Information Security Engineering

2017.12.13

Bitcoin II

北京大学 软件与微电子学院
School of Software and Microelectronics, Peking University

Huiping Sun(孙惠平)

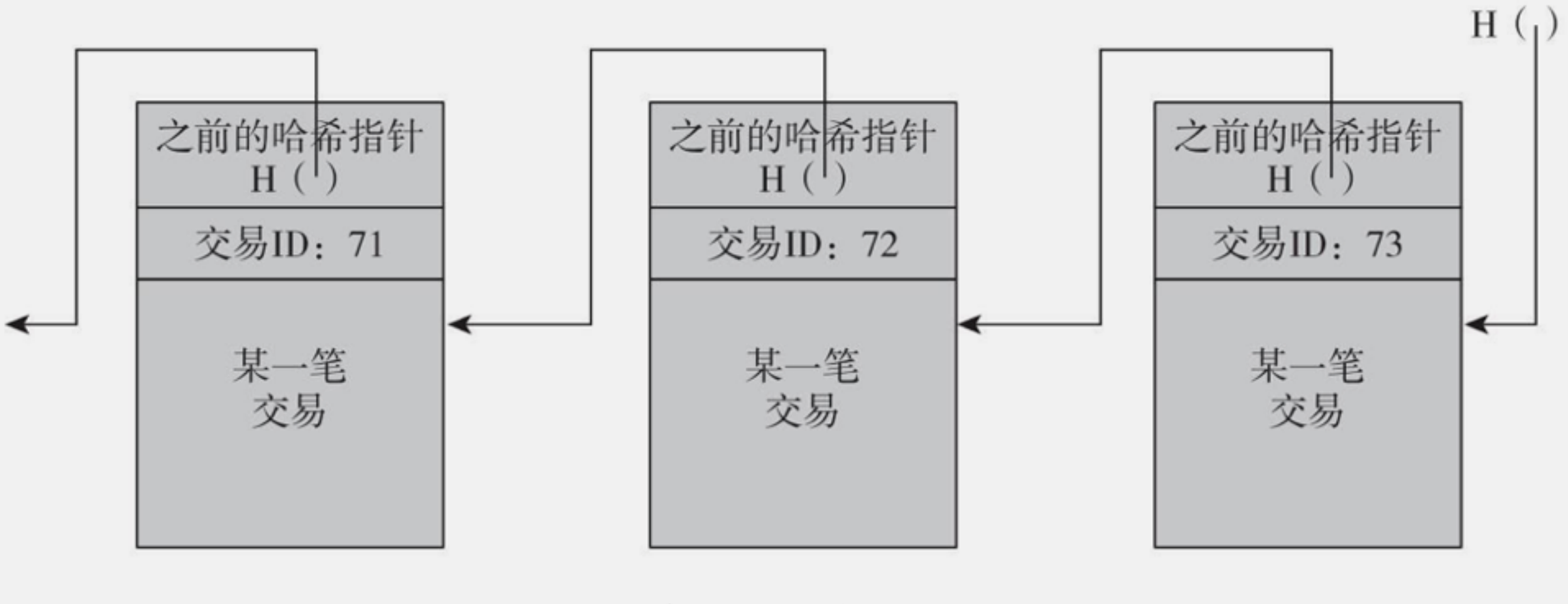
sunhp@ss.pku.edu.cn

主要内容

- 密码学和加密货币
- **比特币如何去中心化**
- **比特币的机制**
- **如何存储和使用比特币**
- 比特币挖矿
- 比特币和匿名性
- 社区、政治和监管
- 其余挖矿难题
- 比特币作为平台
- 其余代币和加密货币生态系统
- 比特币未来

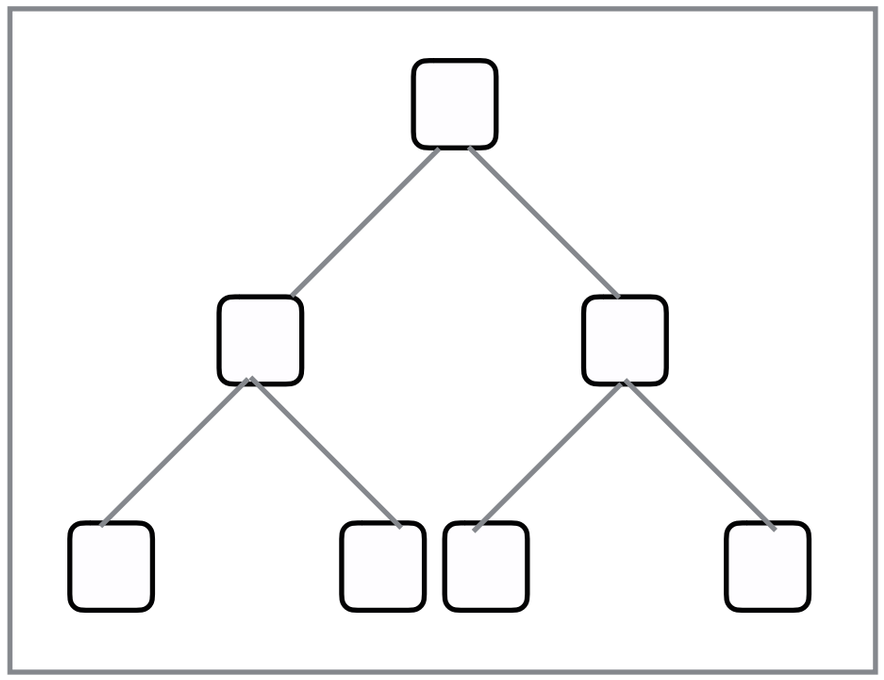
比特币
如何去中心

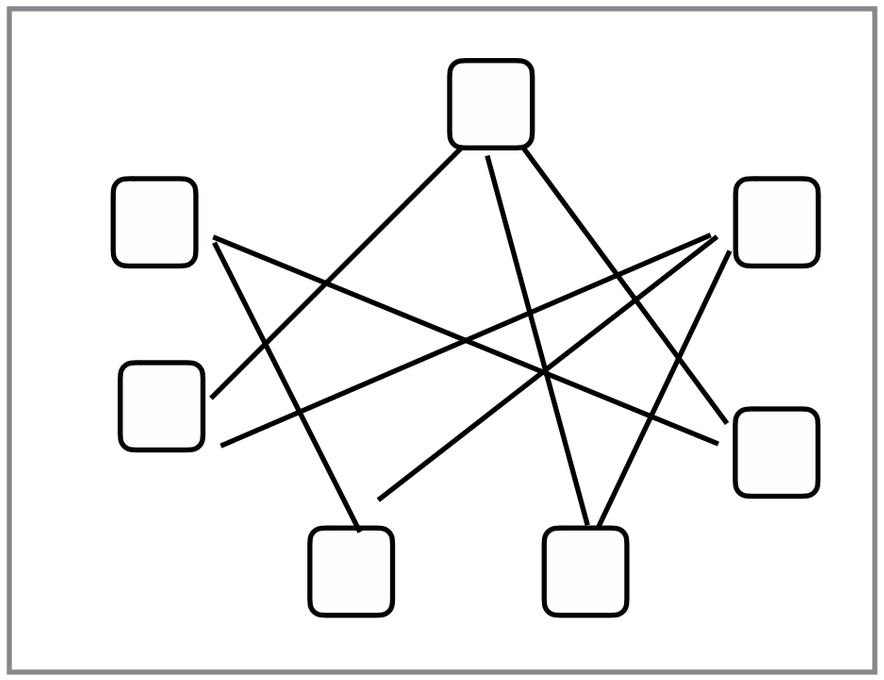
贪心币是中心化的



- 比特币如何要去中心?

中心化 vs. 去中心





神话

- 没有纯粹的中心化系统或者分布式系统
- 各有优缺点 *Internet, Email, IM,SNS*
- 大多数系统都是混合类型的 比特币?

