比特币与它背后的区块链技术

The Bitcoin and Blockchain

目录

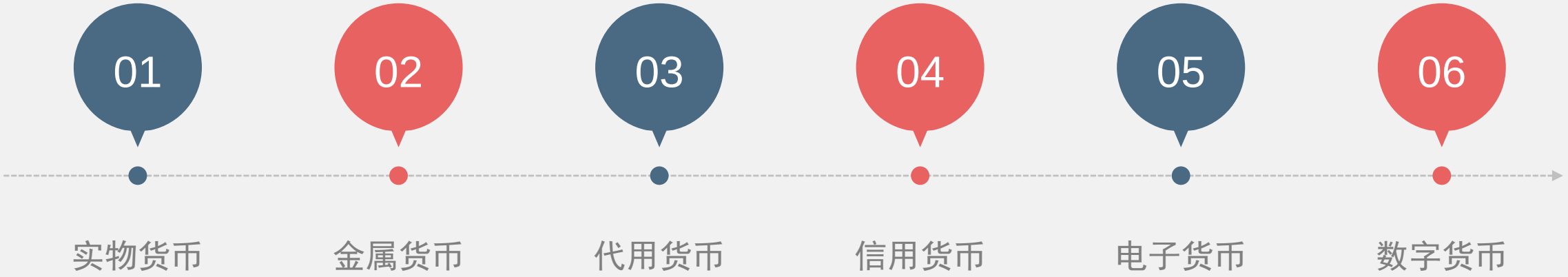
CONTENTS

01

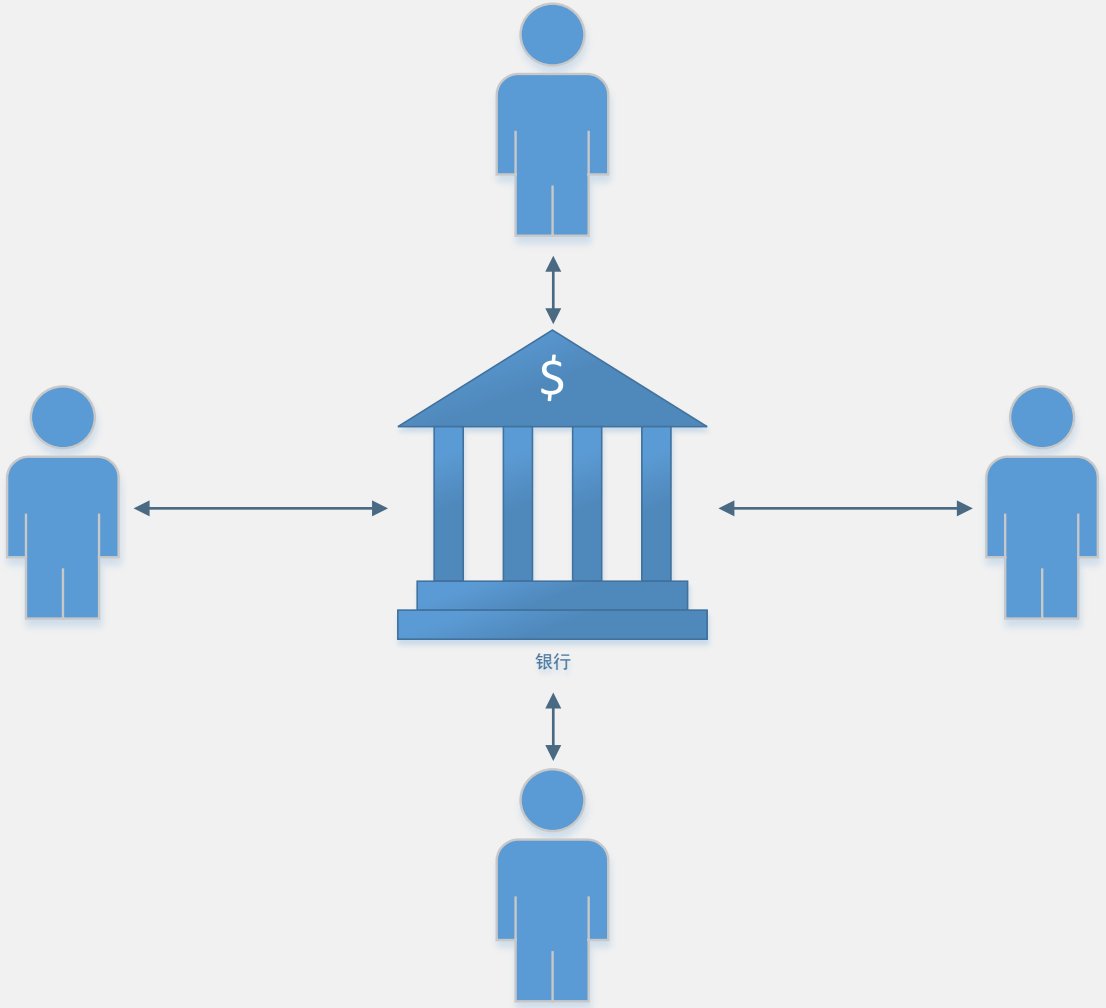
比特币介绍

一个前所未有的社会学实验

从货币的历史说起



现在的银行货币系统



现在的银行货币系统

> 中心场景下，实际上是假定存在一个安全可靠的第三方记账机构来实现，这个机构利用信用作为抵押，来完成交易。

但是，很多时候...

- 贸易两国可能缺乏足够的外汇储备；
- 网络上的匿名双方进行直接买卖；
- 交易的两个机构彼此互不信任；
- 汇率的变化；
- 可能无法连接到第三方的系统；
- 第三方的系统可能会出现故障
- ……

去中心化设计
真的不容易...

