
  - Avoid fraud?
  - Maintain a growing economy for the nation?

Purpose of this talk

- Present bitcoin as it is today
- Illustrate what it is from the point of its creators and users
  - What problems it solves and how
- Discuss how the government fits into this scheme
  - In an evolving crypto-economy
  - What can a government do?

From a government perspective…

- Some of the bitcoin creators and users don’t like governments
- Bitcoin is inherently an economy based on anarchy
- Many governments don’t like bitcoin
- But a government needs to know what bitcoin is
- It cannot be ignored; it cannot be easily illegalized
- If bitcoin creates problems for the government, we need to discuss how to solve them

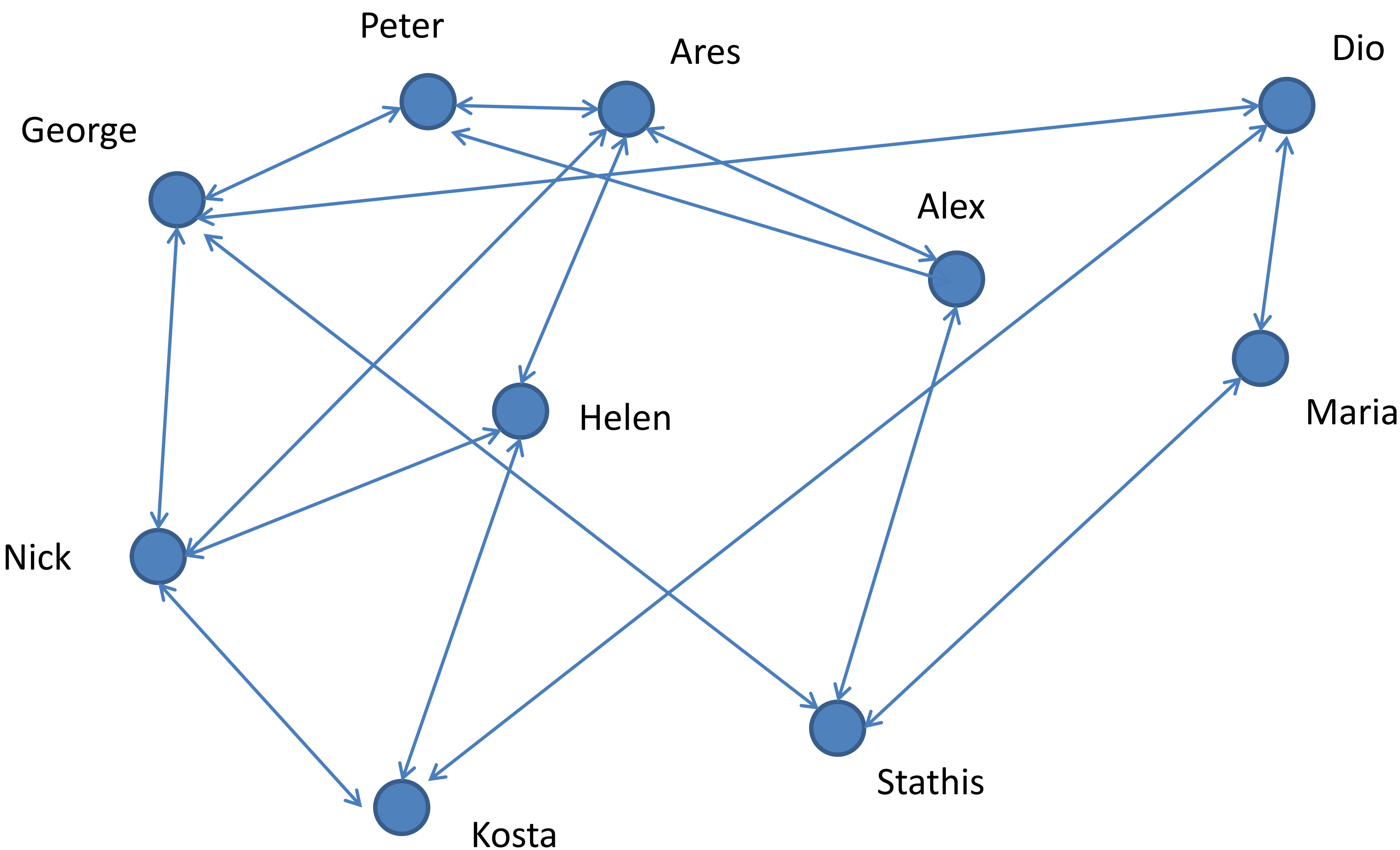
The basic idea

- Modern currencies **$** and **€**
- They’re **virtual** – no **real** value
- They can be **any object**
- Providing it cannot be cloned
- We agree, as a nation, to make a piece of **paper** into a **currency**

This doesn’t require a central authority!