比特币如何实现去中心

- **谁维护交易账本?**
- **谁有权限验证交易的有效性?**
- **谁创造新的比特币?** 技术

- 谁决定系统如何改变规则? 激励
- 比特币如何获得交易价格

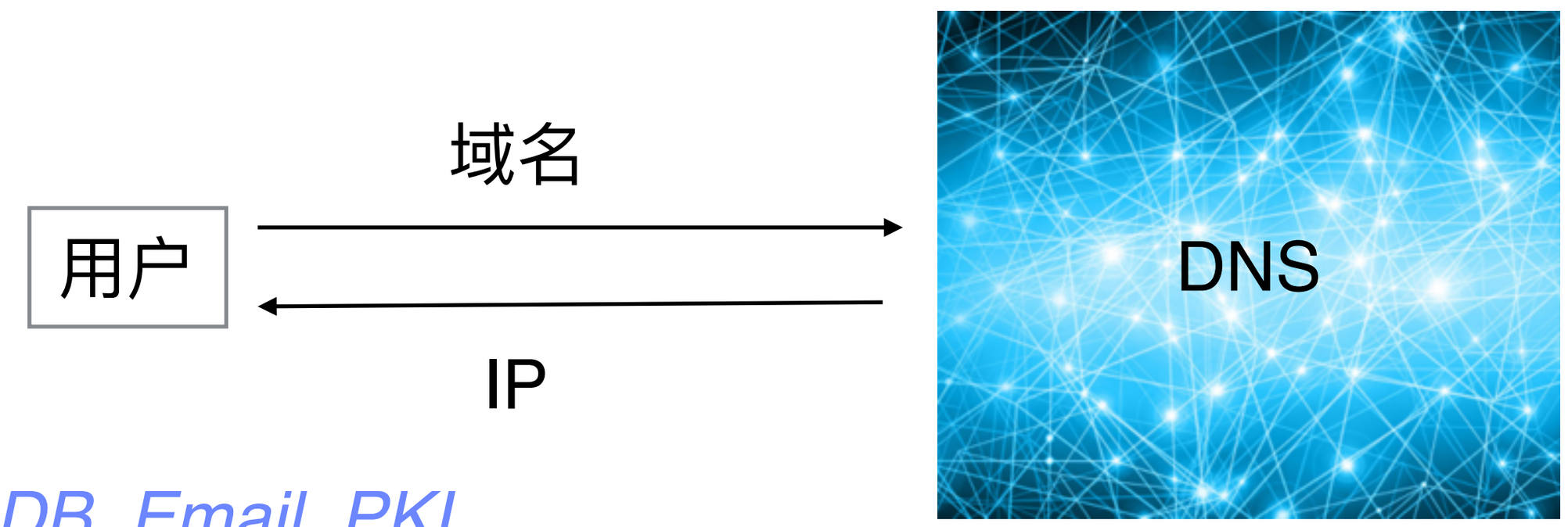
用户:对等网络 / 矿工 挖矿 / 开发人员:软件更新

分布共识

- 在一个有*n*个节点的系统中，每一个节点都有一个输入值，其中有一些节点是错误的或者恶意的。一个分布式共识协议具有如下两个属性：

  ✳ 结束时所有诚实的节点均认同该值；

  ✳ 该值由诚实节点产生



Distributed DB, Email, PKI

比特币的分布共识

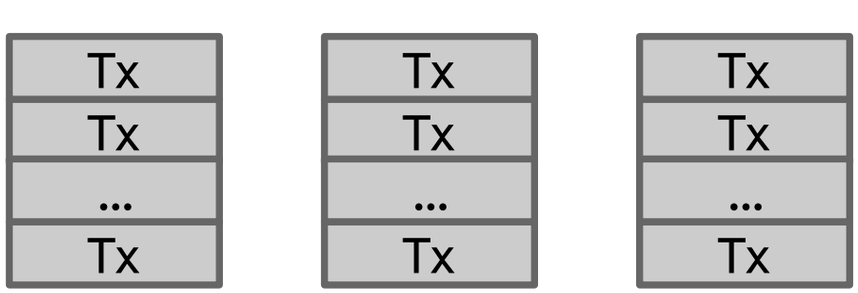
signed by Alice

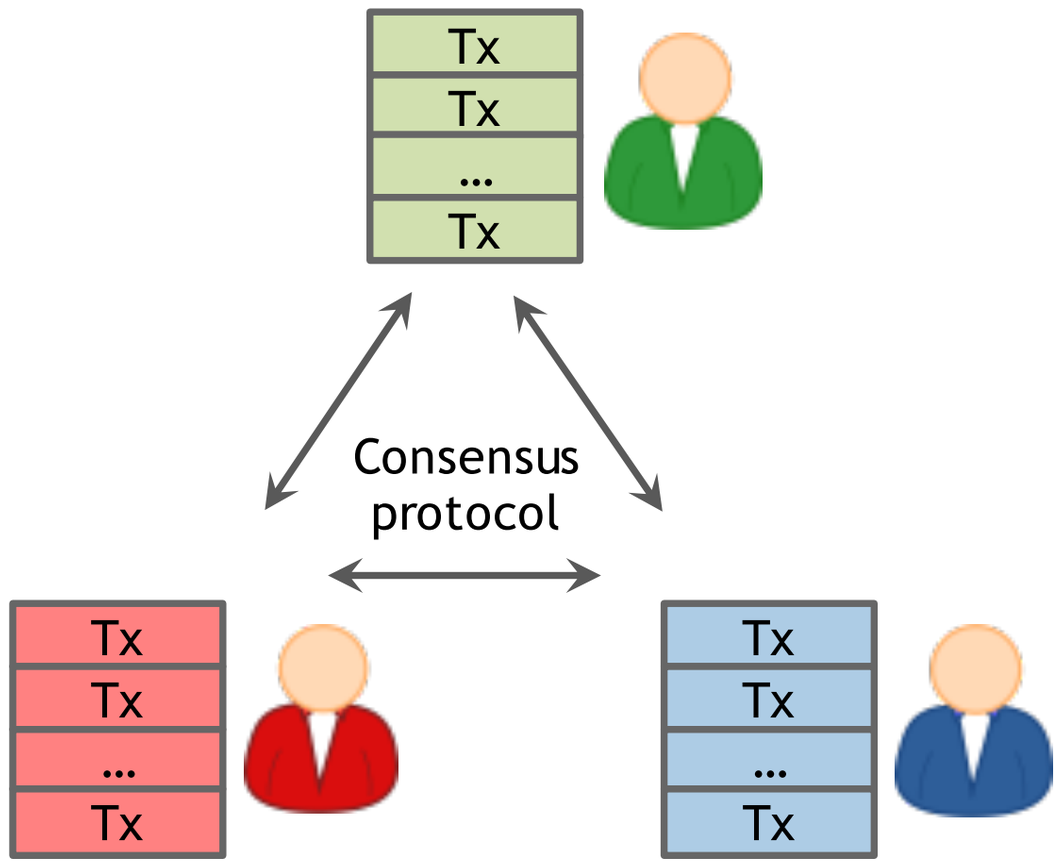
Pay to pk_{Bob} : H( )

- **比特币是一个*P2P*网络**
- ***Alice* 需要广播她完成的交易給所有的节点**
- ***Bob*计算机当时可以不在*P2P*网络中**
- *A single, global ledger for the system*
- 等待共识的业务、已共识的业务

比特币的分布共识

- 每一个节点输出它的未共识的业务竞争下一个*Block*

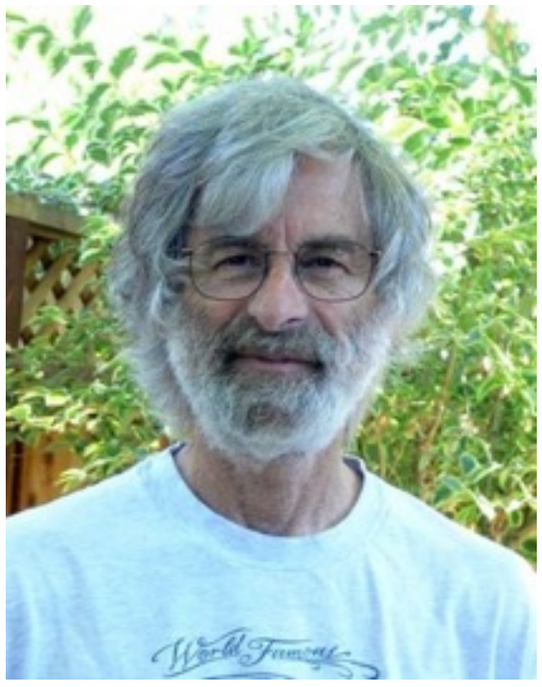




- 共识是难的
  - ➡ *Node: crash, malicious*
  - ➡ *Network: Imperfect (online, latency)*

Global Time

拜占庭将军问题和Paxos



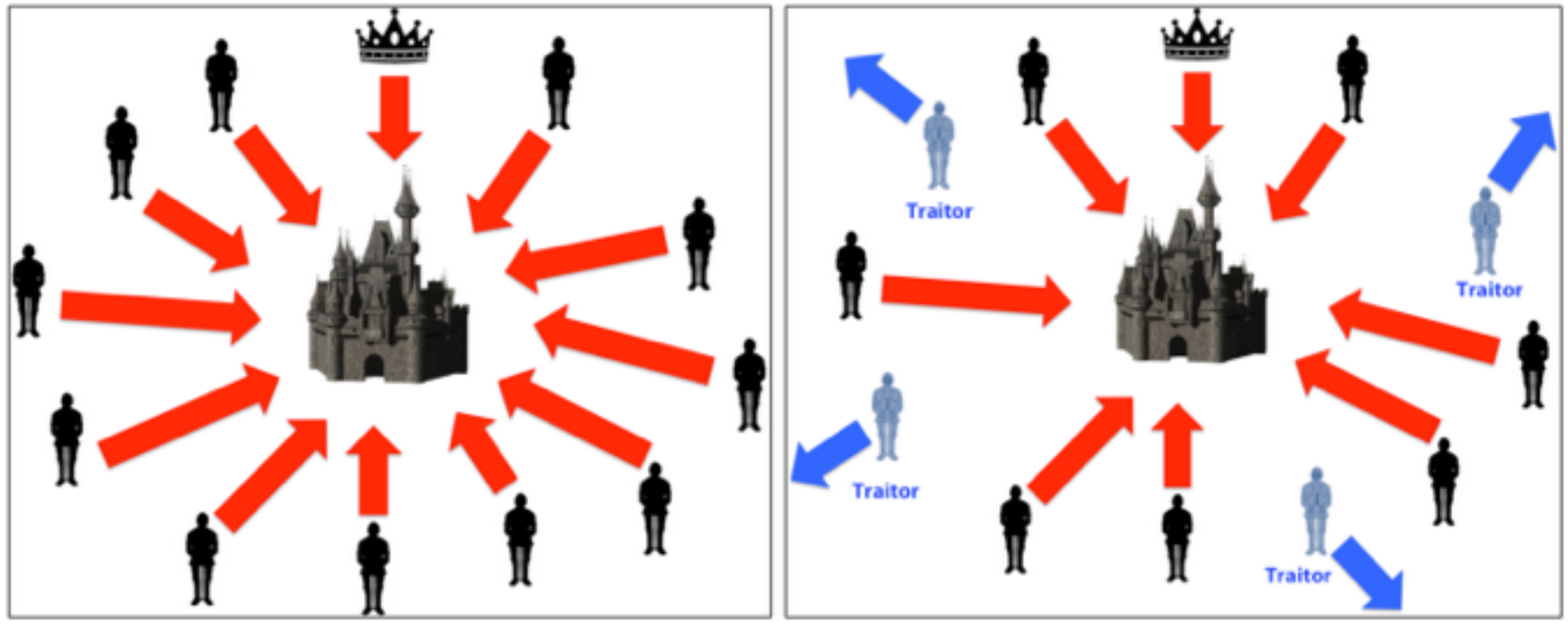
LESLIE LAMPORT

2013图灵奖

The Byzantine Generals Problem

1982

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE
SRI International



Coordinated Attack Leading to Victory　　Uncoordinated Attack Leading to Defeat

Paxos Made Simple

Abstract

2001

The Paxos algorithm, when presented in plain English, is very simple.

比特币的共识

- 理论落后于实践
- 引入了*Incentive*
  - ✳ 是电子货币
- 利用了随机性
  - ✳ 很长时间后才取得共识，*1*小时
  - ✳ 随着时间的增加，对某一块的共识的概率越来越大

比特币和身份

- 比特币节点需要身份 *(ID)*
- 比特币假设恶意节点小于*50%*
- 但是*P2P*系统中，*ID*面临很大问题
  - ✴ *Sybil Attack*
- *Pseudonymity*是比特币的目的

- 比特币跟踪和验证*ID*是困难的
- 比特币采用的应对方法：随机的选择节点

比特币的共识机制

- 新的交易被广播到所有节点
- 每个节点将新的交易放进一个区块
- 在每一轮中，一个随机的节点被选择可以广播它的区块
- 其余节点可以选择接受这个区块，前提是区块的教育是可验证的
- 节点将以上区块的*Hash*放进自己的区块，表示它认可这个新区块

隐形共识：接受该块并扩展 *vs.* 拒绝该块，扩展前面的块

恶意节点

- 窃取比特币
- 拒绝服务攻击
- 双重支付攻击

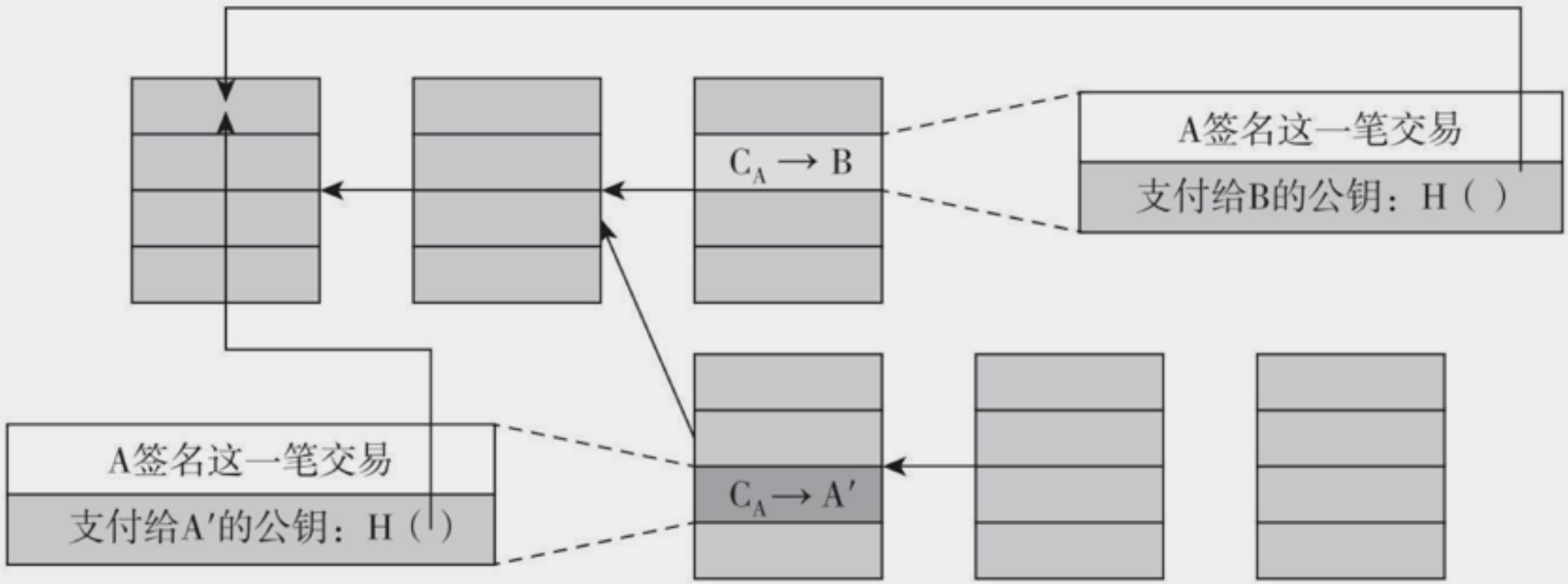


图2.2 双重支付攻击

注：爱丽丝创建了两笔交易：一笔是她付给鲍勃比特币的交易，另一笔是她将这笔比特币重复支付到她控制的另一个地址。因为这两笔交易用相同的比特币支付，所以只有一笔会被放进区块链。图中的箭头表示一个区块链接到前一个区块的指针，通过在前一个区块自己的内容中包含了一个哈希值进行了扩展。C_A代表爱丽丝拥有的币。