1. 货币防伪：谁来负责验证货币
2. 货币交易：如何确定货币从一方转移到另外一方
3. 双重支付：如何避免出现双重支付

比特币出现之前的数字货币

1983

e-Cash
首个匿名化
数字加密货币

1997

Hashcash
首次提出 PoW
工作量证明机制

1998

B-money
首个去中心化
设计的数字货币

2008

Bitcoin
首次从实践意义上实现了一套去中心化的数字货币系统

比特币起源

“

2008年，一位化名为中本聪的人，在一篇名为《比特币：一个点对点的电子现金系统》的论文中首先提出了比特币。中本聪结合以前的多个数字货币发明，如 B-money 和 HashCash，创建了一个完全去中心化的电子现金系统，不依赖于通货保障或是结算交易验证保障的中央权威。关键的创新是利用分布式计算系统（称为“工作量证明”算法）每隔 10 分钟进行一次的全网“选拔”，能够使去中心化的网络同步交易记录。这个能优雅的解决双重支付问题，即一个单一的货币单位可以使用两次。此前，双重支付问题是数字货币的一个弱点，并通过一个中央结算机构清除所有交易来处理。

”

比特币发展路线图

2008.11

中本聪发表一篇关于点对点现金支付的论文

2009.1

中本聪在赫尔辛基的一台服务器上挖出了第一 批50个比特币

2010.5

第一次比特币交易：程序员Laszlo Hanyecz 用1万BTC购买了价值25美元的披萨优惠券

2011.2

价格首次达1美元，与美元等价

比特币发展路线图

2011.6

黑客攻击了MT.Gox，6万个用户数据被泄漏，8750000美元的帐户受到影响，比特币价格迅速降到0.01$/BTC

2012.11

区块供应量首次减半调整，从之前每10分钟50个递减至25个，此时比特币价格为12.35美元

2013.11

价格突破1000美元，价格首次超过黄金价格

2013.12

中国央行发声，明确不支持比特币在中国的合法地位，比特币暴跌

比特币发展路线图

2014.2

全球最大比特币平台MT.GOX由于安全因素被黑客洗劫70万比特币，宣告破产

2016.7

区块供应量再次减半调整，递减至12.5个，此时比特币价格为576.45美元

2017.6

价格突破3000美元，超过20000人民币

区块链技术引起了世界关注

目录

CONTENTS

02

比特币使用体验

使用比特币钱包收款、付款

比特币钱包

加入比特币网络并开始使用通货，所有用户需要做的就是下载应用程序或使用在线应用程序。

完整客户端 01

轻量级客户端 02

在线客户端 03

Choose your Bitcoin wallet

Find your wallet and start making payments with merchants and users.

