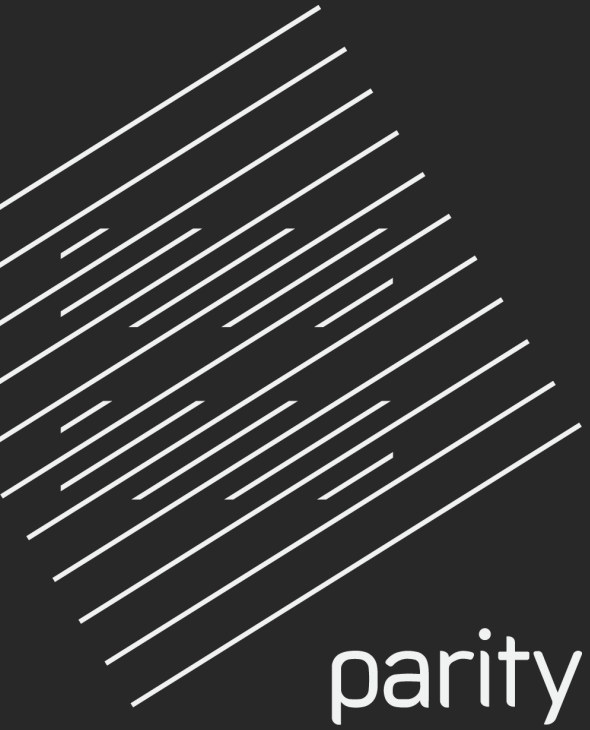
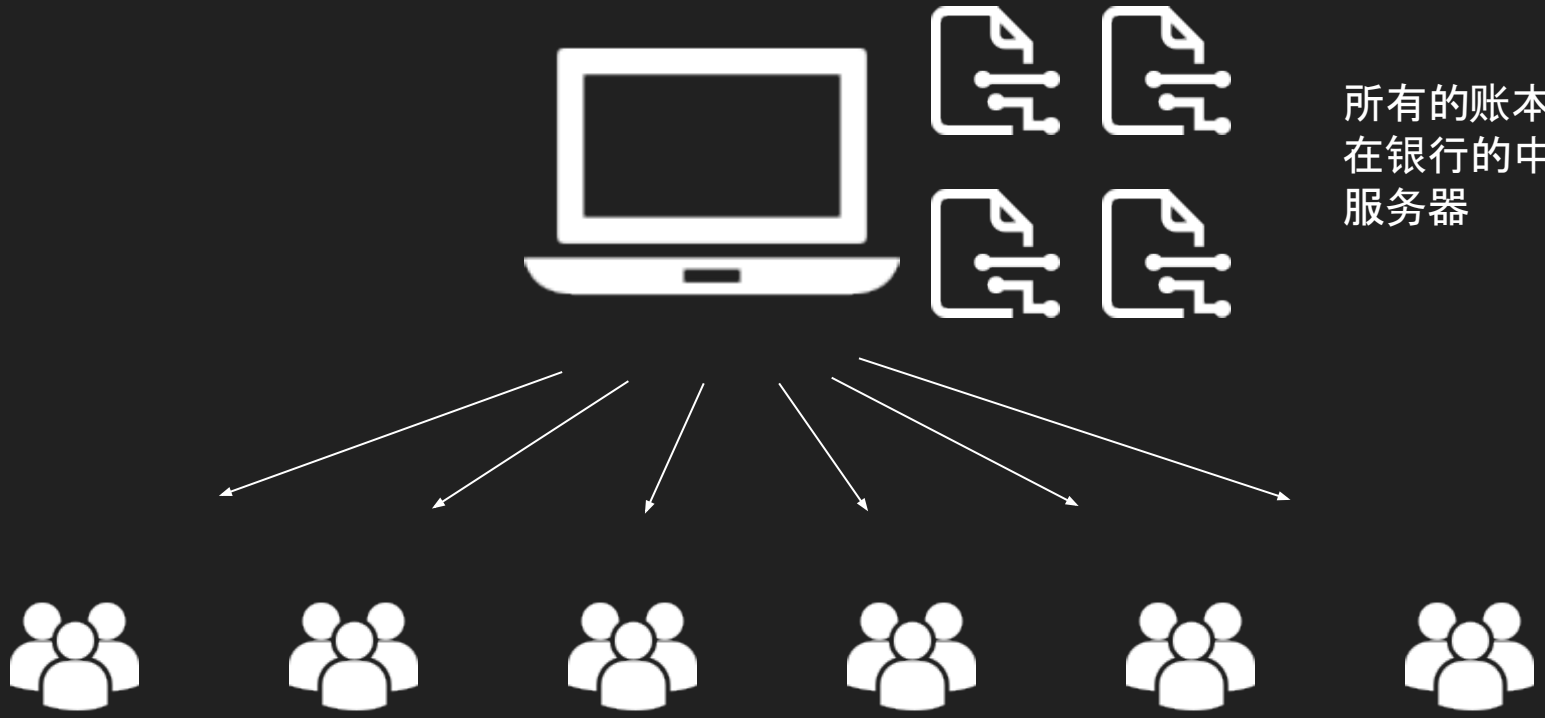