...cryptography replaces
the central authority

The bitcoin peer-to-peer network



Authentication

- Every **node** has a **private/public key**
- This ensures that **whoever** has the money, **it's them who make payments**
- **Public key** is **broadcasted** to the network
- Private key is stored locally on the node

Bob **Alice**

Has 12BTC Has 0BTC

$m \leftarrow$ "Send 12BTC to Alice"

$h \leftarrow H(m)$

$s \leftarrow \mathrm{sign}_{SB}(h)$

$$\xrightarrow{\quad s \quad}$$

$\mathrm{verify}_{PB}(h)$

Has 0BTC

Has 12BTC

Validity

- How do we ensure that the coin came from a **valid source** and is not **self-made?**

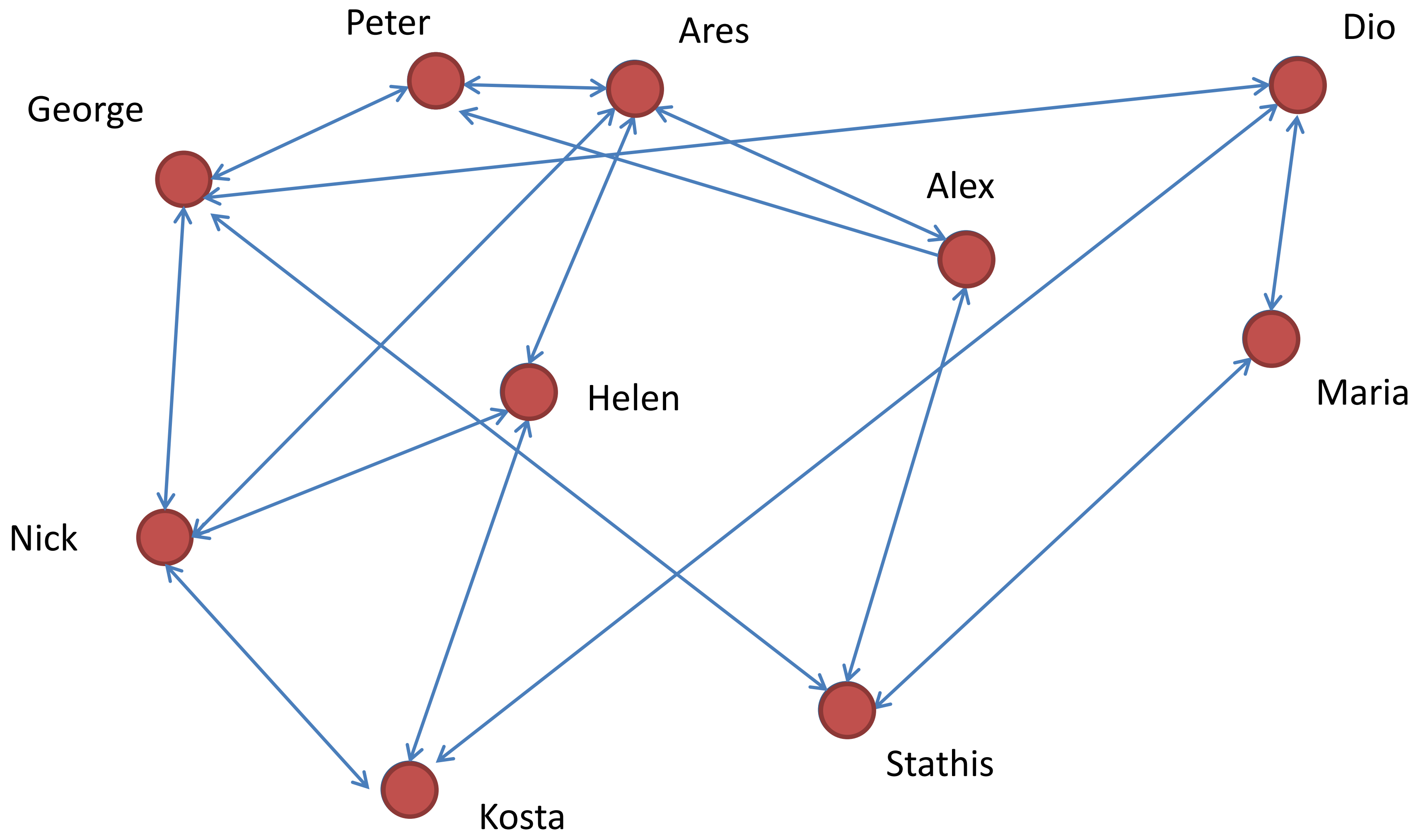
Who has what

- The network stores **collectively** who has how much money
- **Everyone** knows how rich Bob is
- **Everyone** knows how rich Alice is

- Therefore, Bob cannot send money he doesn’t have
- To **give** money, I have to have **received** it

Broadcasting

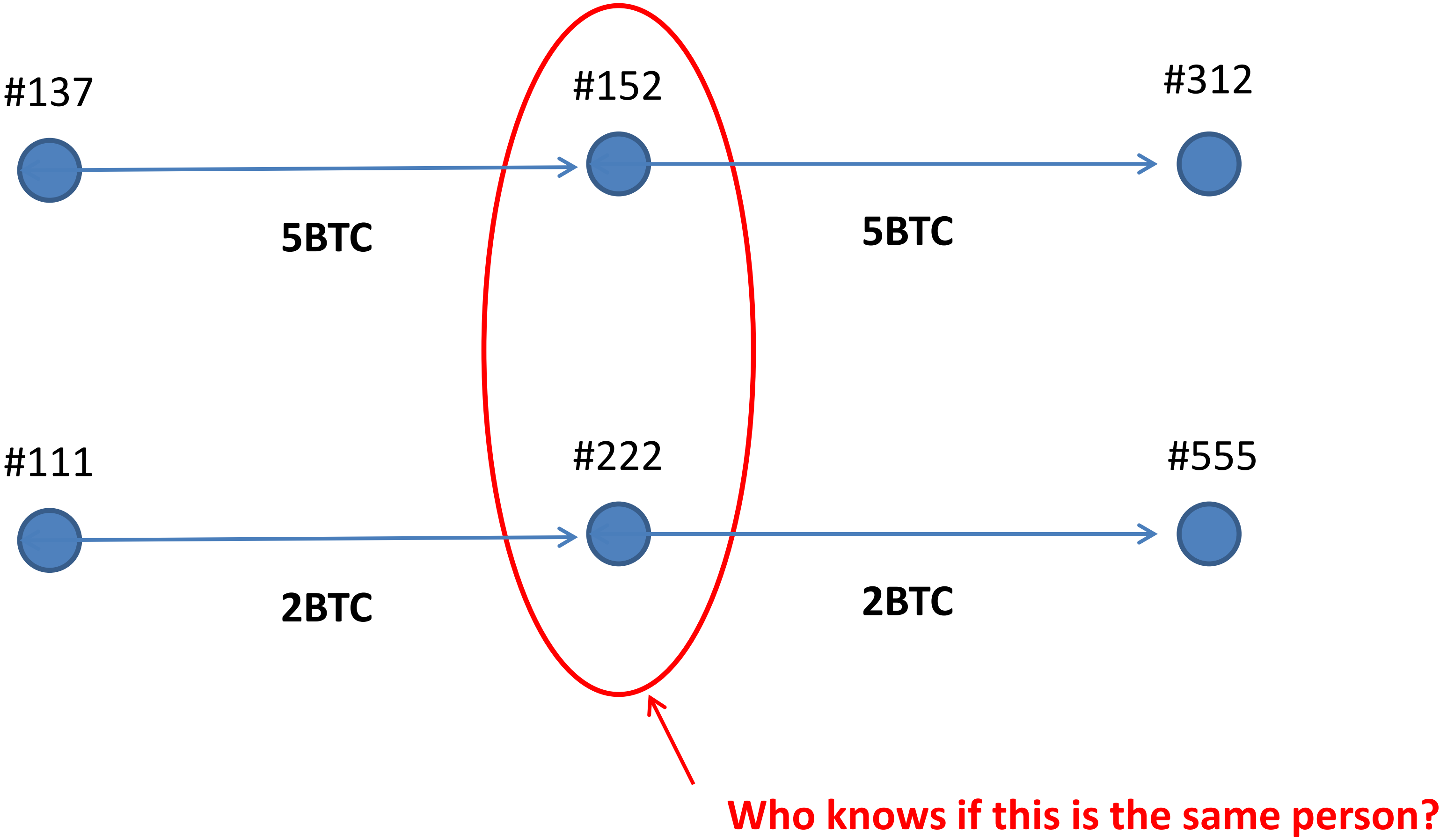
- Every transaction is **published** to the network
- Whenever I send or receive money, I communicate it to my neighbors