Mobile Desktop Hardware Web


Bitcoin Knots


Bitcoin Core


Bitcoin Wallet


breadwallet


Bither


GreenBits


BitGo


Green Address


Coin.Space


Simple Bitcoin

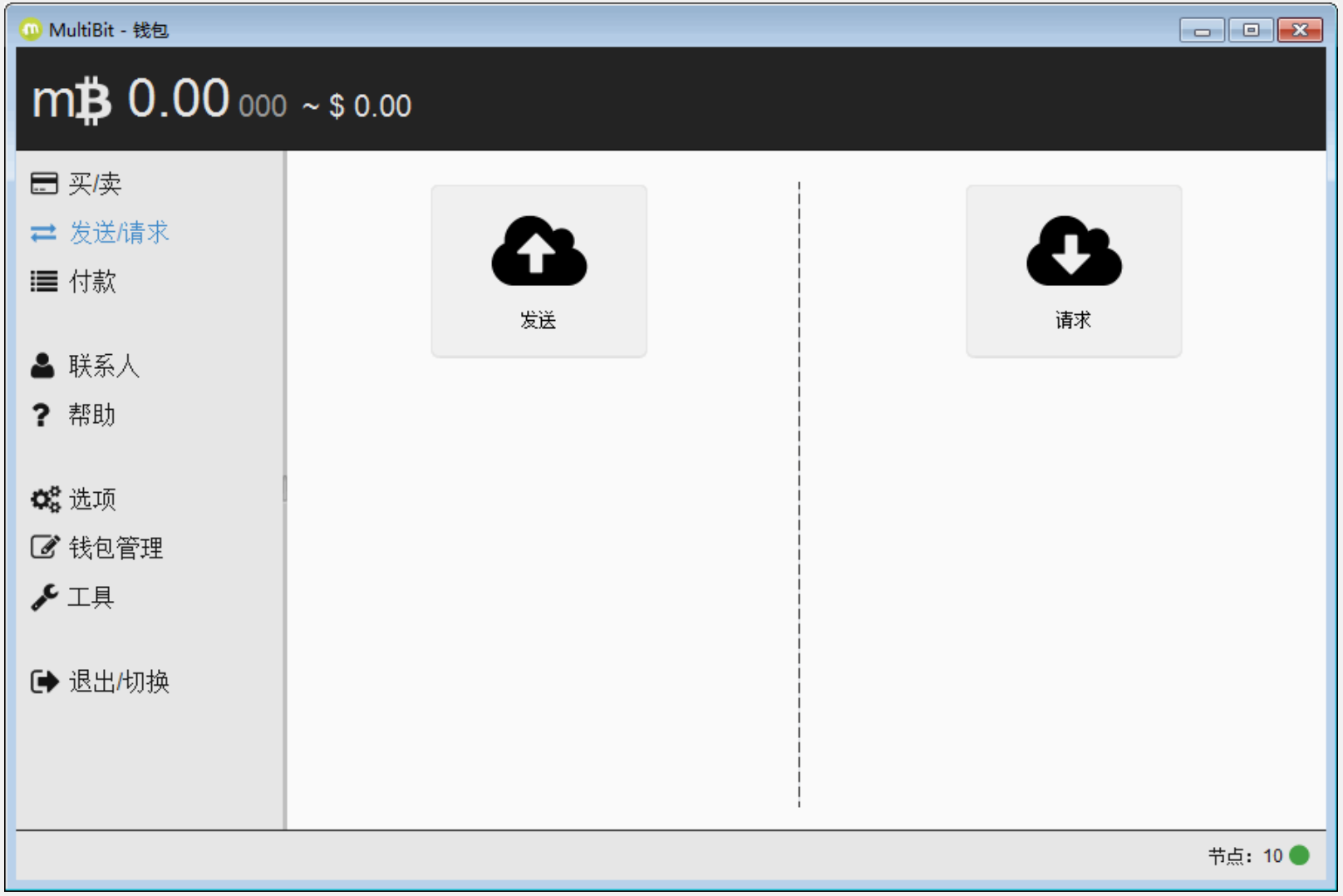

Armory


Electrum


mSIGNA


ArcBit

比特币钱包


MultiBit - 钱包
mB 0.00 000 ~ $ 0.00
买/卖
发送/请求
付款
联系人
帮助
选项
钱包管理
工具
退出/切换
发送
请求
节点：10

获取比特币

Alice

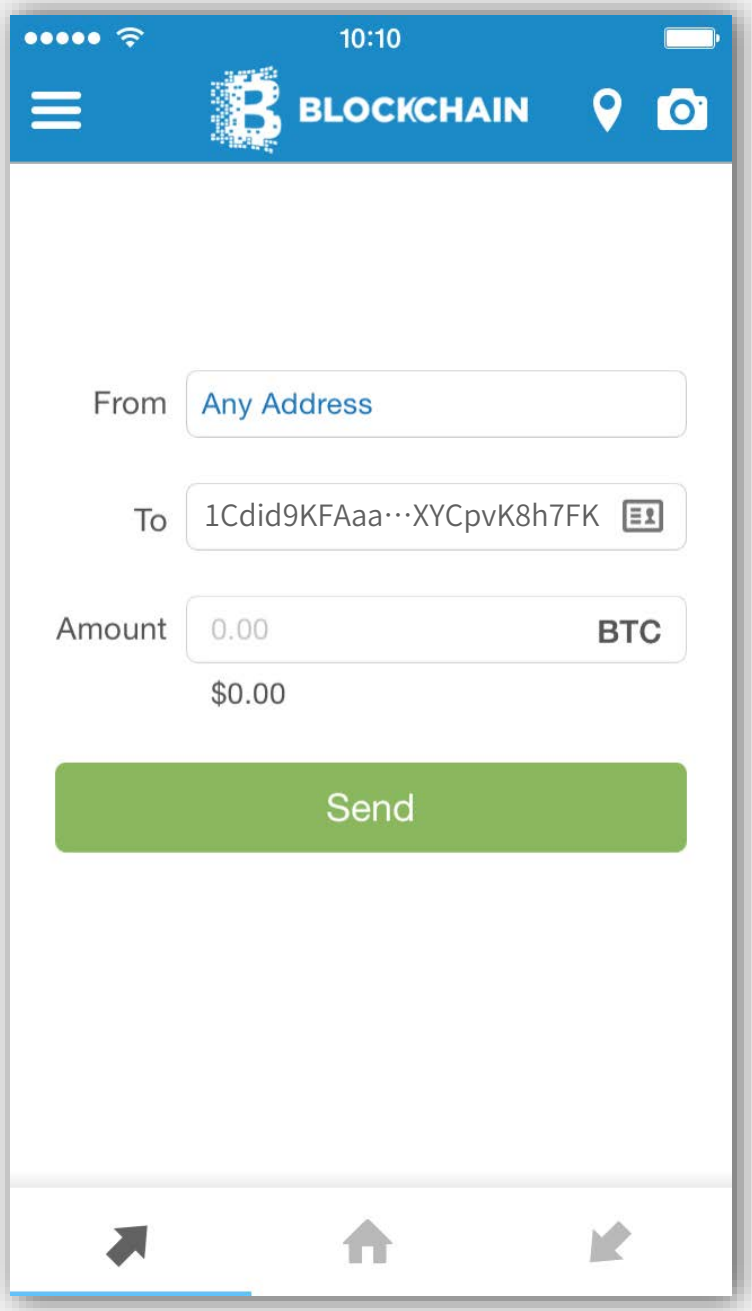
1. 使用现金向朋友购买比特币
2. 在比特币交易平台购买比特币
3. 出售某种产品或服务来换取比特币

获取比特币


Bitcoin (24h)
USD ▲ 2.41%
$2349.72
EUR €2040.30
CNY ¥16139.34
GBP £1804.32
$2407
$2343
$2279