双重攻击防止: 等待多次确认

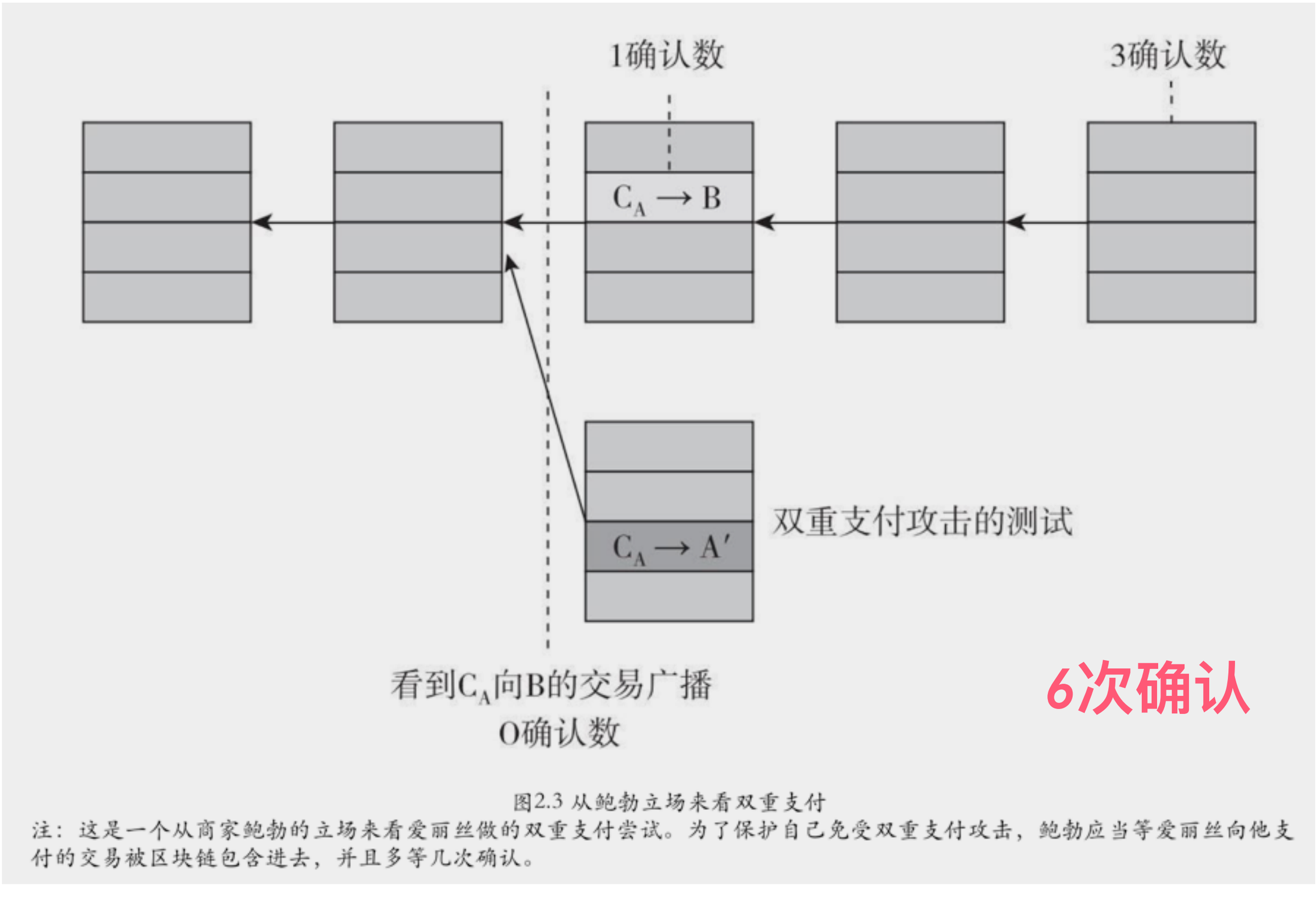
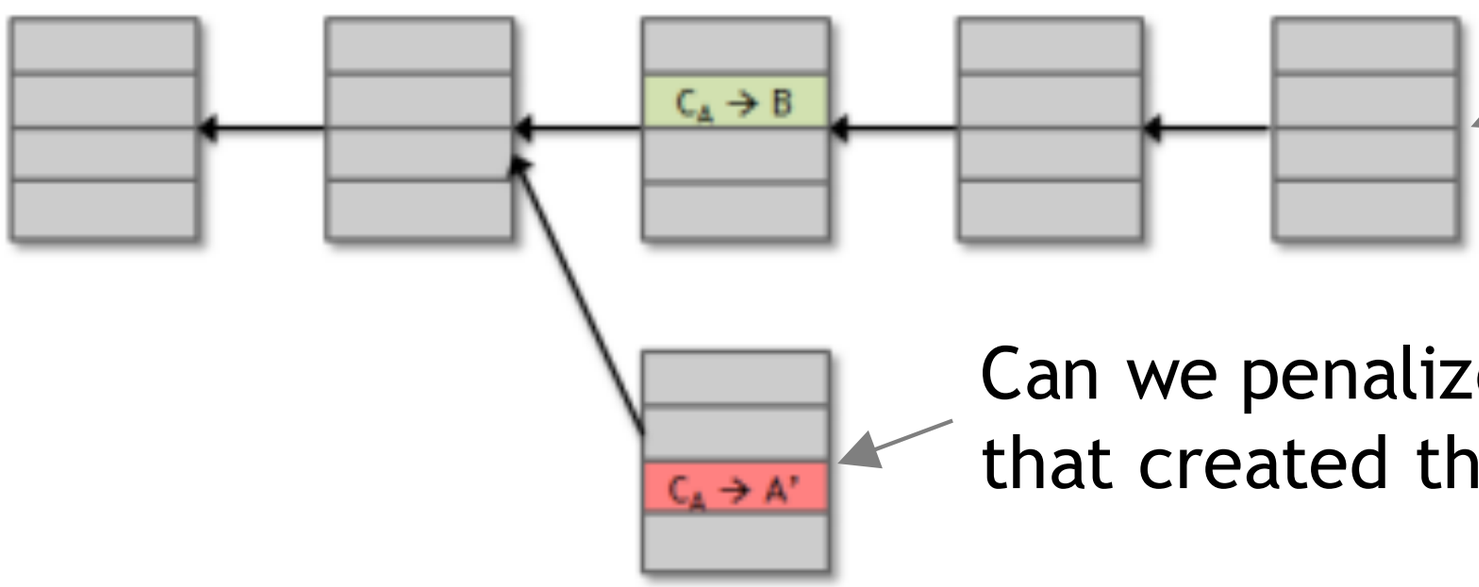


图2.3 从鲍勃立场来看双重支付

注：这是一个从商家鲍勃的立场来看爱丽丝做的双重支付尝试。为了保护自己免受双重支付攻击，鲍勃应当等爱丽丝向他支付的交易被区块链包含进去，并且多等几次确认。

激励节点诚实



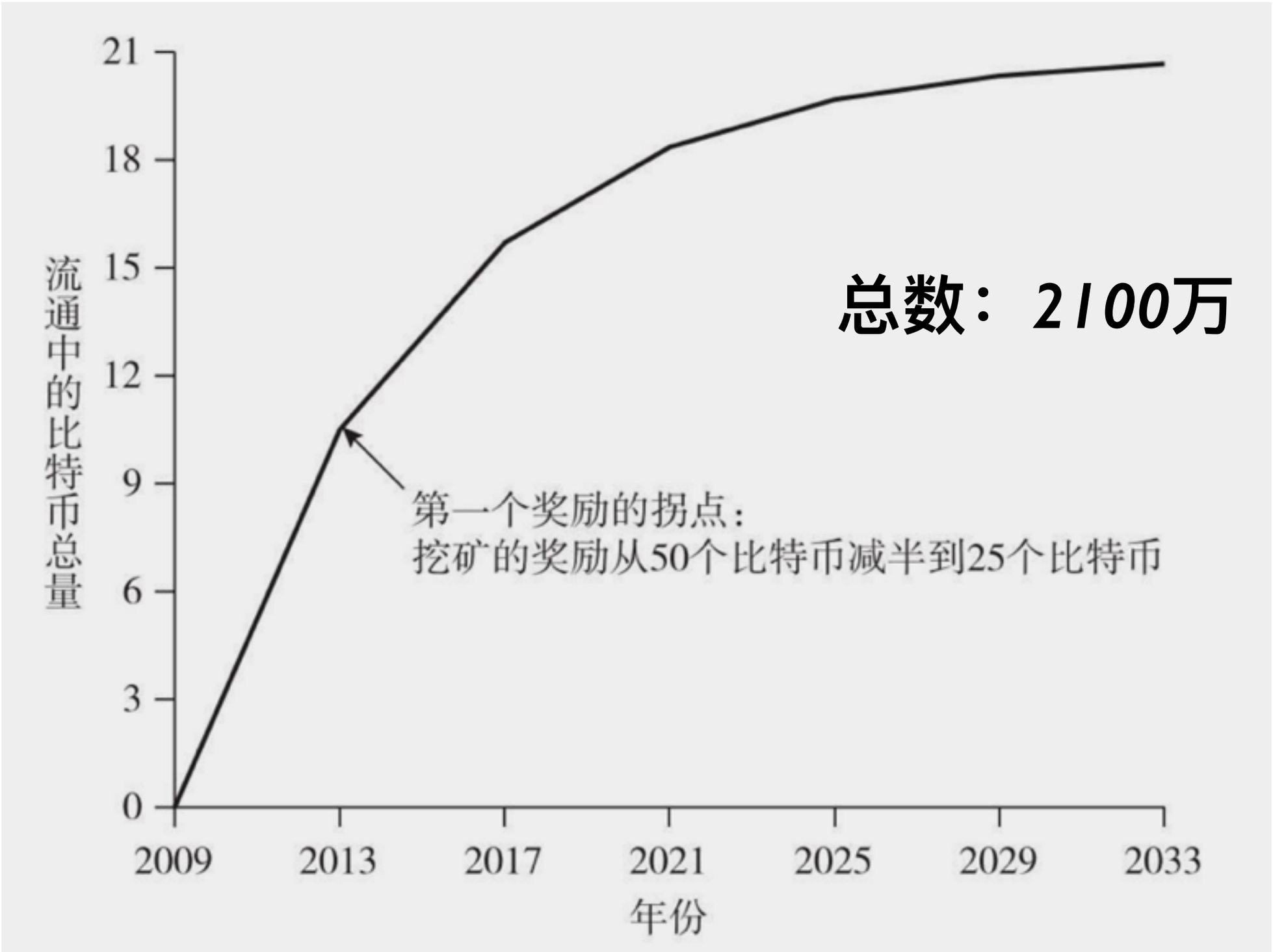
Can we reward nodes that created these blocks?

Can we penalize the node that created this block?

