



什么是区块链

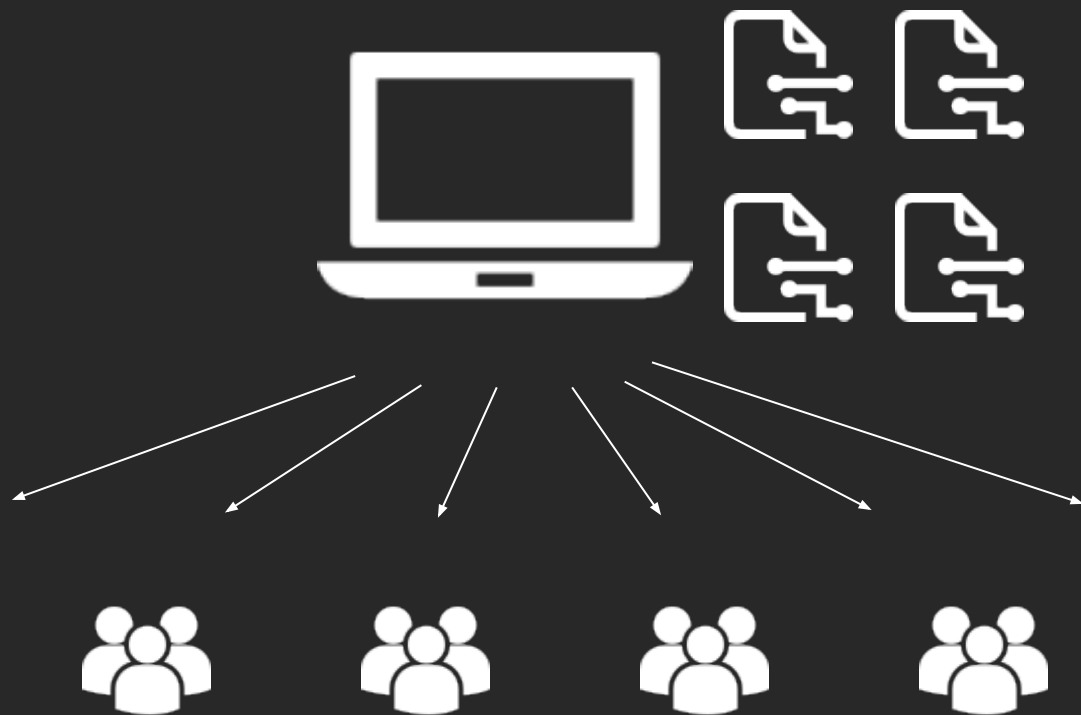
Maggie Dong

Developer Advocate, Parity Technologies Ltd.

maggiedong@parity.io

声明

本视频仅设计区块链的基本原理，和具体的某条区块链技术有所出入



所有的账本都存在
银行的中心化
服务器

传统的记账方式



每个人都可以
保存一份账本



随之而来的问题

谁可以往账本上写东西？

1. 如何证明你是你？
2. 如何防止恶意攻击？

如何确保所有人维护的账本都一样？

1. 为什么要维护同一份账本？
2. 如果有人的账本和其他人的不一样，会怎么办？

谁可以向账本里写入内容

任何人

账本

A 转给 B 30 元 (BTC)

B 转给 C 20 元 (BTC)

C 转给 D 10 元 (BTC)

A 转给 D 40 元 (BTC)

新问题:

如果B伪造A, 向账本里新加了一行:

A 转给 B 100 元

可以成功吗?



只有自己本人可以操纵自己的账户, 所以每当发起交易时, 必须向所有人证明你是你。

如何证明你是你

数字签名算法

协议

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

MAGGIE 

协议

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

abc

MAGGIE 

账本

A 转给 B 30 元 (BTC) *Alice*

B 转给 C 20 元 (BTC) *Bob*

C 转给 D 10 元 (BTC) *Charlie*

A 转给 D 40 元 (BTC) *Alice*

新问题:

如果B伪造A, 向账本里新加了一行:

A 转给 B 100 元

并且复制了A的签名, 可以成功吗?