区块链技术与应用入门

什么是区块链

Suvi Dong
Developer Advocate, Parity Technologies Ltd.
maggiedong@parity.io



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

区块链的记账方式

账户	余额
Alice	100
Bob	20
Charlie	10

传统的账本

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，e.g. 双花？

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？

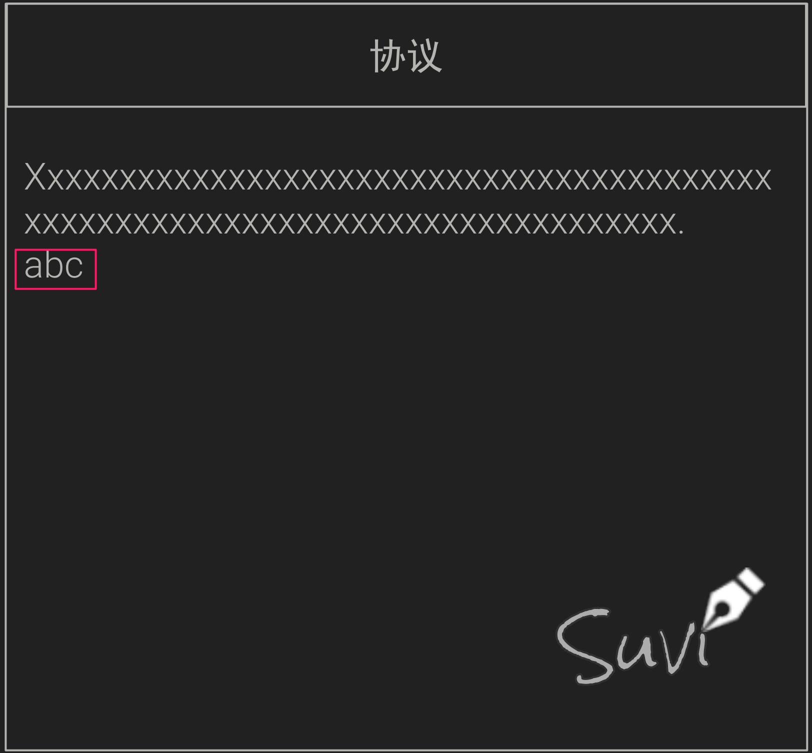
只有自己本人可以操纵自己的账户，所以每当发起交易时，必须向所有人证明你是你。

如何证明你是你

数字签名算法

parity


协议
Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
Suvi


协议
Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.
abc
Suvi

传统手写签名

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？

X

数字签名的签名可以随着签名内容而变化

协议

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

协议

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

abc

数字签名算法

私钥(sk): 0100111111111111111111000000000.....10101011100

公钥(pk): 1010000001111111111111000000000.....10101011100

签字

签名(Signature):
0101111000101010000.....1100100101000111101010001

数字签名算法

Sign(secret_key, message) -> Signature

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

并不会有比穷举法更好的攻击方法

2^{256}到底有多大？

账本

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*

新问题：

B往账本里复制一条：

A 转给 B 30 元 *01010*

可以成功吗？ X

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

万一有人欠钱怎么办呢？

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

账本

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*

新问题：

账本中的转账记录需要一个权威机构作为信用背书，如果到了清算日发现C并没有足够的余额清算，整个账本体系就崩溃了。

账本

A 得到了 100 元 (BTC)

B 得到了 100 元 (BTC)