200 $
0.1 BTC
Alice
Joe

获取比特币



Joe 通过自己的比特币移动钱包

向

Alice 的比特币地址发送比特币

消费比特币

Alice 确认收到比特币之后，就可以去消费一杯咖啡了。

然而，

比特币的这些交易是如何完成的呢？

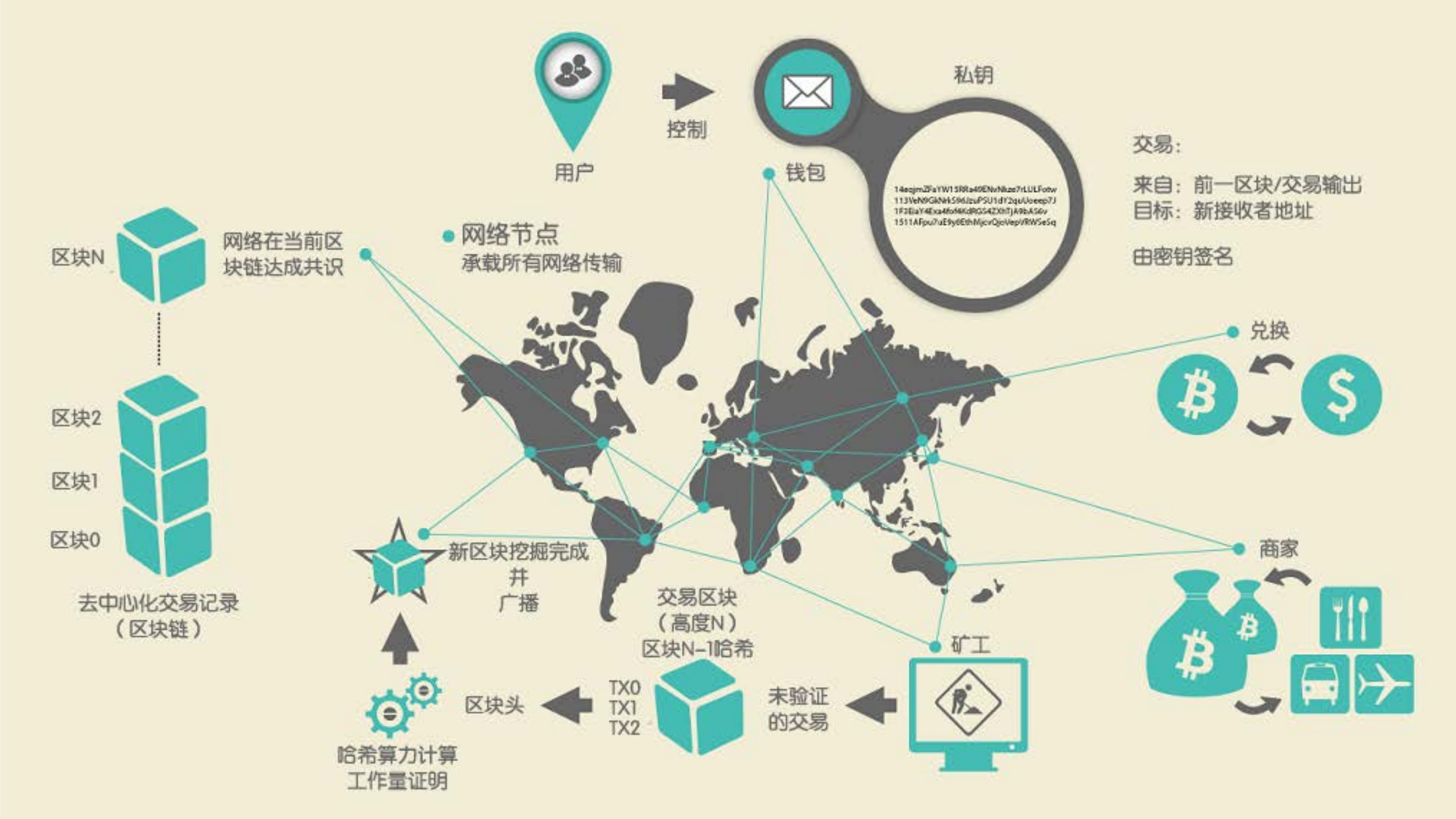
目录

CONTENTS

03

比特币工作原理

比特币钱包、交易、挖矿共识和区块链


用户
控制
私钥
钱包
交易：
来自：前一区块/交易输出
目标：新接收者地址
由密钥签名
网络节点
承载所有网络传输
网络在当前区块链达成共识
区块N
区块2
区块1
区块0
去中心化交易记录（区块链）
兑换
商家
新区块挖掘完成并广播
交易区块（高度N）
区块N-1哈希
矿工
TX0
TX1
TX2
区块头
未验证的交易
哈希算力计算
工作量证明