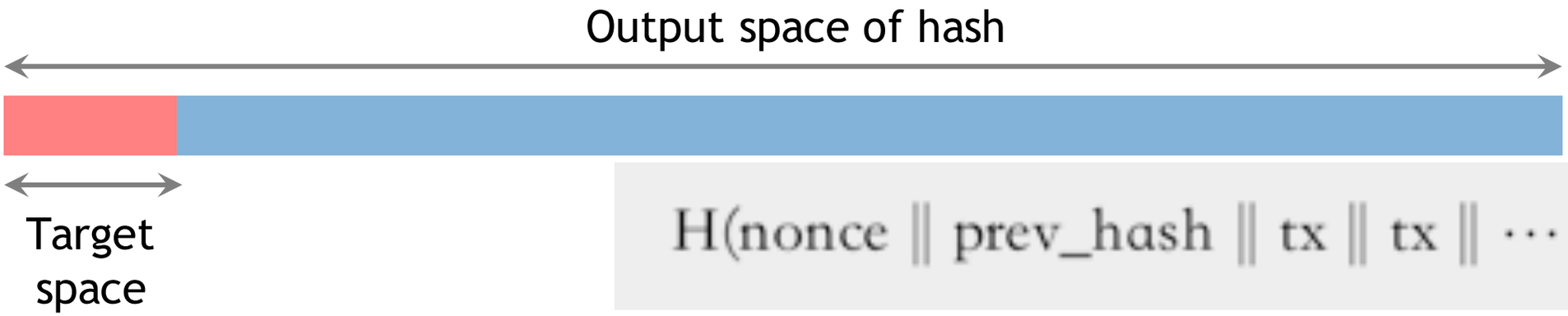
区块奖励 *vs.* 交易费奖励

交易费：输入和输出不等

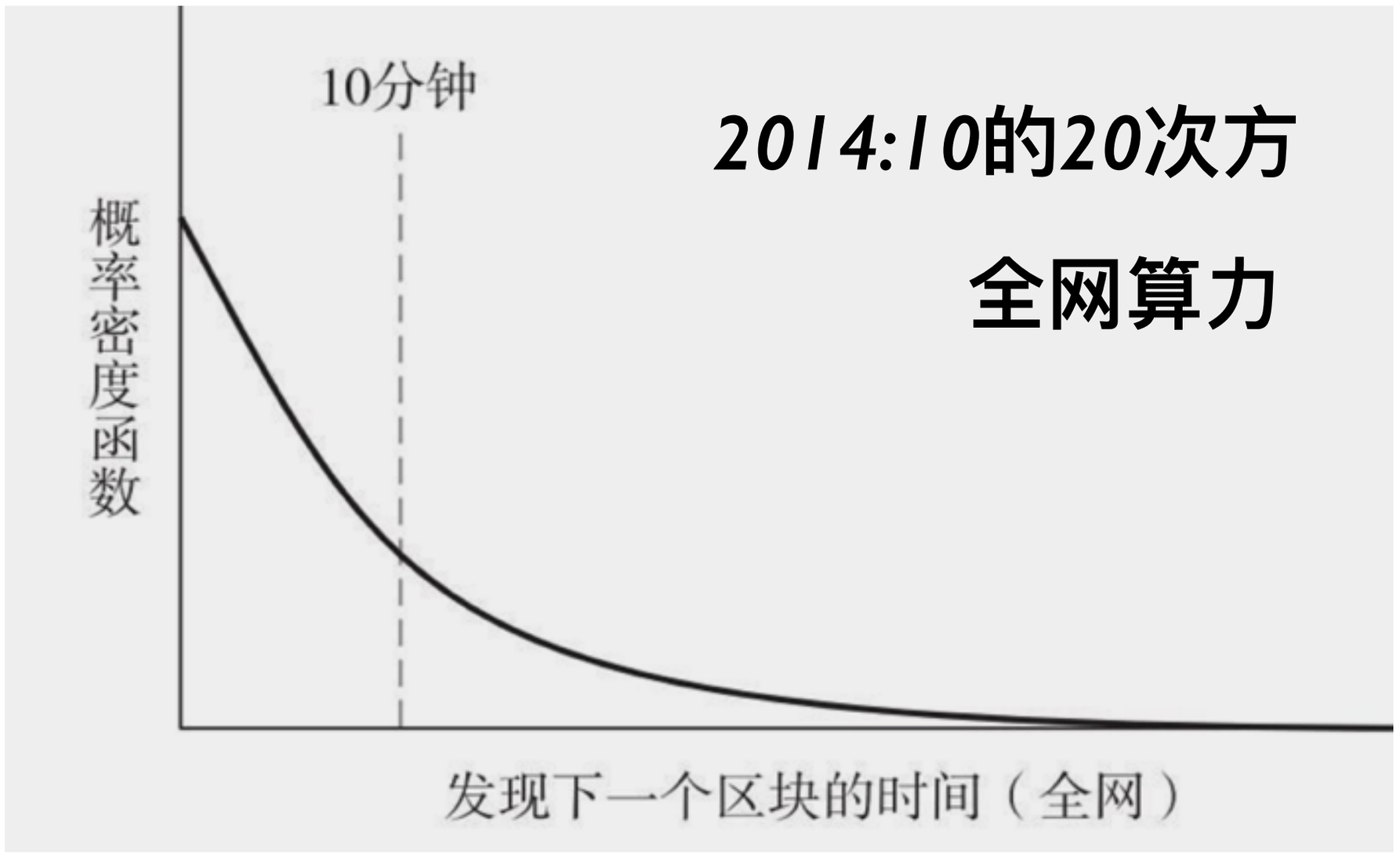
比特币奖励


流通中的比特币总量
21
18
15
12
9
6
3
0
2009
2013
2017
2021
2025
2029
2033
年份
总数：2100万
第一个奖励的拐点：
挖矿的奖励从50个比特币减半到25个比特币

挖矿



$$H(nonce \parallel prev\_hash \parallel tx \parallel tx \parallel \cdots \parallel tx) < target$$



*2014:10*的*20*次方

全网算力

限定*Hash*的输出范围

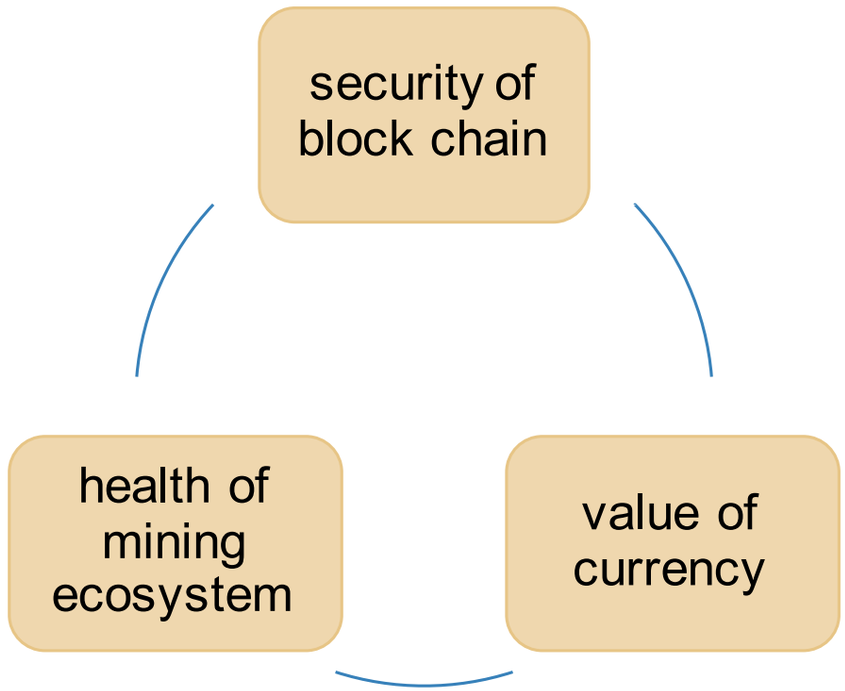
临时随机数

***PoW*：工作量证明**

***PoS*：权益证明**

总结

- 身份
- 交易
- *P2P*网络
- 区块链
- 共识
- *Hash*难题
- 挖矿
- 经济



- *51%*攻击

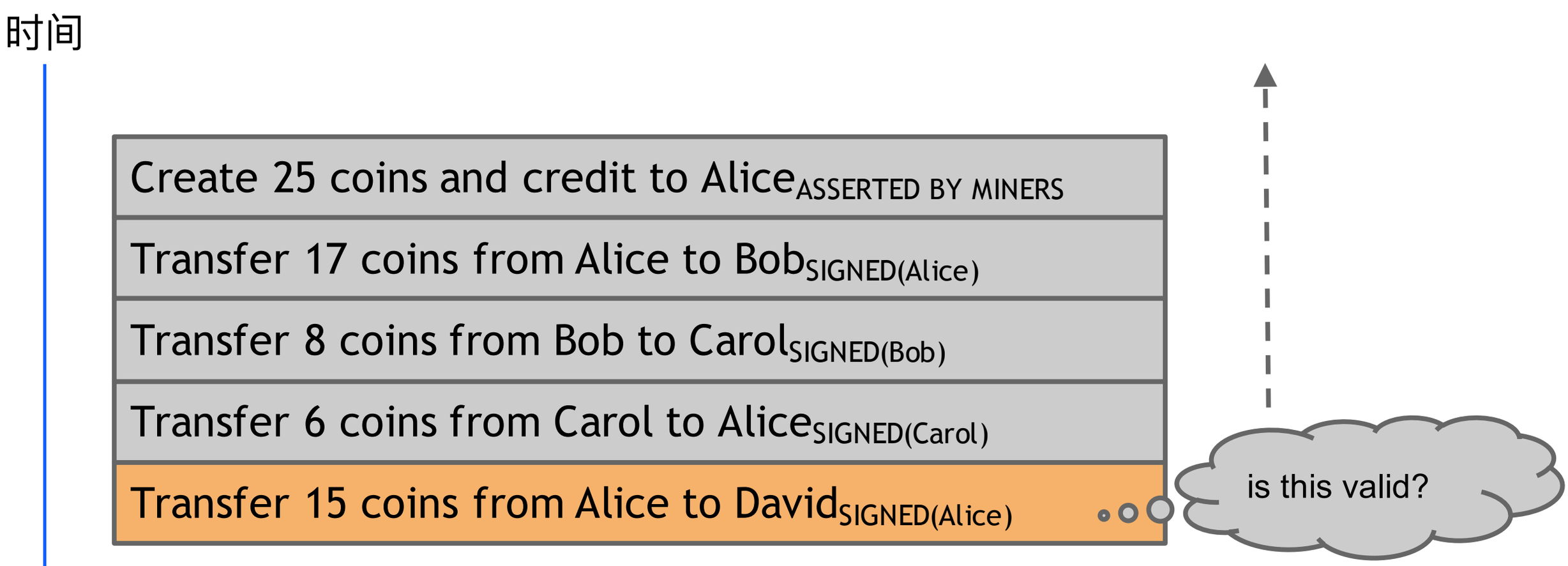
➡ 窃取币、操纵交易、改变激励、破坏信心

思考

- 比特币现在的情况
- 比特币是分布式电子货币吗?
- 比特币实现匿名了吗?
- 比特币安全吗?
- 比特币不能操控吗?