C 得到了 100 元 (BTC)

D 得到了 100 元 (BTC)

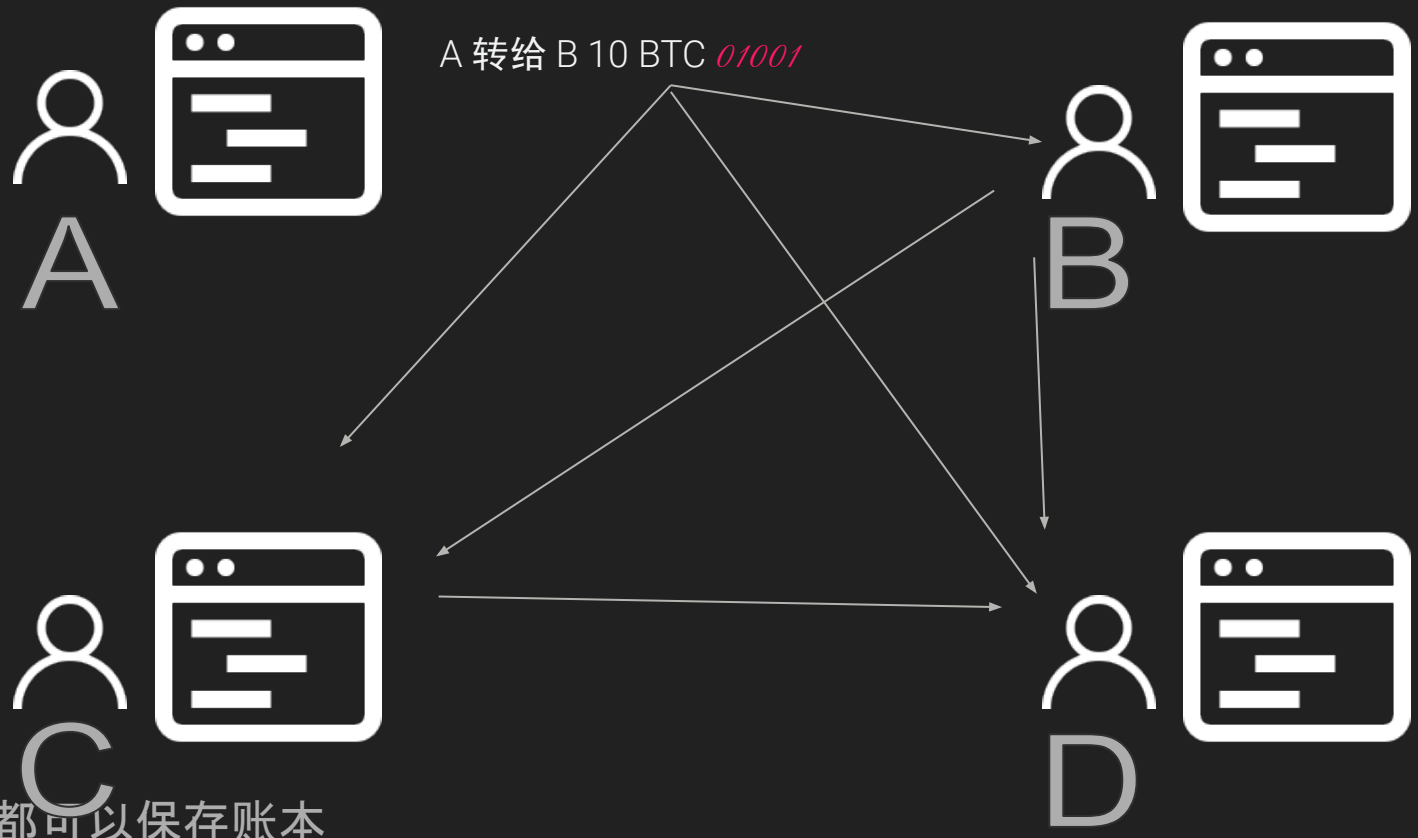
A 转给 B 30 元 (BTC) *01010*

B 转给 C 20 元 (BTC) *11000*

C 转给 D 10 元 (BTC) *10101*

A 转给 D 40 元 (BTC) *00111*

确保没有人会有欠账的问题



每个人都可以保存账本

A 转给 B 10 BTC Signature

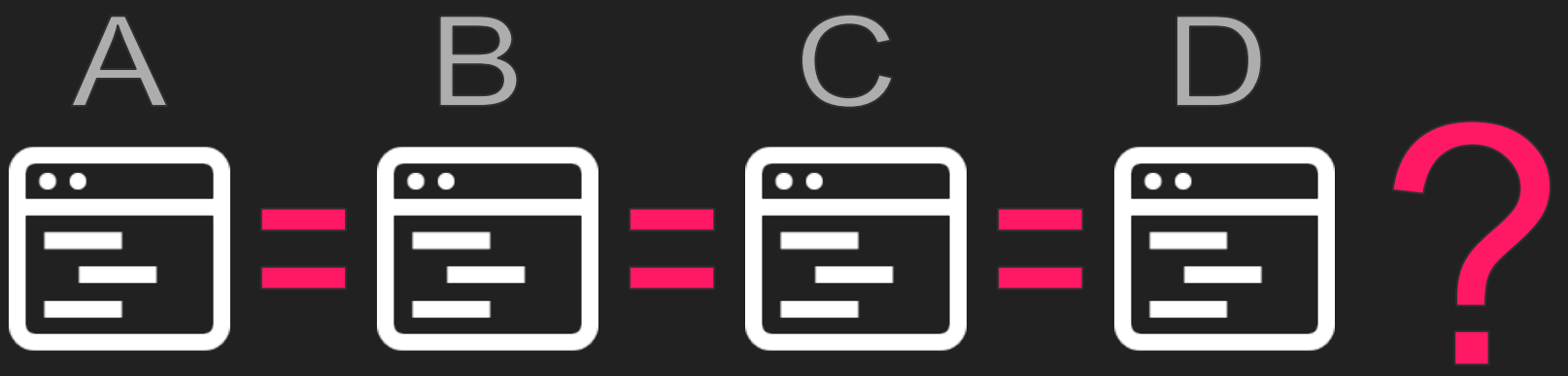
B 转给 F 50 BTC Signature

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
01010111110101010100101000101011010101111101010
10100101000101011010101111101010101001010001010
11010101111101010101001010001010110101011111010
10101001010001010110101011111010101010010100010
10110101011111010101

hash(“maggie”)

00101000101011010101111101010101100101000101011
01010111110101101000000011111111111000000000111
11111001010100000000010101001010001010110101011
11101010101010110111111111111000000000011111111
10110010101001010110101011111010101010010100010
10110101011111010101

hash(“Maggie”)

什么是hash

11111111111111111111111111111111101010001010111
01010111110101101000000011111111111000000000111
11111001010100000000010101001010001010110101010
00000000000000000000001111111111111111111111111
11111010010101101010111110101010100101000100000
00000000000000000001

hash -?-> Proof of Work

什么是工作量证明PoW (Proof Of Work)

60个0

0000000000000000000000000000000000000000000000000000000
00000000000000000000000000000011111111111100000000011
11111100101010000000001010100101000101011010101
111101010101010110111111111111000000000001111111
110110010101001010110101011111010101010010100001
0101101010101111101010101

hash(transaction
s + **nonce**)

1/2/3/4/.....9587291...

特点：难计算，易证明

什么是工作量证明


区块
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

什么是区块

区块1

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*

1234689272

00000000000000000000000000000000
00000000000000000000000000000000
00000000000000111111111111000
00000011111111001010100000000
01010100101000101011010101111
10101010101011011111111111100
00000000111111111011001010100