Peter
Ares
Dio
George
Alex
Helen
Maria
Nick
Stathis
Kosta

Anonymity

- For **every transaction** the participants use a **new private key**
- The nodes **don't have names** – only keys

Anonymity


#137
#152
#312
5BTC
5BTC
#111
#222
#555
2BTC
2BTC
Who knows if this is the same person?

Bob

Uses the key with which he **received** the money
PB, SB

m1 $\leftarrow$ “12BTC to PA”
h1 $\leftarrow$ H(m1)

s1 $\leftarrow$ sign_{SB}(h1) $\rightarrow$

Charlie

Genearates a **new** key
For this transaction
PC, SC

ver_{PA}(s2)

s2 $\leftarrow$ sign_{SA}(h2) $\rightarrow$

Alice

Generates a **new** key
For this transaction
PA, SA

ver_{PB}(s1)

m2 $\leftarrow$ “12BTC to PC”
h2 $\leftarrow$ H(m2)

Currency

- The measure according to which financial values are expressed or valuated.

- **A chain of digital signatures.**

Currency = Chain of digital signatures

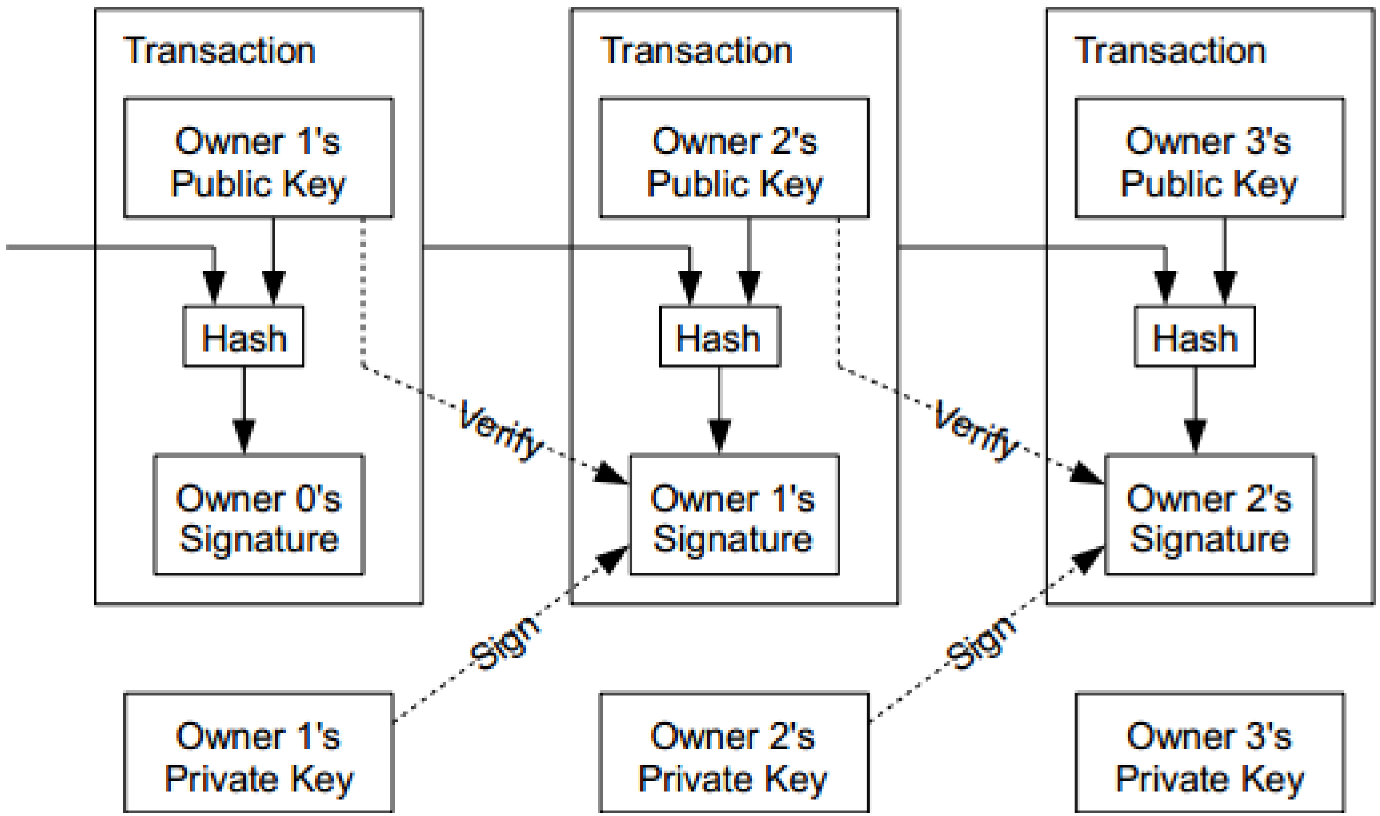
…

coin1 ← sign_{S0}(H(coin0 || P1))

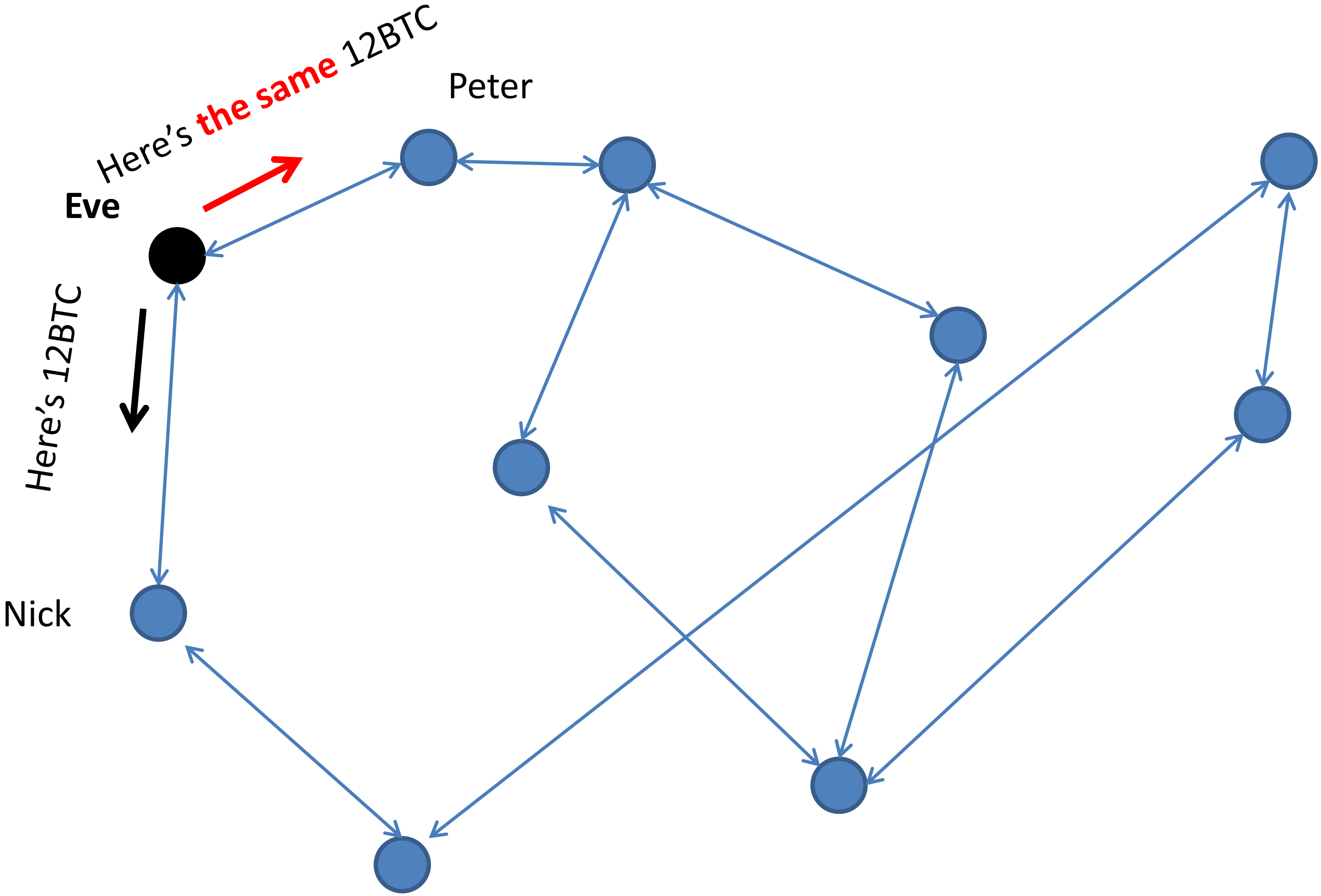
coin2 ← sign_{S1}(H(coin1 || P2))