比特币机制

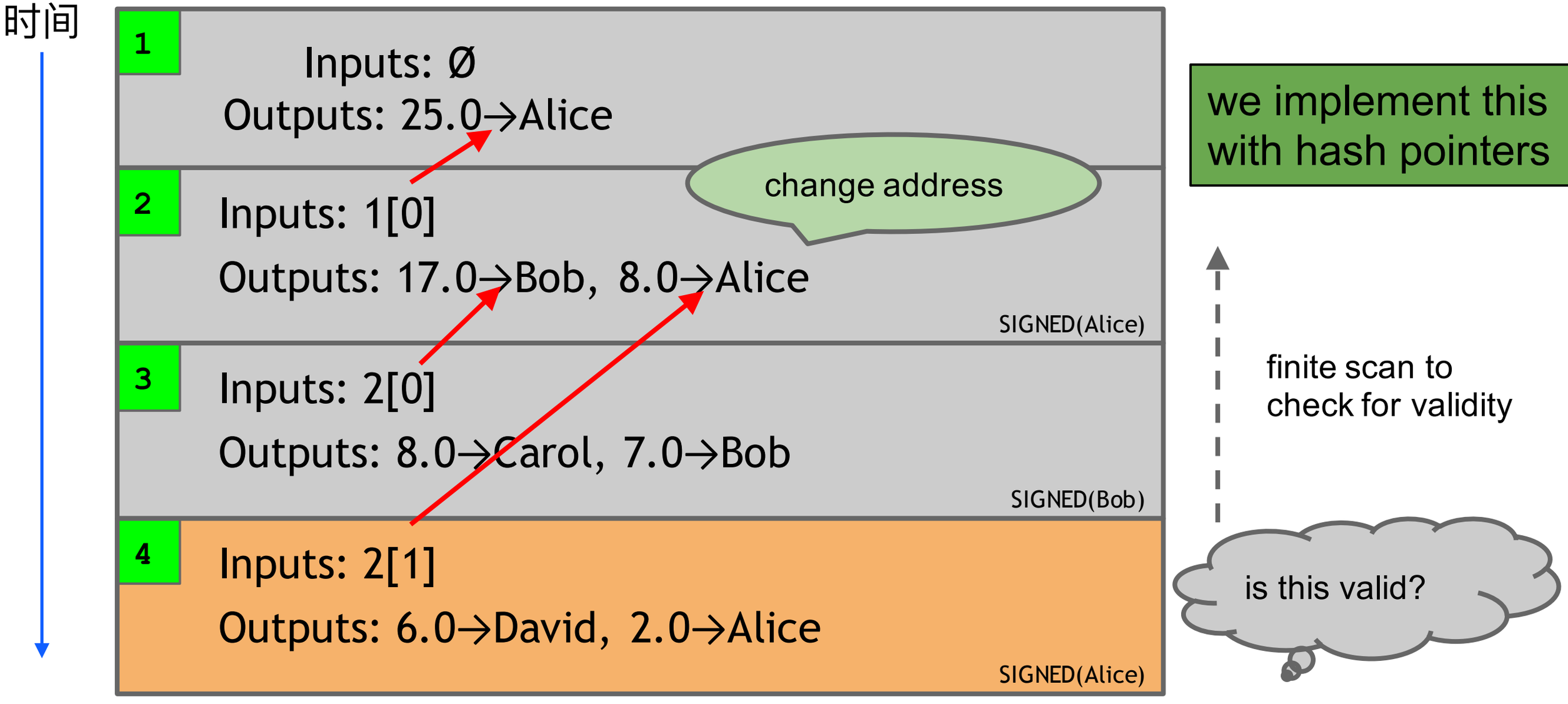
普通账本



一个块包含一个交易

交易验证需要扫描以前所有的块

比特币

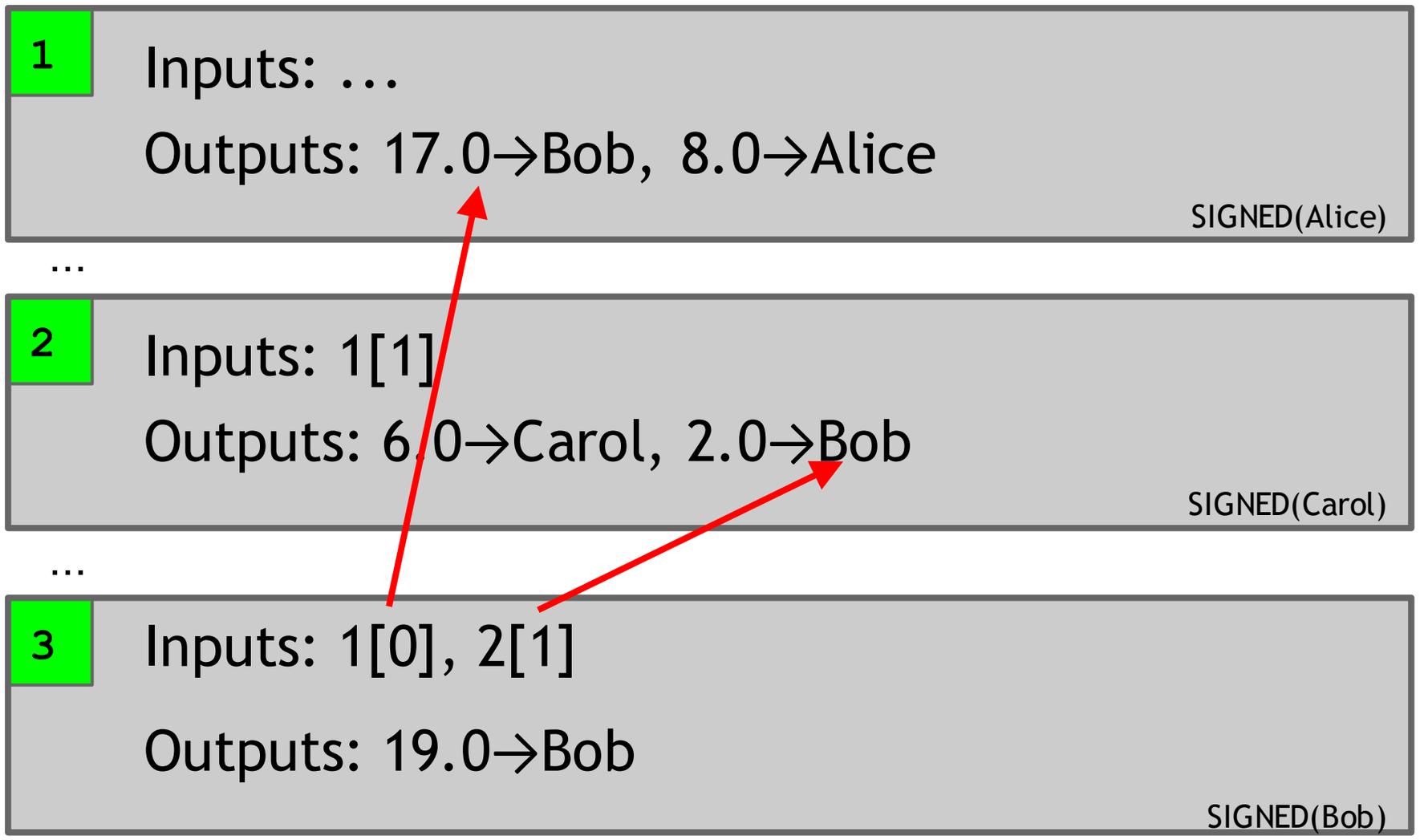


一个块包含一个交易

交易验证需要扫描以前所有的相关块

合并

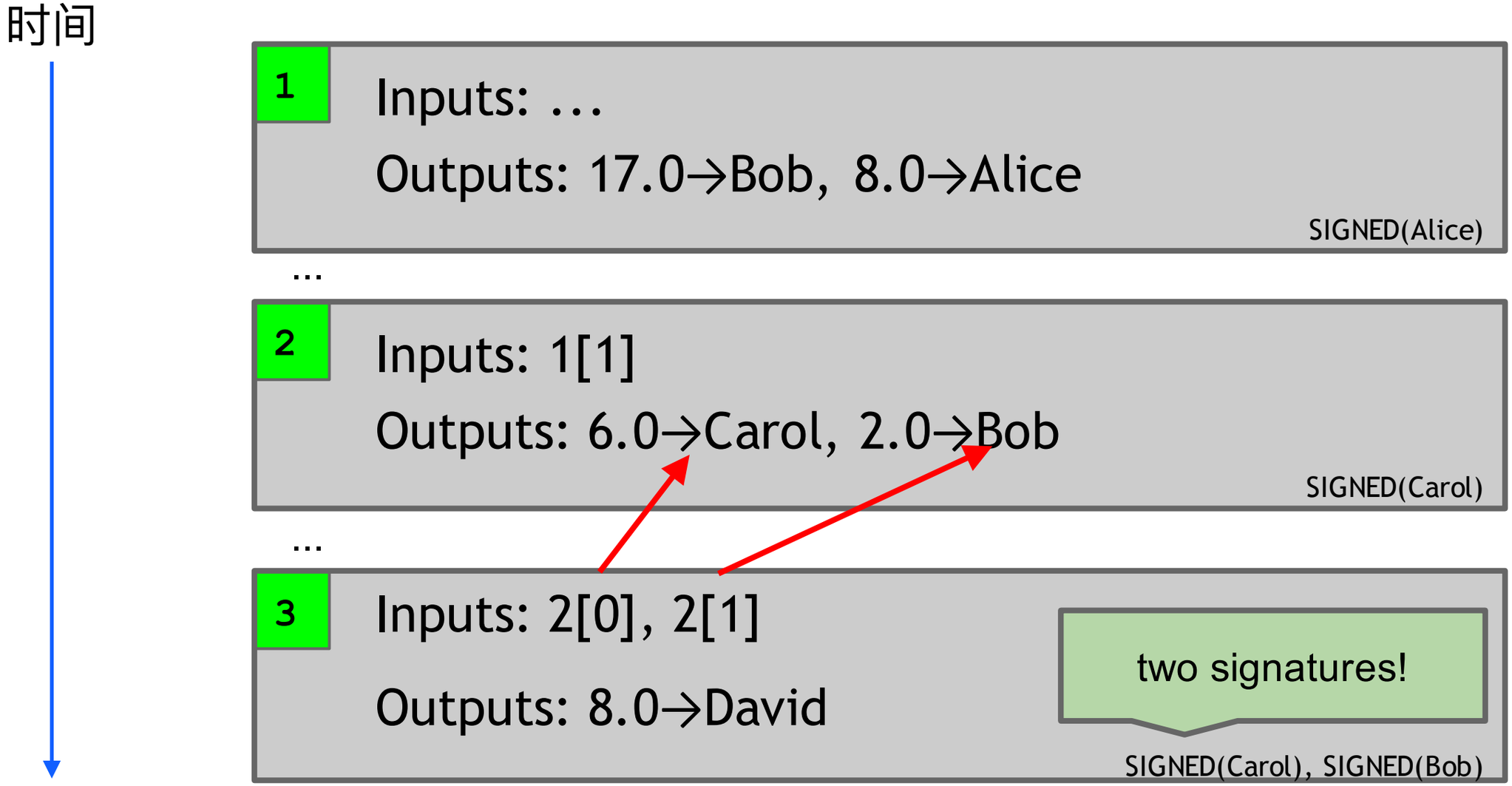
时间



一个块包含一个交易

交易验证需要扫描以前所有的相关块

联合支付



一个块包含一个交易

比特币交易

元数据

输入

输出

```
{
  "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
  "ver":1,
  "vin_sz":2,
  "vout_sz":1,
  "lock_time":0,
  "size":404,
  "in":[
   {
      "prev_out":{
      "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
      "n":0
     },
       "scriptSig":"30440..."
   },
   {
      "prev_out":{
      "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
      "n":0
     },
     "scriptSig":"3f3a4..."
   }
  ],
  "out":[
   {
     "value":"10.12287097",
     "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e
           OP_EQUALVERIFY OP_CHECKSIG"
   }
  ]
}
```

图3.3 一个真实的比特币交易程序段

比特币脚本

```
OP_DUP
OP_HASH160
69e02e18...
OP_EQUALVERIFY
OP_CHECKSIG
```

图3.4 P2PH脚本范例

```
<sig>
<pubKey>
----------------
OP_DUP
OP_HASH160
<pubKeyHash?>
OP_EQUALVERIFY
OP_CHECKSIG
```

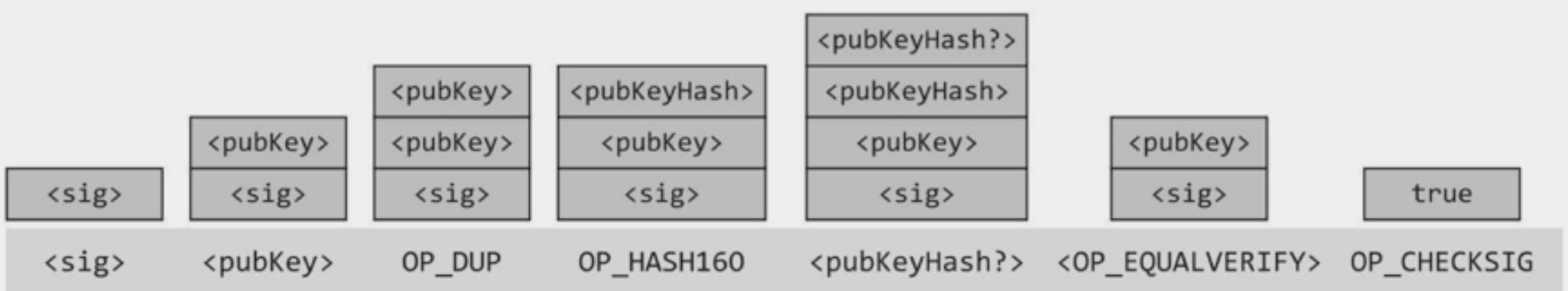
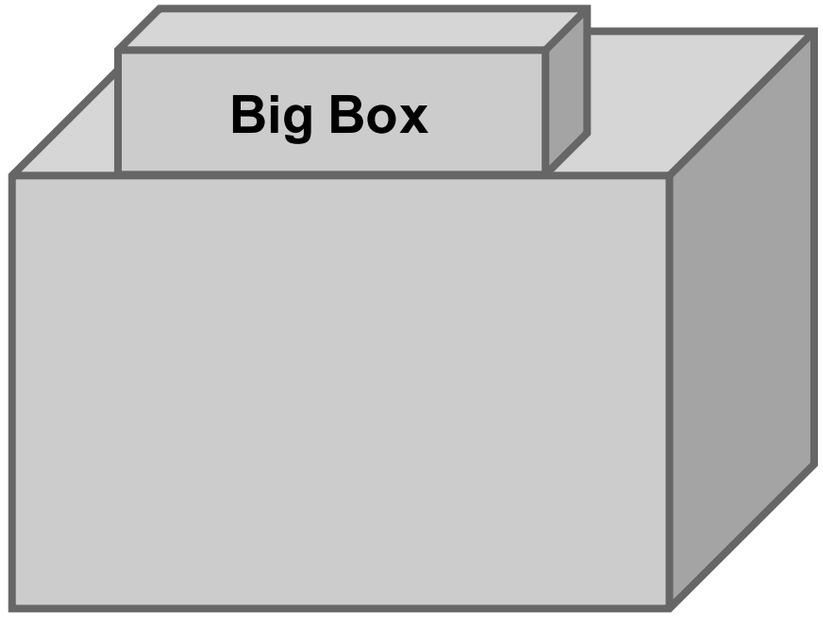


图3.6 比特币脚本的执行堆栈状态图

注：图中底部列出了相对应的指令：尖括号里的是数据指令，以OP开头的是工作码指令，指令上方对应的是指令执行之后的堆栈状态。

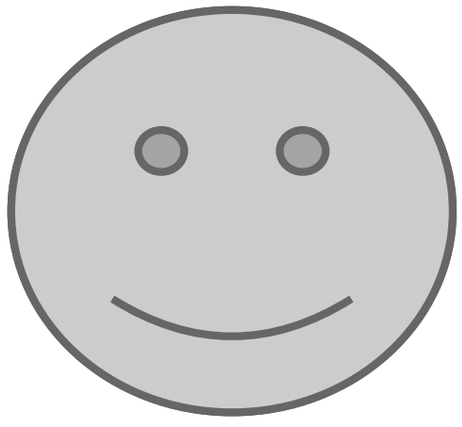
多重签名问题

I’m ready to pay for my purchases!