比特币工作原理

1 比特币钱包

2 比特币交易

3 区块链

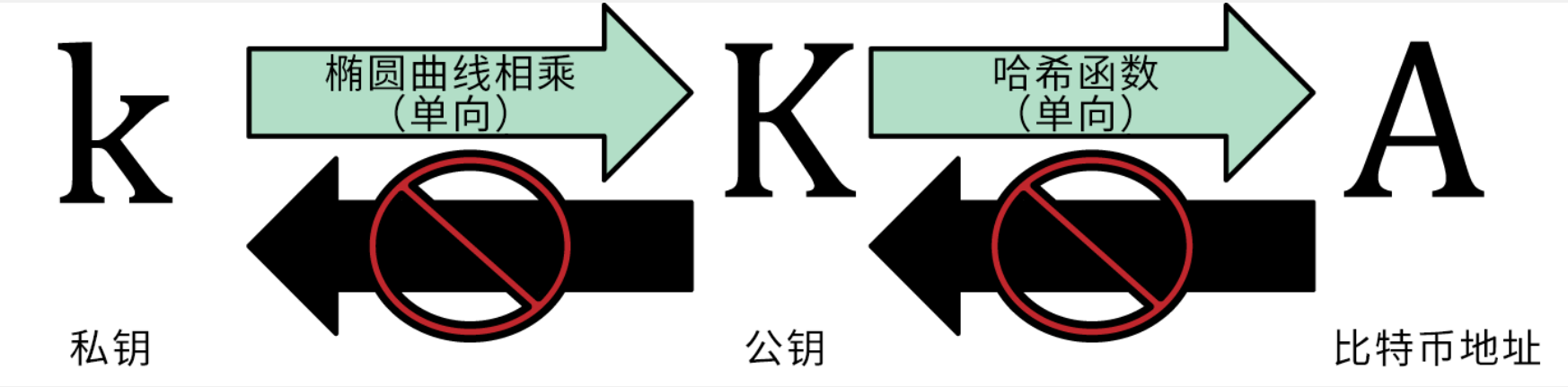
4 挖矿共识

比特币钱包

钱包是一个存储私钥的容器，里边没有比特币。

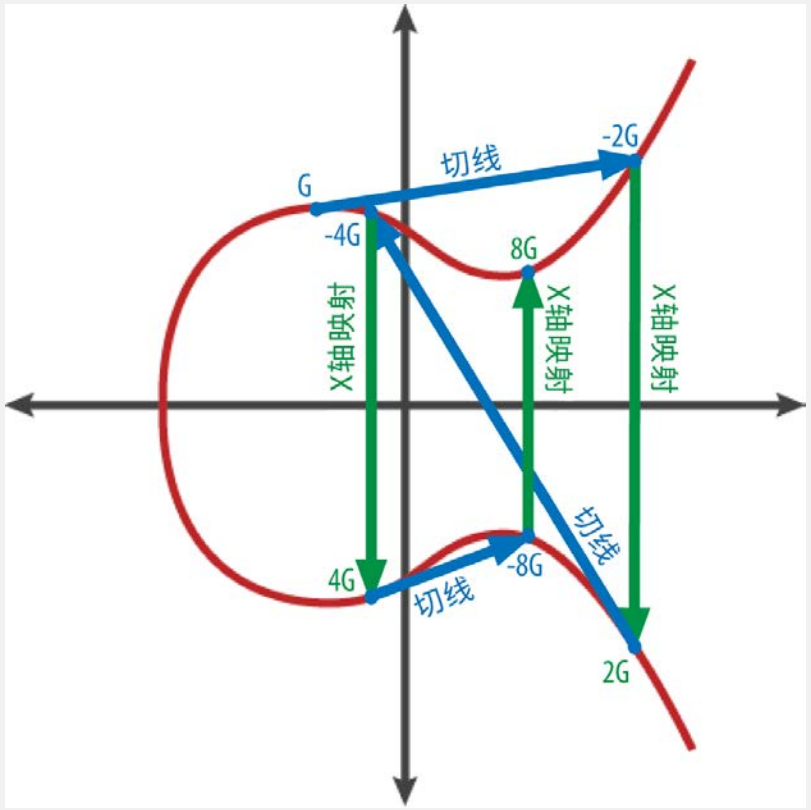
私钥用于付款中的数字签名，公钥对应的比特币地址用于收款。

比特币钱包



私钥、公钥和比特币地址之间的关系

比特币钱包



$y^2 \bmod p = (x^3 + 7) \bmod p$

K = k * G

其中，**k** 是私钥，**G** 是被称为生成点的常数点，**K** 是所得公钥

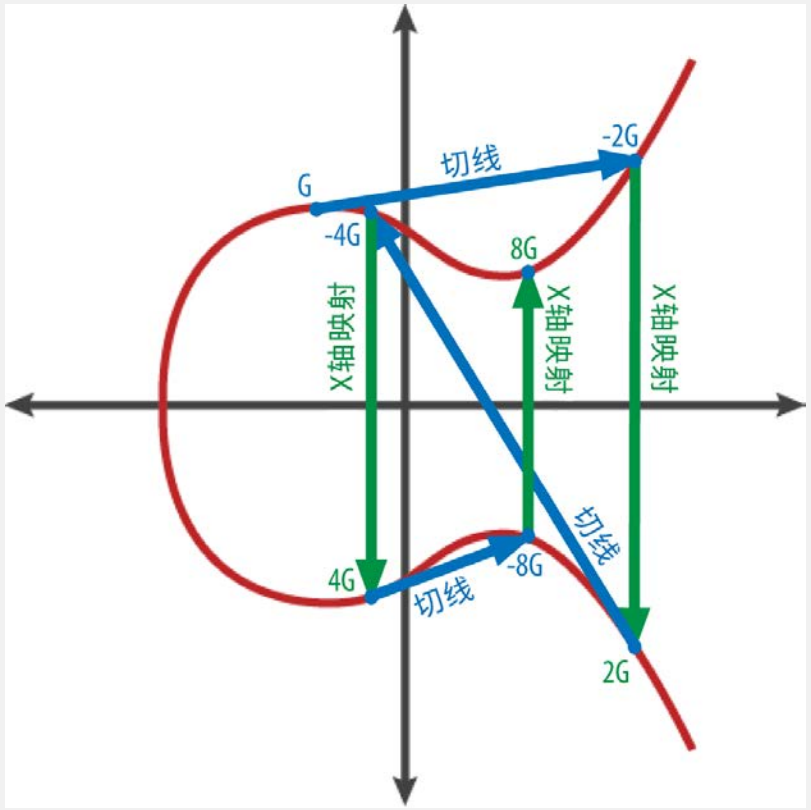
k = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD

K = (x, y)，其中：

x = F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A

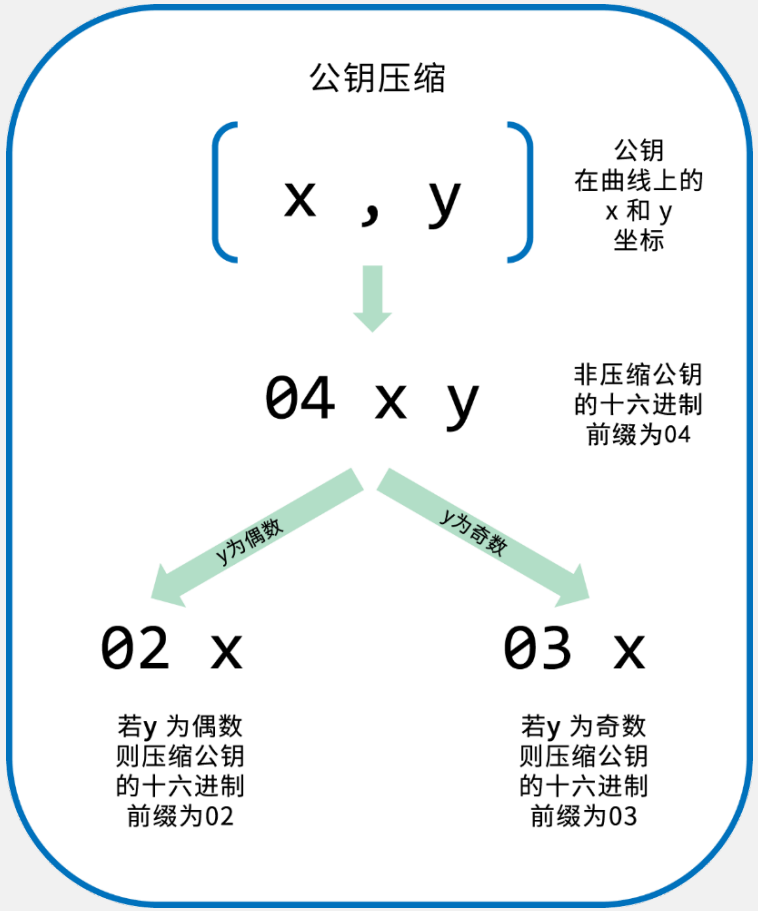
y = 07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB

比特币钱包



$$y^2 \bmod p = (x^3 + 7) \bmod p$$

K = (F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A,
07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB)



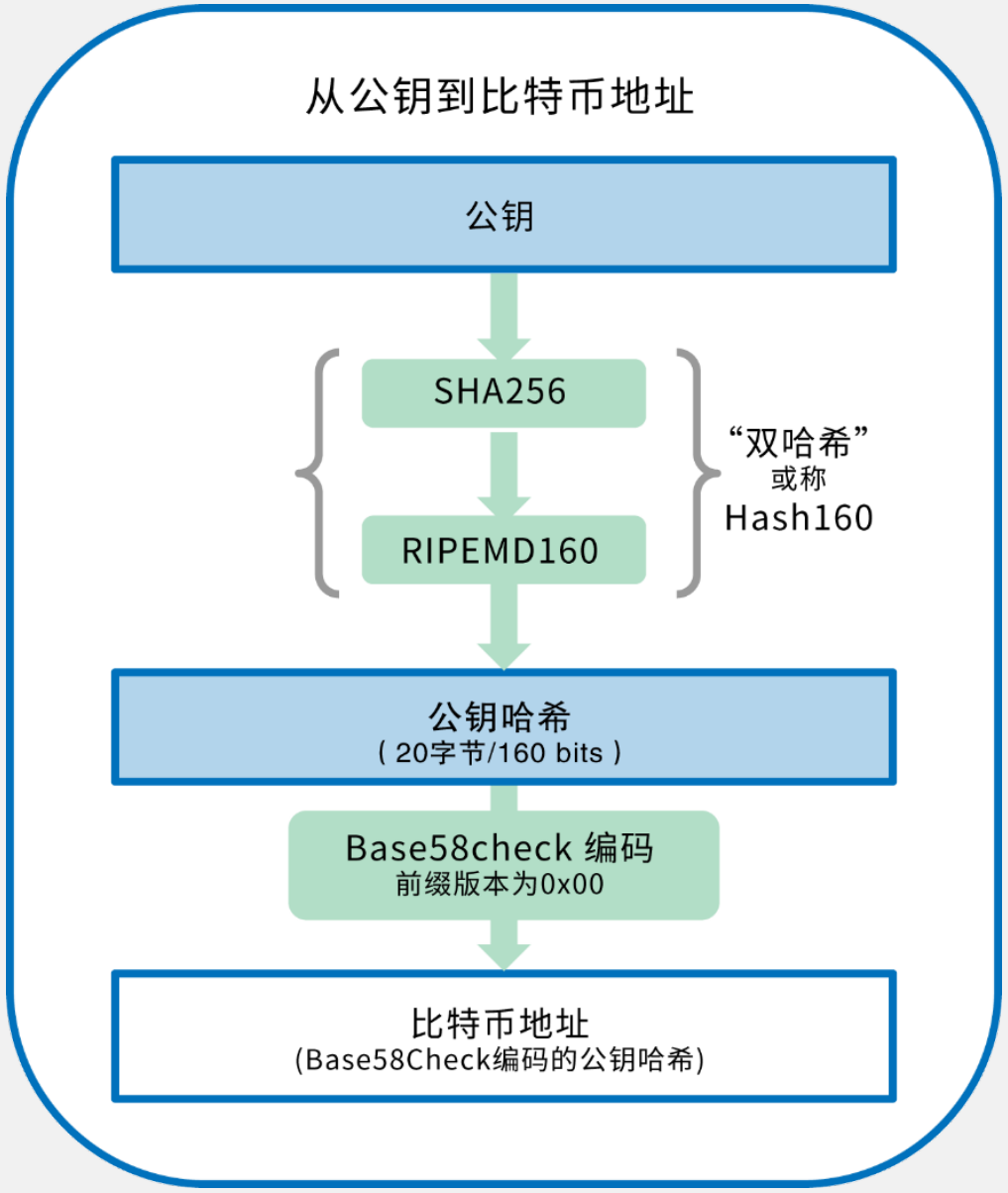
K = 02F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A
或
K = 03F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A