coin3 ← sign_{S2}(H(coin2 || P3))

…


Transaction
Owner 1's Public Key
Hash
Owner 0's Signature
Verify
Sign
Owner 1's Private Key
Transaction
Owner 2's Public Key
Hash
Owner 1's Signature
Verify
Sign
Owner 2's Private Key
Transaction
Owner 3's Public Key
Hash
Owner 2's Signature
Owner 3's Private Key

Double spending



Double spending

- Undesired
- How can we avoid it?

Valid transactions

=

Transactions that have **not** been acted out >= **twice**?

This would mean I can cancel a transaction I don’t like!

Cancelling a transaction

- Bob pays 1BTC to Alice for a cup of coffee
- Alice delivers the cup of coffee to Bob
- Bob pays the same 1BTC to Charlie
- Charlie rejects the transfer
- The network considers both transactions invalid
- Alice loses her money
- Bob loses his money too – but he doesn’t care

We need a better way to prevent double spending!

The arrow of time

- **Valid** is the **first** transaction in the chain
- **Later** transactions are **invalid**

The arrow of time

- **When** did a transaction take place?
- I cannot trust a signature
- The date may be forged

Blocks

- Recent transactions are accumulated into a **block**
- Calculate **the hash** of each block
- Every new block includes the **hash** of its previous block
- Every block is published
- Every next block is in the **future** with respect to its previous block
  - Otherwise **it could not have known** its hash

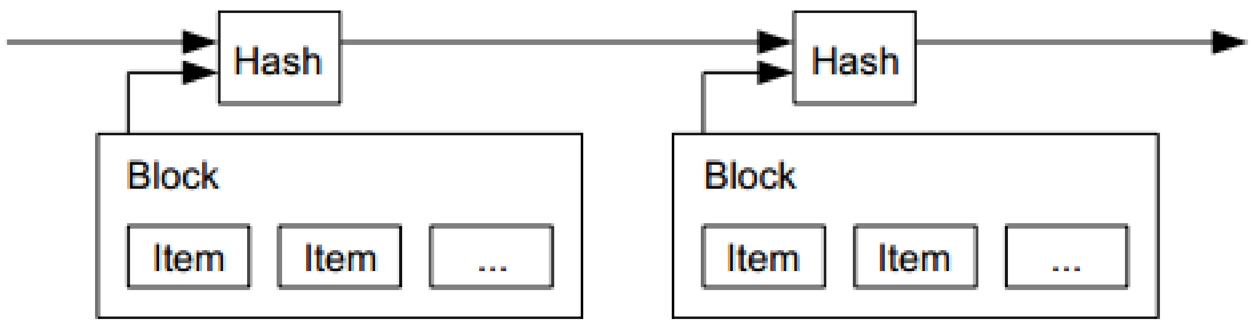
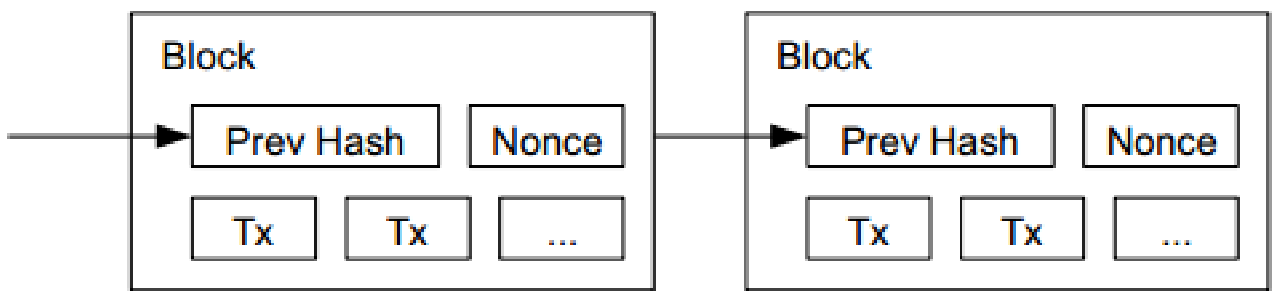