数字签名的签名可以随着签名内容而变化

协议

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

MAGGIE123 

协议

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.

abc

MAG@#\$ 



私钥(sk): 01001111111111111111100000000.....10101011100

公钥(pk): 10100000011111111111100000000.....10101011100

签字

签名(Signature):

0101111000101010000.....11001001010001111010100001



Sign(secret_key, message) -> Signature

Verify(public_key, signature, message) -> true/false

账本

0 A 转给 B 30 元 (BTC) *01010*

0 B 转给 C 20 元 (BTC) *11000*

0 C 转给 D 10 元 (BTC) *10101*

1 A 转给 D 40 元 (BTC) *00111*

新问题:

B往账本里复制一条:

A 转给 B 30 元 *01010*

可以成功吗?



账本

0 A 转给 B 30 元 (BTC) *01010*

0 B 转给 C 20 元 (BTC) *11000*

0 C 转给 D 10 元 (BTC) *10101*

1 A 转给 D 40 元 (BTC) *00111*

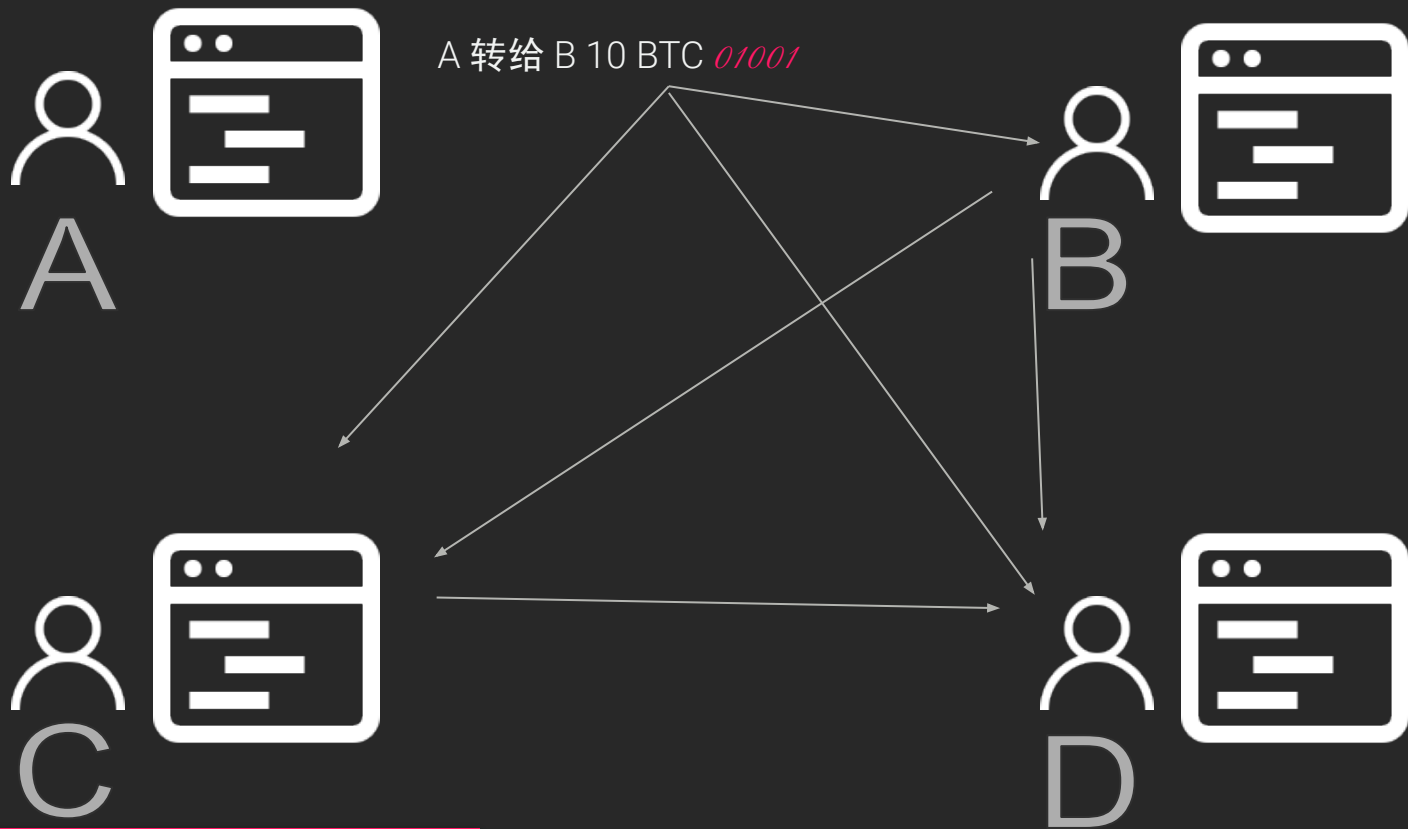
新问题:

伪造转账的记录暂时告一段落

如何保证所有人都维护同一份账本呢？

小结

1. 账本 = 交易历史
2. 每个人都可以向账本里添加内容
3. 用私钥签名，公钥向其他人证明。因此私钥绝对不可以泄露，但是公钥可以展示给别人
4. 每个人的交易里包含一个自增长数字(nonce)，用于防止别人复制攻击



每个人都可以保存账本

A 转给 B 10 BTC *Signature*

B 转给 F 50 BTC *Signature*

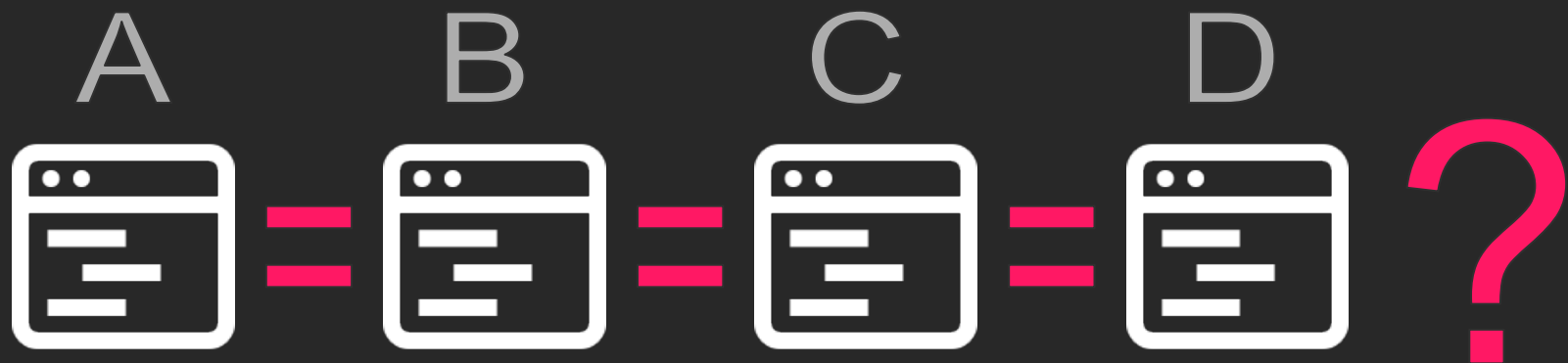


D 转给 E 12 BTC *Signature*

E 转给 A 50 BTC *Signature*

..... *Signature*

如何确保每个人的账本都一样呢？



如何确保每个人的账本都一样呢？

如何维护账本的一致性

共识

什么是Hash

Trapdoor function

256 bit

hash[消息/文件]



```
00101000101011010101111101010101100101000101011
010101111101010101000101000101011010101111101010
10100101000101011010101111101010101001010001010
11010101111101010101001010001010110101011111010
10101001010001010110101011111010101010010100010
10110101011111101010101
```

hash[“maggie”]



```
00101000101011010101111101010101100101000101011
01010111110101101000000011111111111000000000111
1111100101010100000000010101001010001010110101011
11101010101010110111111111111000000000011111111
10110010101001010110101011111010101010010100010
10110101011111101010101
```

hash[“Maggie”]



```
11111111111111111111111111111111110101000101011
01010111110101101000000011111111111000000000111
1111100101010100000000010101001010001010110101010
00000000000000000000000011111111111111111111111
11111010010101101010111110101010100101000100000
0000000000000000000000001
```