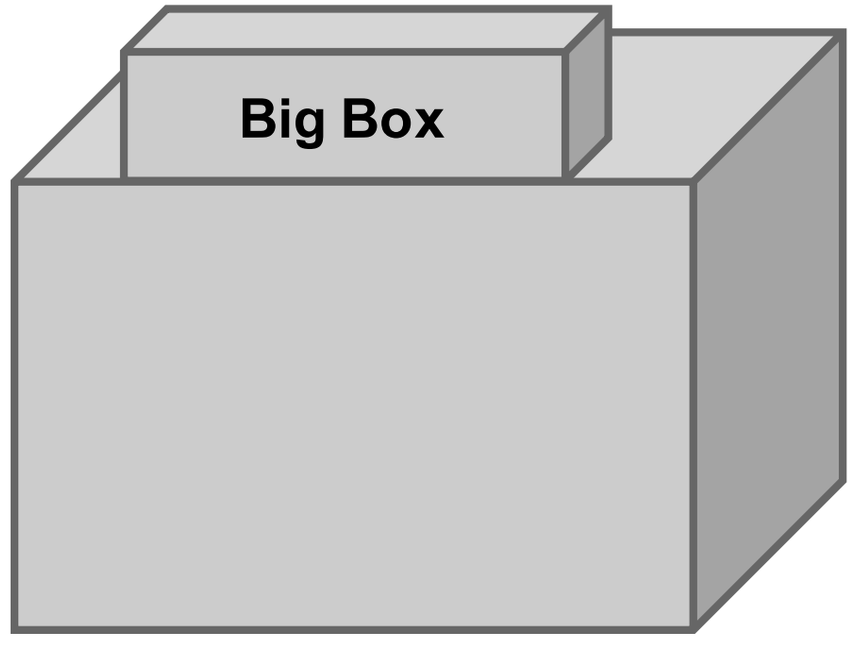
Cool! Well we’re using MULTISIG now, so include a script requiring 2 of our 3 account managers to approve. Don’t get any of those details wrong. Thanks for shopping at Big Box!

Pay for Script

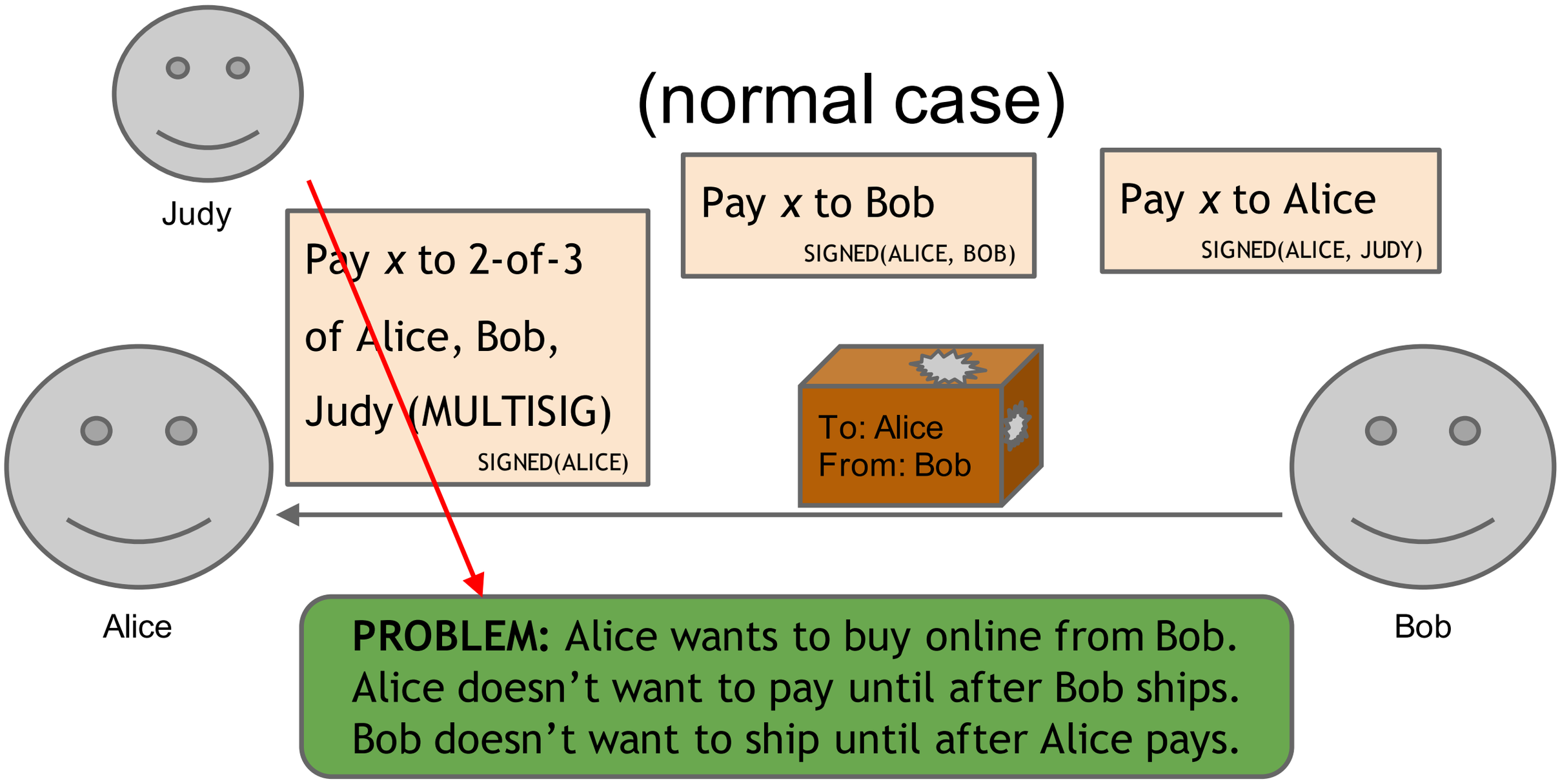


I’m ready to pay for my purchases!

Great! Here’s our address: 0x3454



托管


(normal case)
Judy
Pay x to 2-of-3 of Alice, Bob, Judy (MULTISIG)
SIGNED(ALICE)
Pay x to Bob
SIGNED(ALICE, BOB)
Pay x to Alice
SIGNED(ALICE, JUDY)
To: Alice
From: Bob
Alice
Bob
PROBLEM: Alice wants to buy online from Bob.
Alice doesn't want to pay until after Bob ships.
Bob doesn't want to ship until after Alice pays.

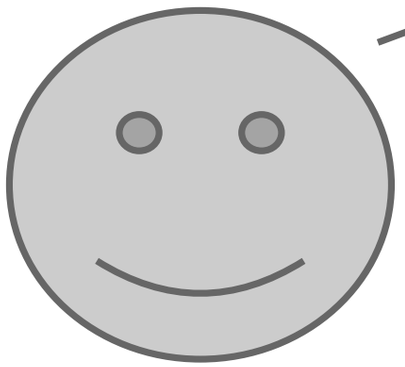
绿色地址

It’s me, Alice! Could you make out a green payment to Bob?

Bank

004 days since last double spend!

Faraday cage

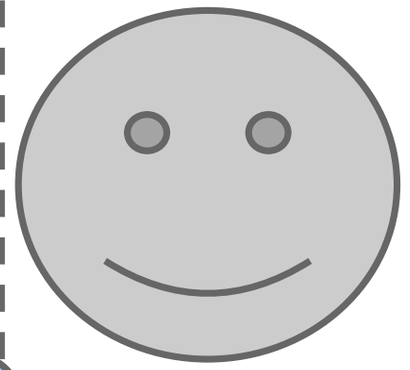


Alice

Pay x to Bob, y to Bank

No double spend

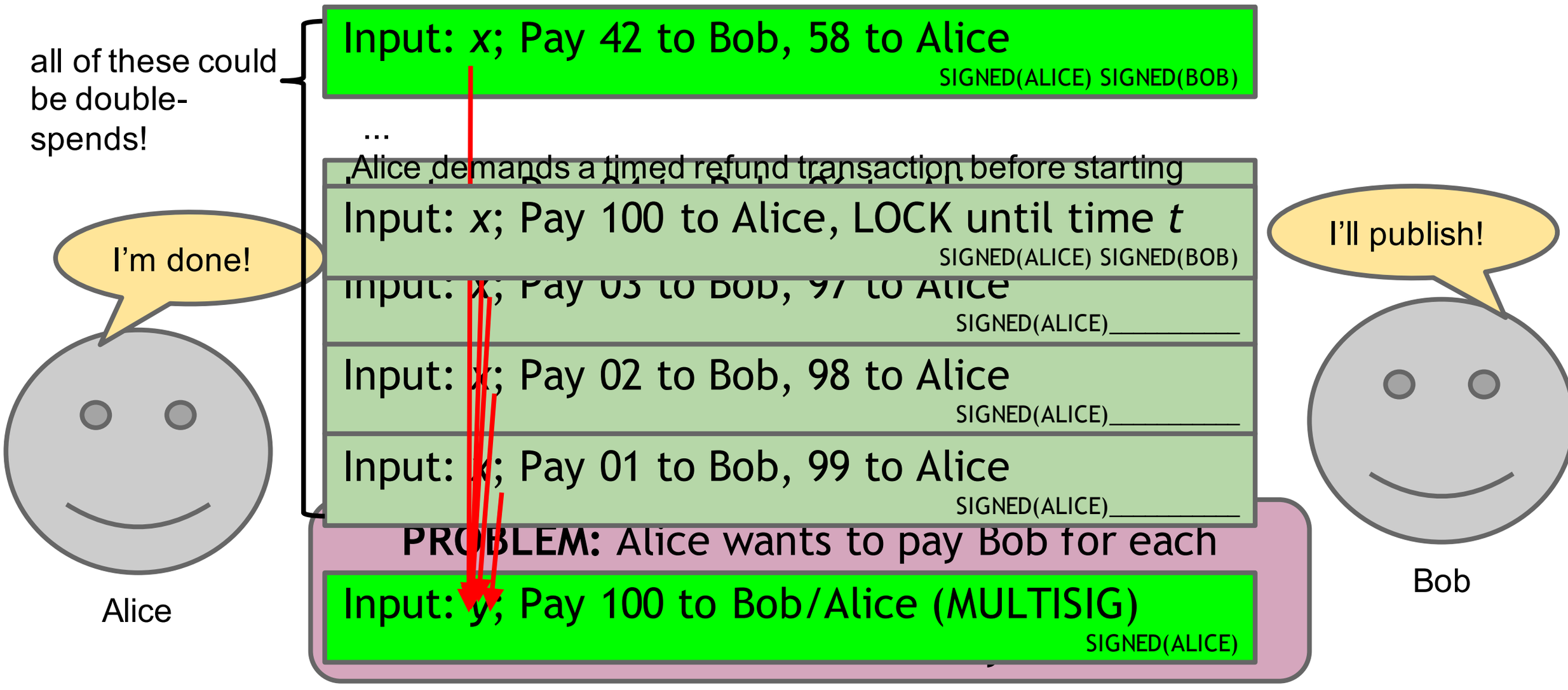
SIGNED(BANK)



Bob

PROBLEM: Alice wants to pay Bob.
Bob can’t wait 6 verifications to guard against double-spends, or is offline completely.