Image ©Satoshi Nakamoto

Proof of work

- We cannot just publish blocks
  - We'd need a trusted party
- Blocks are calculated at the node level and broadcasted
- We introduce an **artificial difficulty** to block generation
- It's **hard** to generate a block


Block
Prev Hash
Nonce
Tx
Tx
...
Block
Prev Hash
Nonce
Tx
Tx
...

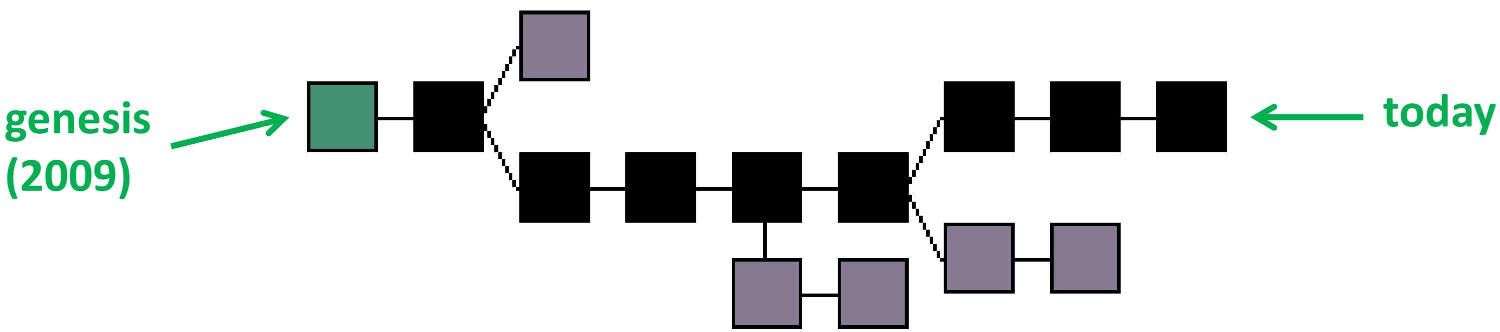
```
nonce ← 000000
while H( block || nonce ) ≠ “000000”:
    nonce ← nonce + 1

broadcast( block )
```

Proof of work

- Each block **validates** the transactions it includes
- A block chain is generated
- Every valid block inherits from genesis

Image ©theymos



Proof of work

- All nodes try to generate the block
- The first node to do so publishes
- The next block continues from there

Transaction validation

- A transaction is **validated** when included in the next block
- It becomes **exponentially difficult** to construct fraudulent blocks as time passes
- Every next block **secures** all previous blocks
- A transaction change incurs a change in all the next blocks

Transaction validation

- An adversary would need the majority of the network CPU to alter the chain
- Altering becomes **exponentially** harder as a transaction becomes validated by more and more blocks

Bitcoin mining

- Block generation = bitcoin earnings for the lucky CPU
- Controlled, mathematically predictable inflation