比特币钱包

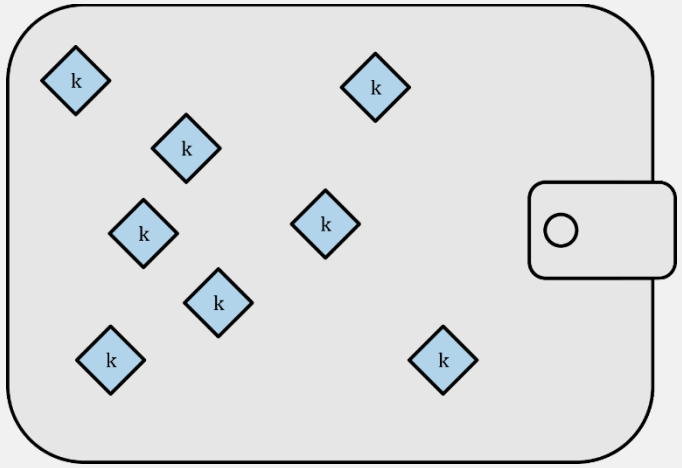
A = RIPEMD160(SHA256(K))

1J7mdg5rbQyUHENYdx39WVWK7fsLpEoXZy

Base58Check 编码用于提高可读性、避免歧义



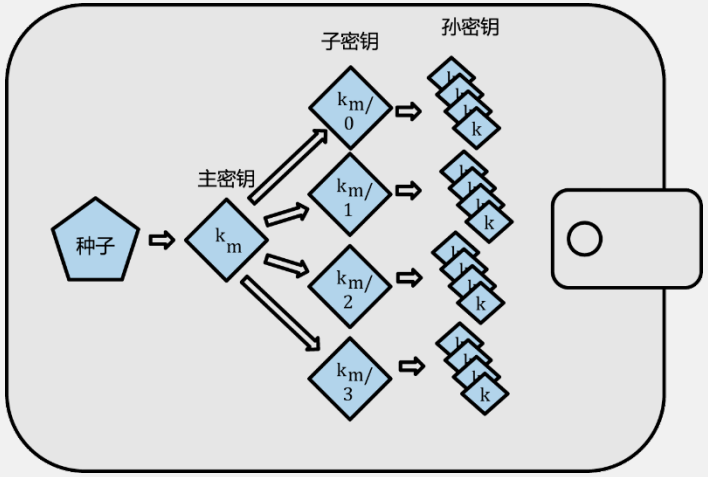
比特币钱包



第一种钱包
非确定性（随机）钱包

仅存储一些随机产生的私钥，每个私钥仅使用一次即作废。缺点是难以管理、备份以及导入。这种钱包需要经常性地备份，每一把密钥都必须备份，否则一旦钱包不可访问时，钱包所控制的资金就付之东流。现在正在被确定性钱包替换。