小额多次交易


all of these could be double-spends!
Input: x; Pay 42 to Bob, 58 to Alice
SIGNED(ALICE) SIGNED(BOB)
...
Alice demands a timed refund transaction before starting
Input: x; Pay 100 to Alice, LOCK until time t
SIGNED(ALICE) SIGNED(BOB)
Input: x; Pay 03 to Bob, 97 to Alice
SIGNED(ALICE)
Input: x; Pay 02 to Bob, 98 to Alice
SIGNED(ALICE)
Input: x; Pay 01 to Bob, 99 to Alice
SIGNED(ALICE)
PROBLEM: Alice wants to pay Bob for each
Input: y; Pay 100 to Bob/Alice (MULTISIG)
SIGNED(ALICE)
I'm done!
I'll publish!
Alice
Bob

锁定时间


```
{
    "hash":"5a42590...b8b6b",
      "ver":1,
      "vin_sz":2,
      "vout_sz":1,
      "lock_time":315415,
      "size":404,
...
}
```


Block index or real-world timestamp before which this transaction can’t be published

比特币的区块结构

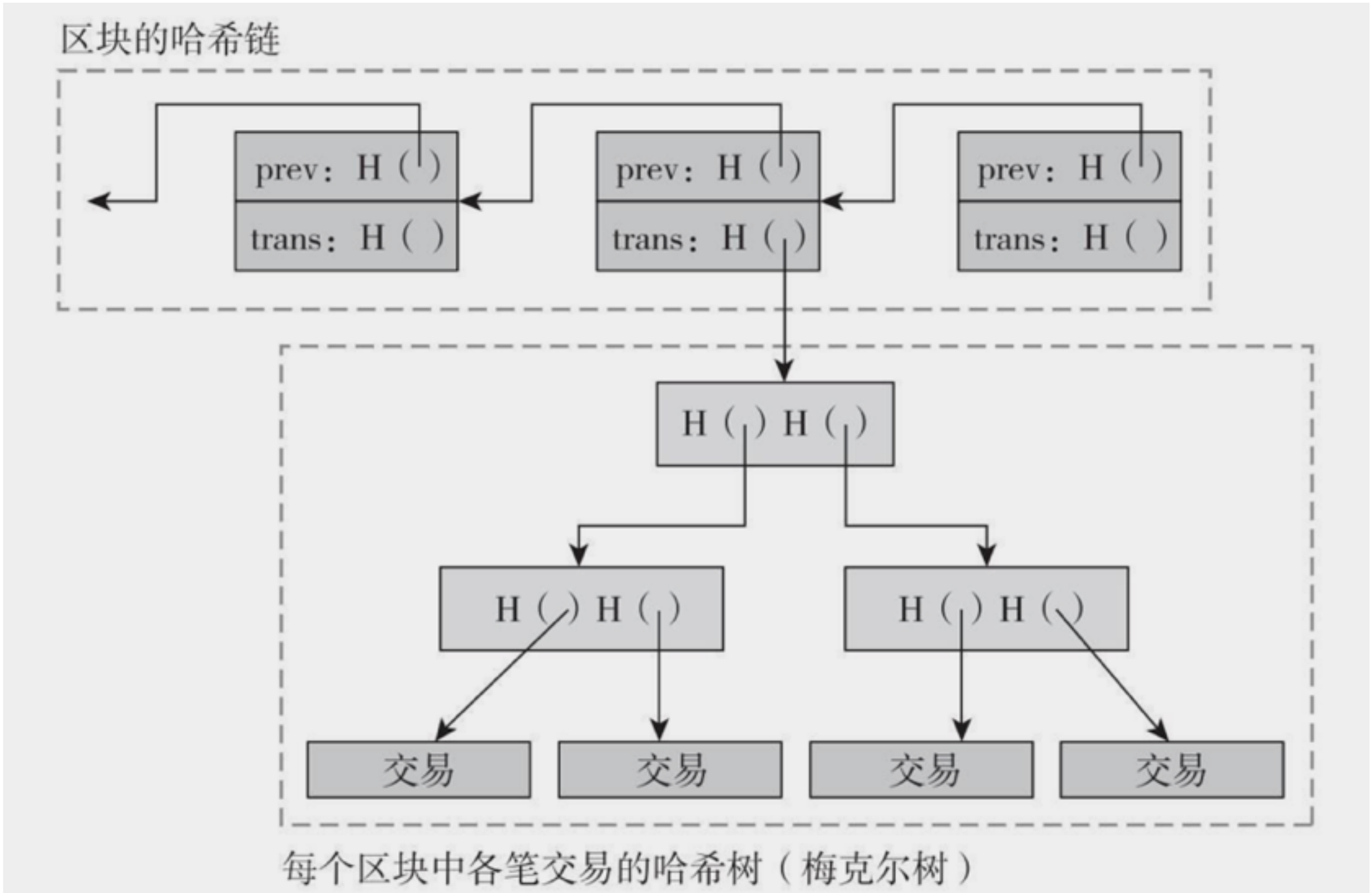


图3.7 比特币的区块链有两个哈希结构

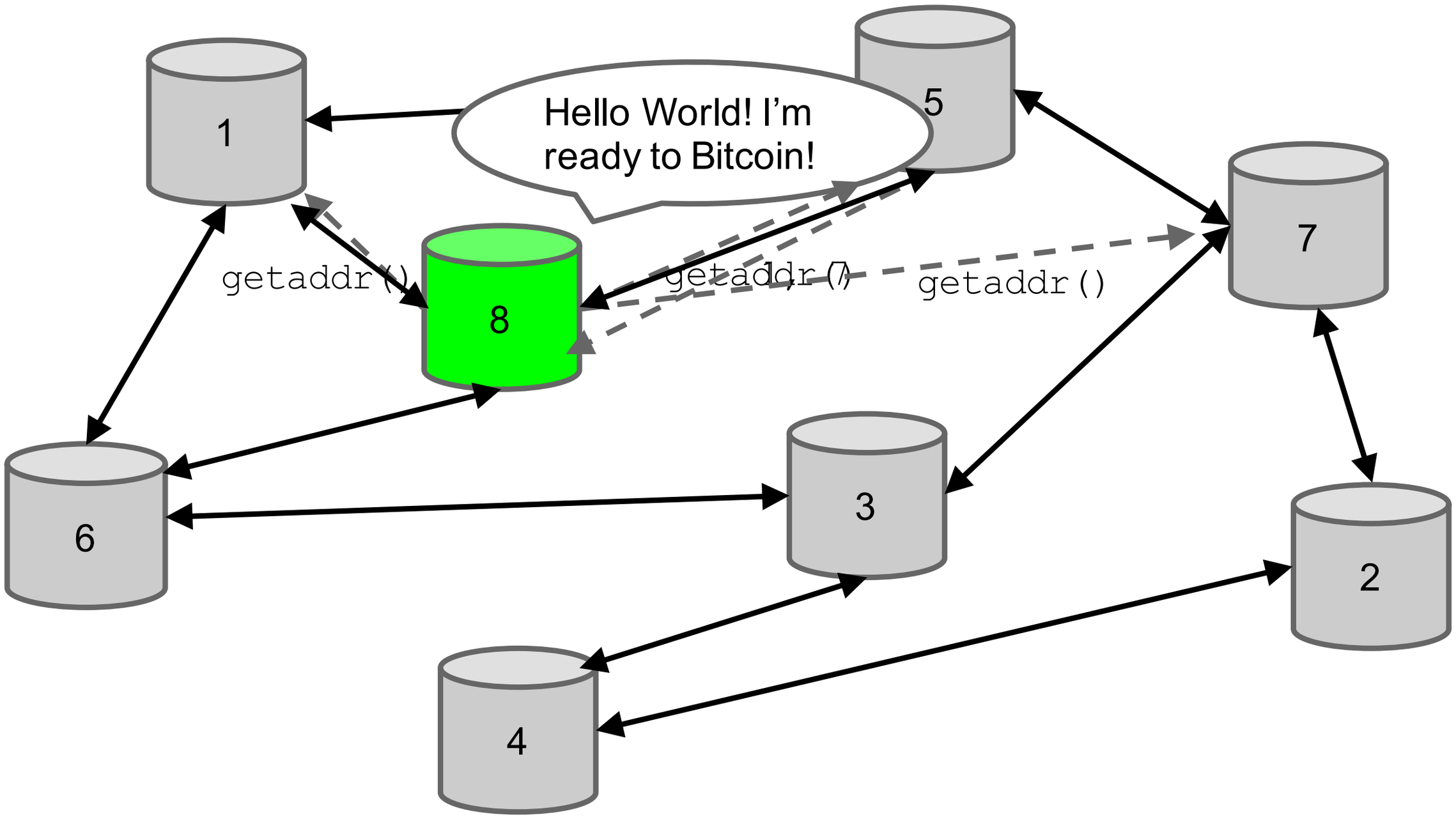
注：一个就是把区块联结在一起的哈希链，另一个就是区块内部的交易哈希值梅克尔树。

比特币输入输出

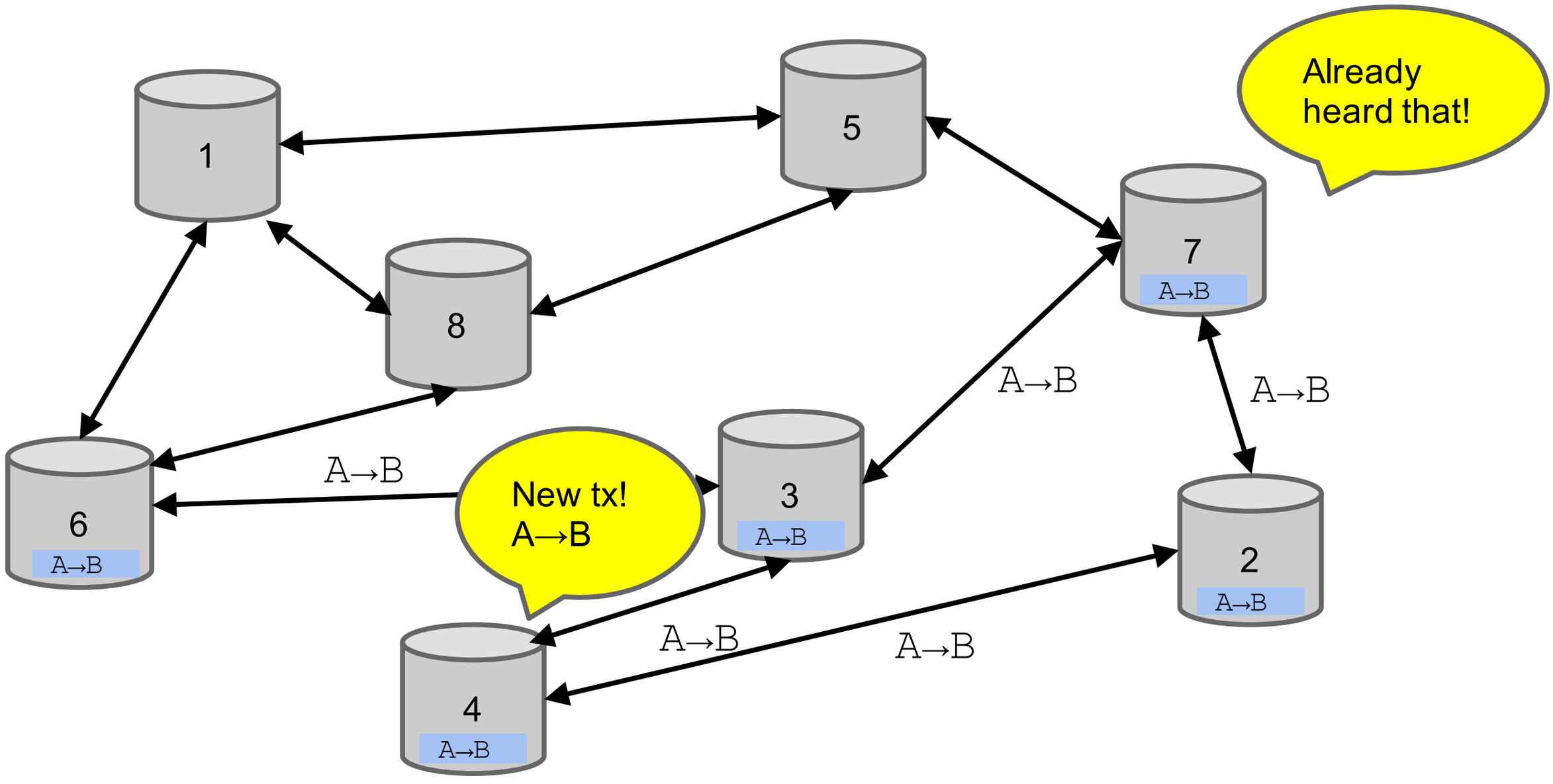
```
"in":[
        {
          "prev_out":{
            "hash":“000000.....0000000",
            "n":4294967295
          },
           "coinbase":"..."
      },
      [
  "out":[
        {
          "value":"25.03371419",
          "scriptPubKey":"OPDUP OPHASH160 ... ”
        }
       ]
```