什么是hash

hash(
transactions +
nonce)

账本	
0 A 转给 B 30 元 (BTC)	01010
0 B 转给 C 20 元 (BTC)	11000
0 C 转给 D 10 元 (BTC)	10101
1 A 转给 D 40 元 (BTC)	00111
12346	

nonce

=

```
0010100010101101010
1111101010101100101
0001010110101011111
0101010100101000101
0110101011111010101
0100101000101011010
1011111010101010010
1000101011010101111
1010101010010100010
1011010101111101010
1010010100010101101
0101111101010101001
0100010101101010111
1101010101
```

什么是工作量证明PoW (Proof Of Work)

区块

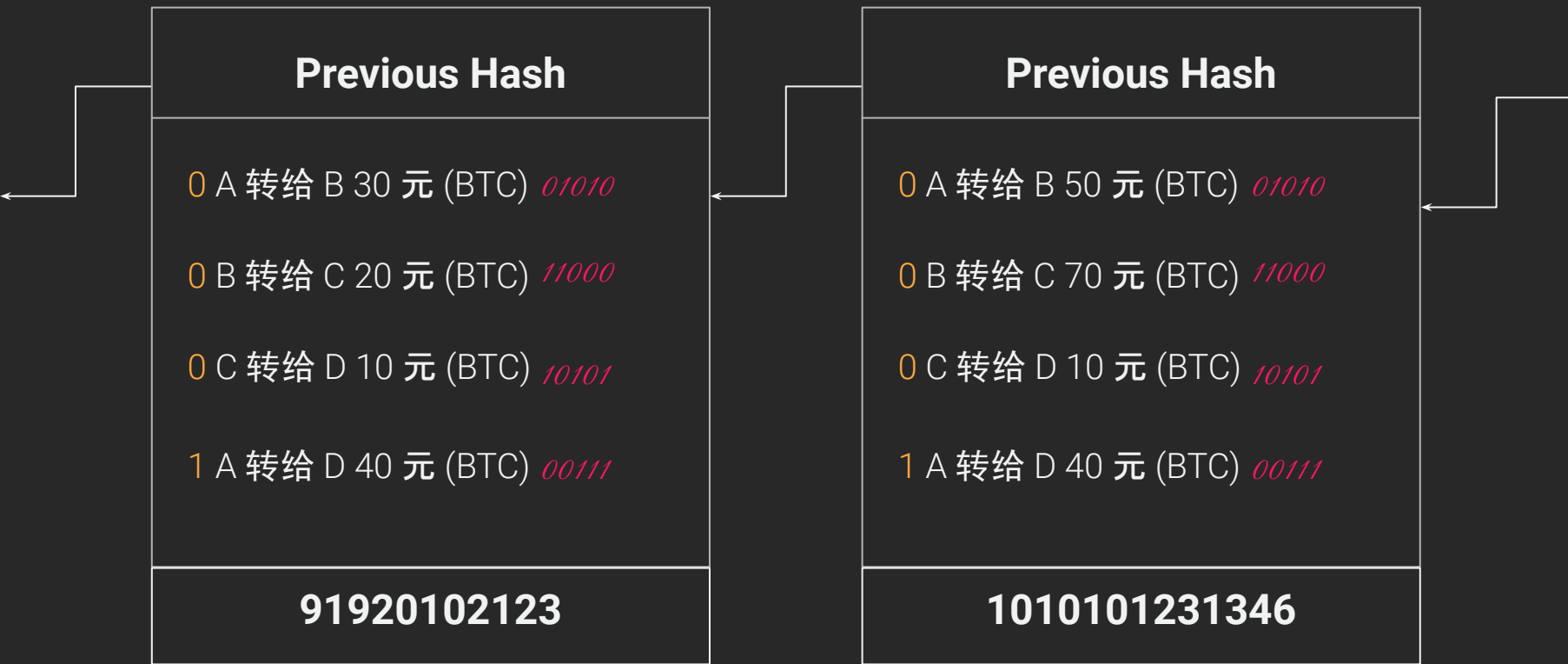