区块2

2 A 转给 C 40 元 (BTC) *01010*

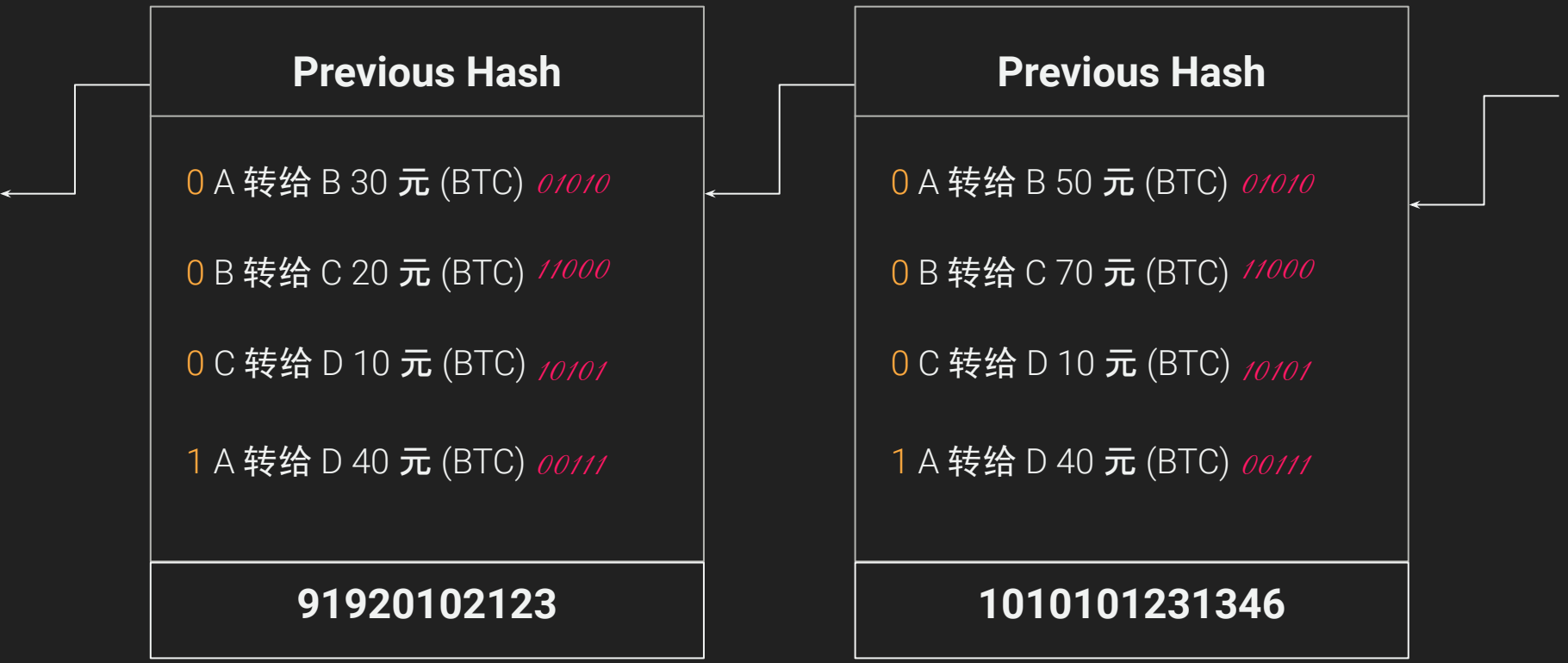
1 B 转给 C 20 元 (BTC) *11000*

1 C 转给 D 10 元 (BTC) *10101*

3 A 转给 E 40 元 (BTC) *00111*

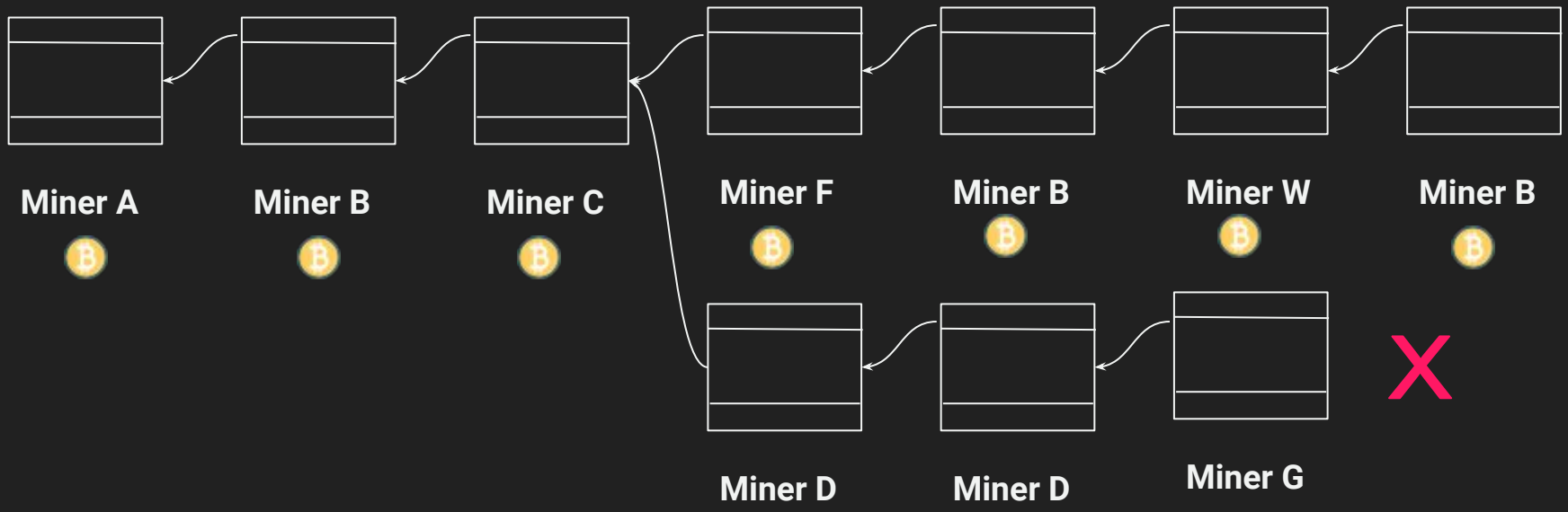
878218276129

00000000000000000000000000000000
00000000000000000000000000000000
000000000000000001111111111
11000000000111111110010101000
00000010101001010001010110101
01111101010101010110111111111
11100001111101010101111110001


Previous Hash
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
91920102123
Previous Hash
0 A 转给 B 50 元 (BTC) 01010
0 B 转给 C 70 元 (BTC) 11000
0 C 转给 D 10 元 (BTC) 10101
1 A 转给 D 40 元 (BTC) 00111
1010101231346

什么是区块链


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



大家如果选择维护那一条链？

xia

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

how safe is 256 bit

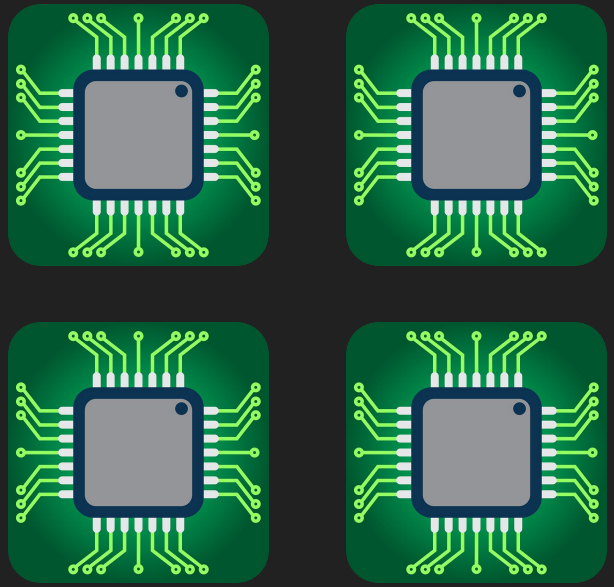
00000000000000000000000000000000000000000000000000000000000000001111111111100000000011
1111110010101000000000101010010100010101101010
1111010101010101101111111111110000000000111111
1101100101010010101101010111110101010100101000
1
0101101010111101010101

2^{256}

2^{256}

2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}
40亿	40亿	40亿	40亿	40亿	40亿	40亿	40亿

2^{32}



一个性能非常好的GPU大致计算10亿次/秒

一台可以每秒可以计算40亿次的电脑

2^{256}

2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}
40亿	40亿	40亿	40亿	40亿	40亿	40亿	40亿

40亿次/s

2^{32}

≈几百万台电脑

Kilo Google

2^{256}

2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}	2^{32}
40亿	40亿	40亿	40亿	40亿	40亿	40亿	40亿
	KG+	KG+ 	 		126.8年	5070亿年≈宇宙年龄37倍	1/40亿
40亿次/s				亿万星系超级计算机			

Any Questions?

- 区块链的现状
- 区块链目前有哪些赛道
- 区块链的从业者在做什么
- 区块链有哪些应用
- 人工智能 vs 区块链
- 择业选择
- BTC之后区块链海油发展吗
- 区块链什么编程语言最受欢迎
- 区块链团队一般都是什么样的 多大规模的
- ...