图3.8 币基交易

比特币网络


1
Hello World! I'm ready to Bitcoin!
5
getaddr()
8
getaddr()
getaddr()
7
6
3
2
4

比特币网络交易消息传播



块传播

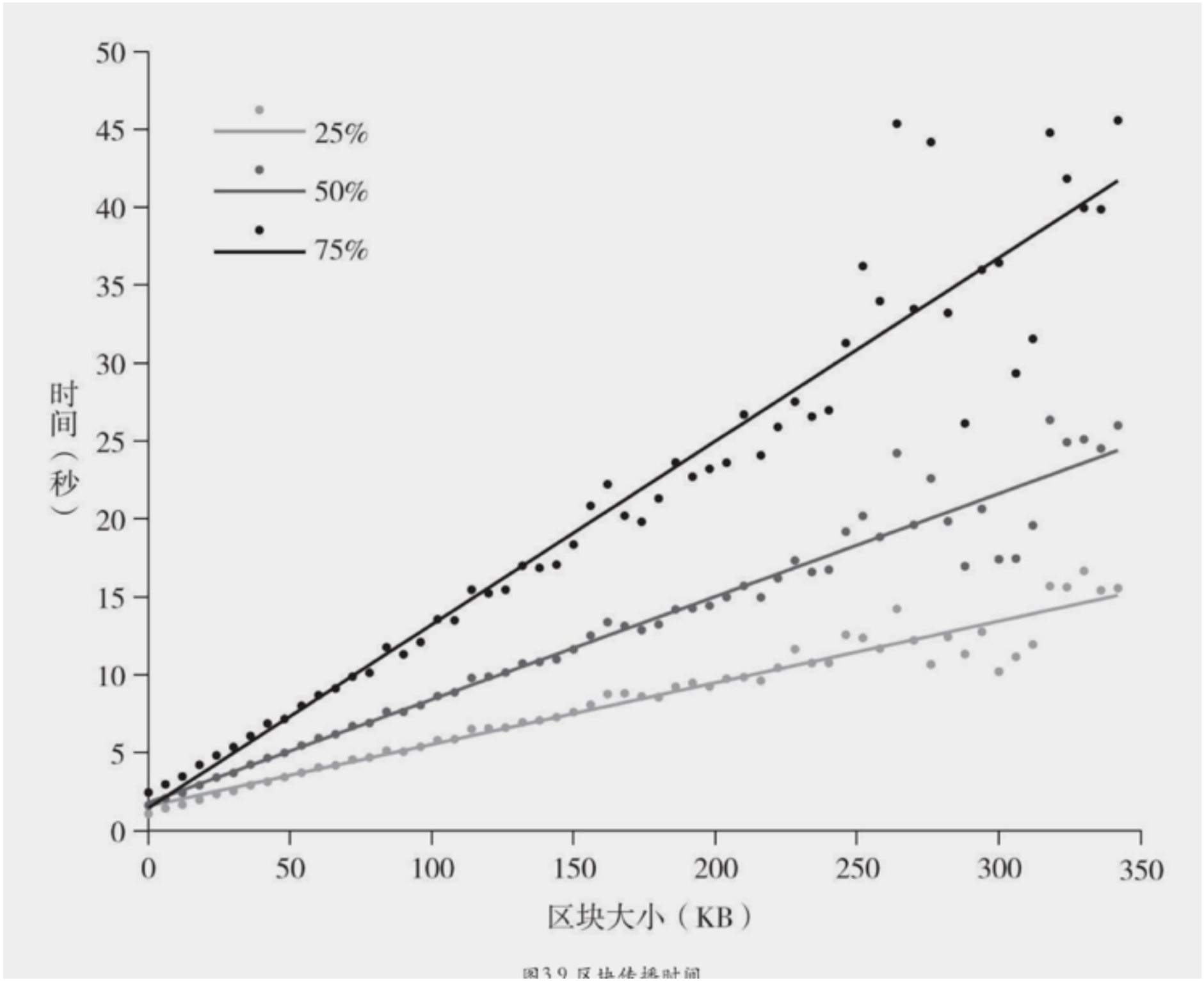


图3.9 区块传播时间

存储花费

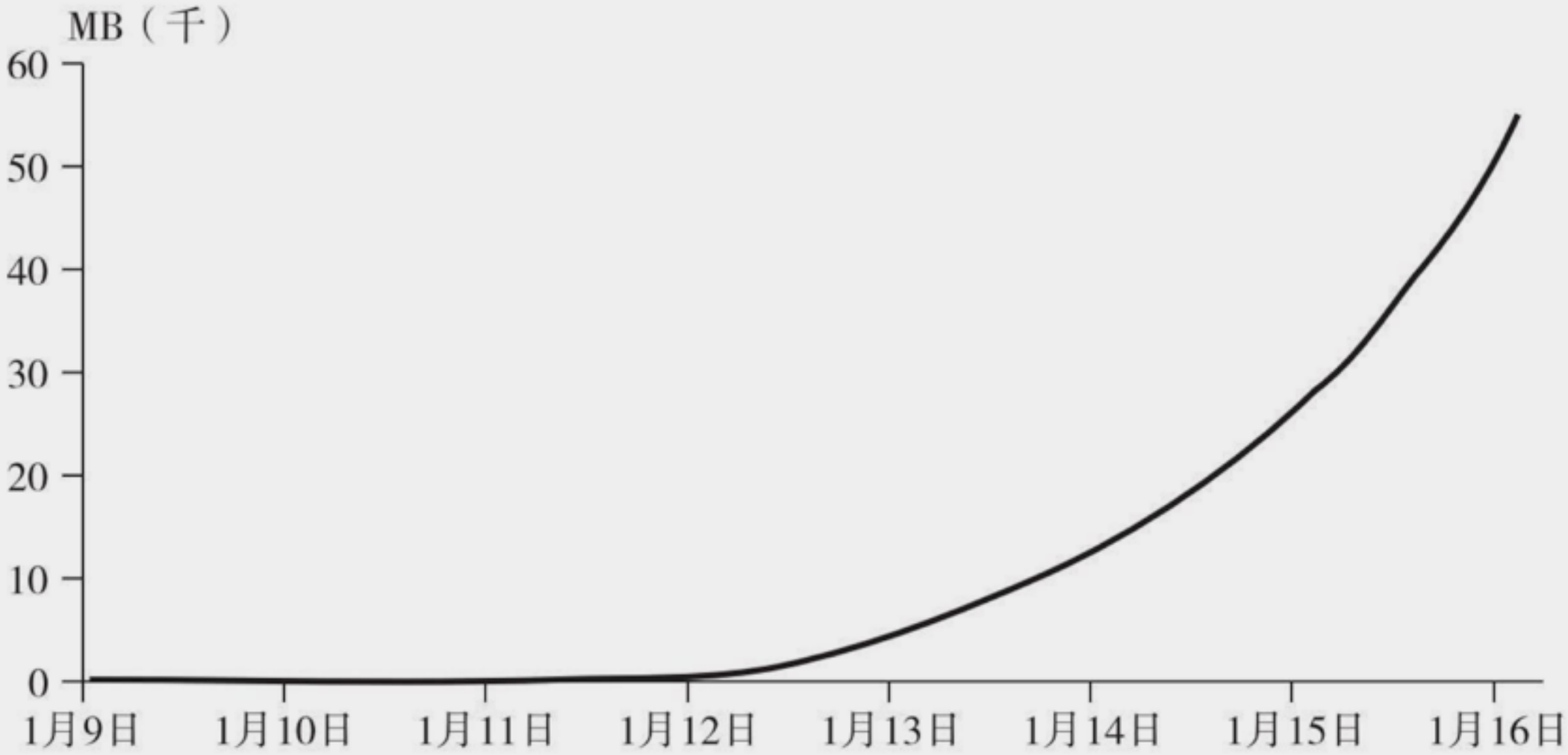


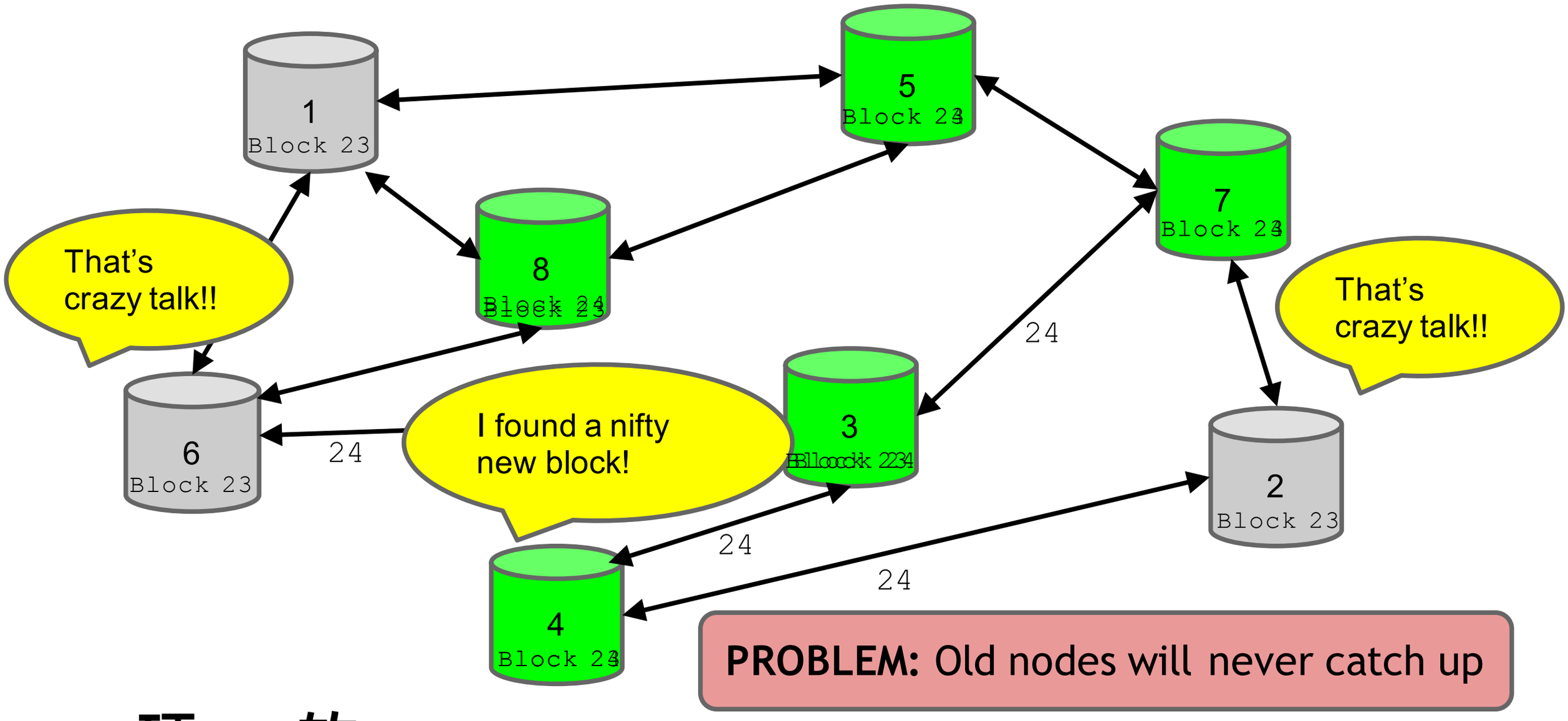
图3.10 区块链的大小

注：全节点必须保持整个区块链，在2015年年底，区块链大小在50GB以上。

比特币限制

- *10*分钟：产生块的间隔
- *1M*：一个快大小
- 2万签名：每个快
- *100M satoshi*：每个币
- *23M*：比特币再大
- *50*、*25*、*12.5*….：挖矿奖励
- *250bytes*：每个业务
- *7*交易：每秒*(visa 2*千到*1*万，*Paypal 50-100)*

分叉


1
Block 23
5
Block 24
7
Block 24
8
Block 24
That's crazy talk!!
That's crazy talk!!
24
6
Block 23
24
I found a nifty new block!
3
Block 24
2
Block 23
24
24
4
Block 24
PROBLEM: Old nodes will never catch up

硬 *vs.* 软

如何存储和使用比特币

比特币存储

Hot storage

online

hot secret key(s)

cold address(es)

payments

Cold storage

offline

威胁

Charles Ponzi

交易所


東京で MT.GOX のデモ
へ参加してください。
東京都渋谷区渋谷
2丁目11-5
MTGOX
WHERE IS
OUR MONEY
VAIO

Homework 课后作业

- 要求阅读如下论文：

➡ *Bitcoin: Under the Hood. In CACM 2015.*

下次上课测试！

谢谢！

孙惠平

sunhp@ss.pku.edu.cn