0 A 转给 B 30 元 (BTC) 01010

0 B 转给 C 20 元 (BTC) 11000

0 C 转给 D 10 元 (BTC) 10101

1 A 转给 D 40 元 (BTC) 00111

12346



区块#1

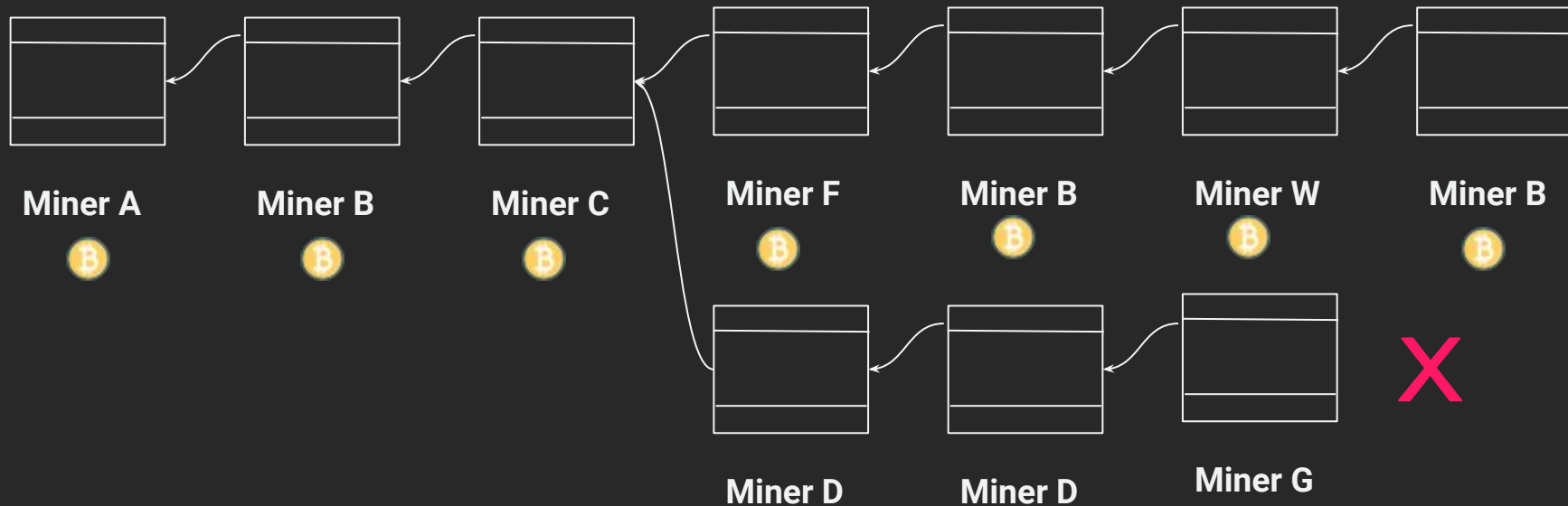
区块头

哈希(hash)
时间戳(timestamp)
高度(height)
矿工地址 (miner)
Merkle root
...

区块体

A 转给 B 30 元 (BTC) *01010*
B 转给 C 20 元 (BTC) *11000*
C 转给 D 10 元 (BTC) *10101*
A 转给 D 40 元 (BTC) *00111*

真实的区块长什么样子



为什么大家要维护一样的账本 —— 有奖励

~~梅事去中心威權幼~~

小结

1. 哈希函数: 将任何文件 / 文本变成一串256bit的01串, 难碰撞, 不可逆
2. 工作量证明 => 不停地算出符号要求的哈希 值
3. 产生分叉怎么办 => 等待最长链的出现

谢谢

区块链的记账方式