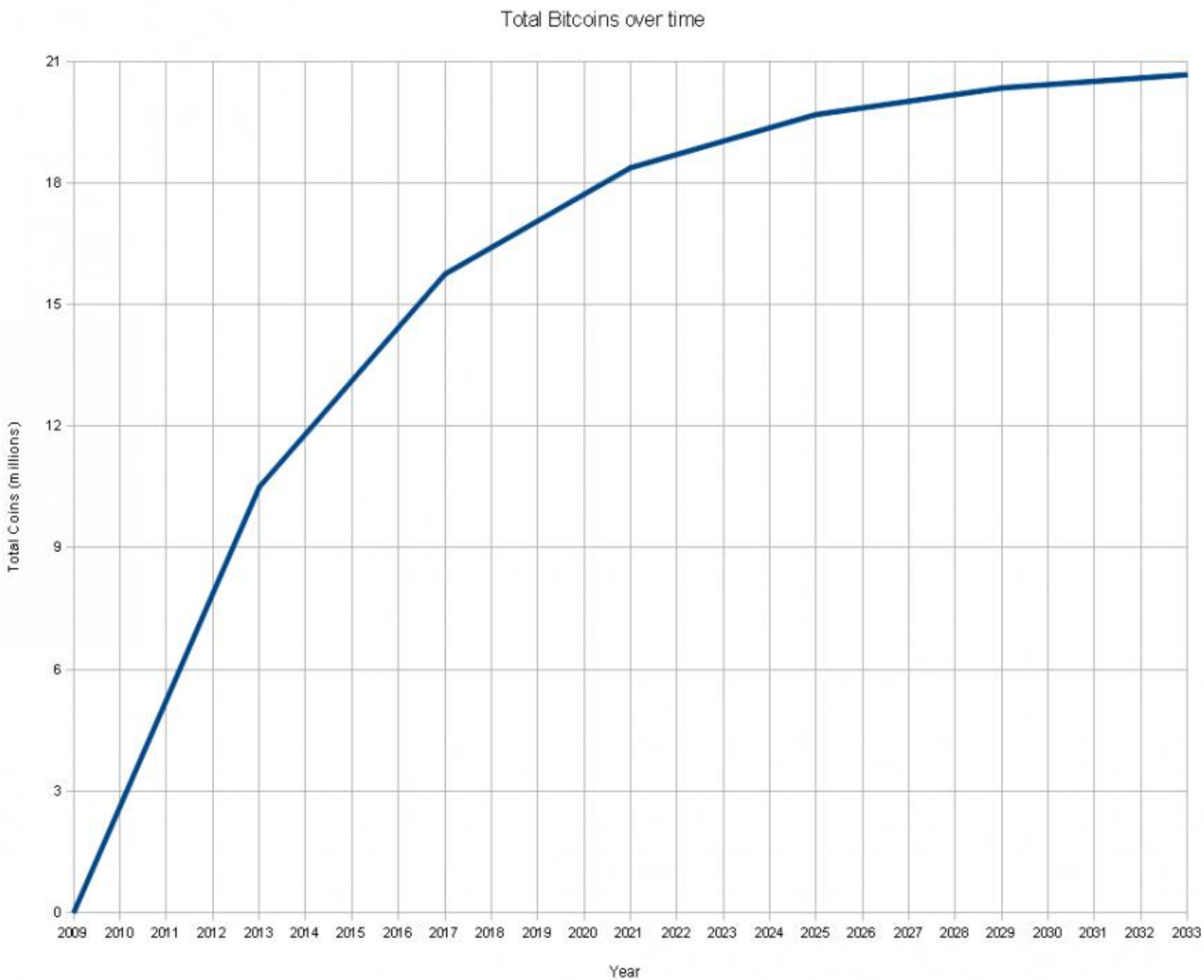


Image ©theymos

Technical details

- Digital signatures
  - Based on Elgamal (DSA)
  - Using elliptic curves
- Hash function
  - SHA256(SHA256(_))
- Work function
  - SHA256(_)

Bitcoin today

25 March 2012:

- 172,000 blocks
- 1BTC = 3.40€
- 8,642,700 BTC in circulation
- **~29,000,000€ in value**
- Network hashing frequency: > 10THz

Thank you! Questions?

bitcoin.org
Twitter: @dionyziz