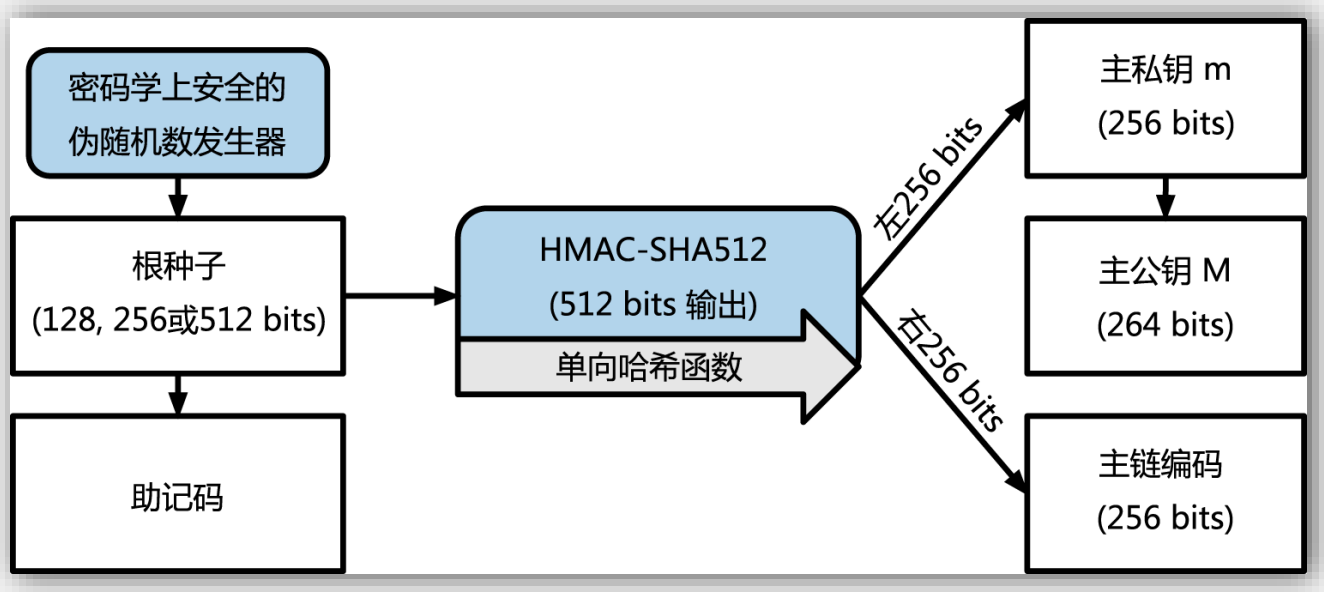
比特币钱包

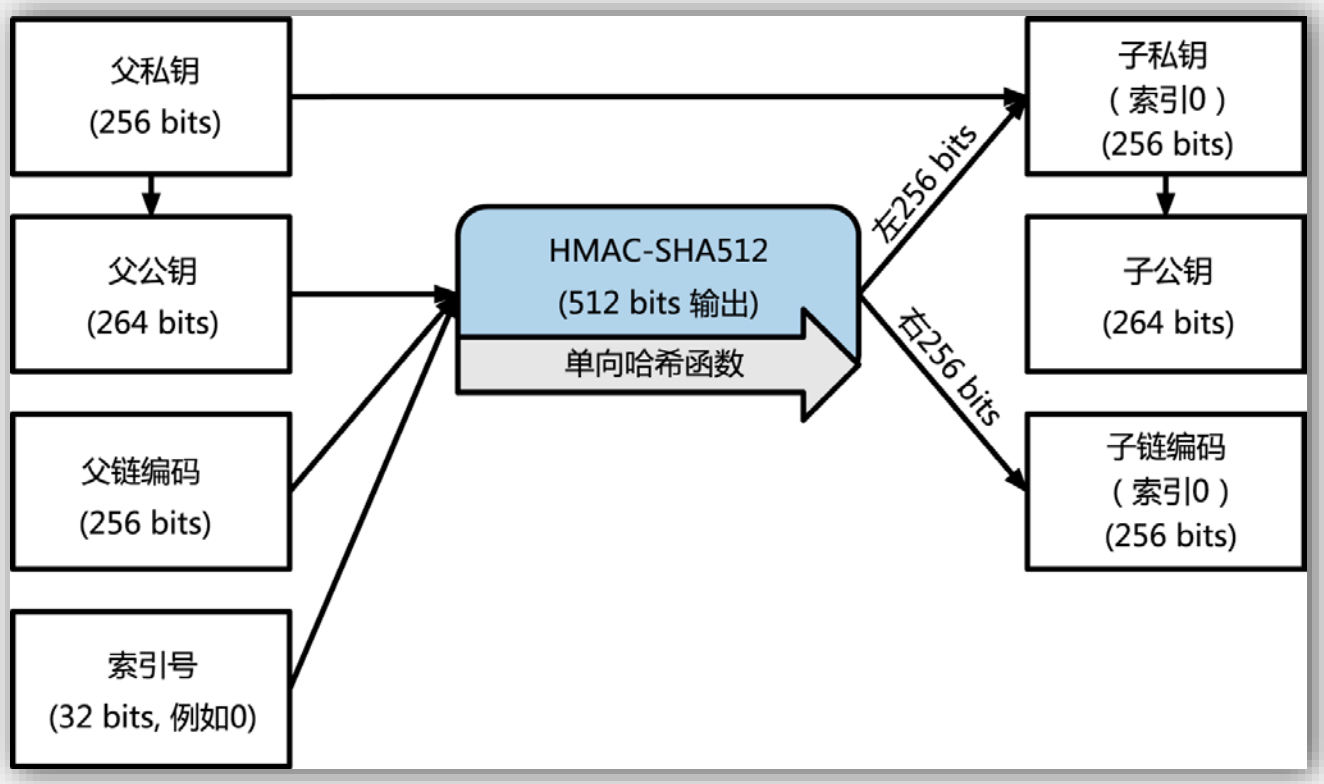


第二种钱包
确定性（种子）钱包

钱包中所有的私钥均由一个种子产生，种子是一个随机数，对其进行哈希运算得到第一个私钥，后续每个私钥均是前一个私钥的哈希运算结果，这样钱包里的私钥就形成了一条完整的私钥链条。只要用户记住链条的头，就可以轻松恢复出整个钱包中的所有私钥。

分层确定性钱包


密码学上安全的
伪随机数发生器
根种子
(128, 256或512 bits)
助记码
HMAC-SHA512
(512 bits 输出)
单向哈希函数
左256 bits
右256 bits
主私钥 m
(256 bits)
主公钥 M
(264 bits)
主链编码
(256 bits)


父私钥
(256 bits)
父公钥
(264 bits)
父链编码
(256 bits)
索引号
(32 bits, 例如0)
HMAC-SHA512
(512 bits 输出)
单向哈希函数
左256 bits
右256 bits
子私钥
(索引0)
(256 bits)
子公钥
(264 bits)
子链编码
(索引0)
(256 bits)

助记码

示例：army van defense carry jealous true garbage claim echo media make crunch

助记码和种子的创建过程如下：

1.创造一个128到256位的随机顺序（熵）；

2.提出SHA256哈希前几位，就可以创造一个随机序列的校验和；

3.把校验和加在随机顺序的后面；

4.把顺序分解成11位的不同集合，并用这些集合去和一个预先已经定义的2048个单词字典做对应。

5.生成一个12至24个词的助记码。

负熵输入	0c1e24e5917779d297e14d45f14e1a1a
种子	3338a6d2ee71c7f28eb5b882159634cd46a898463e9d2d0980f8e80dfbba5b0fa0291e5fb888a599b44b93187be6ee3ab5fd3ead7dd646341b2cdb8d08d13bf

纸钱包

普通纸钱包

加密纸钱包，密码是“test”

硬件钱包


Send 0.541476 BTC to
1MJ2tj2ThBE62zXbBYA5ZaN3fdve5CPAzl


Enter New PIN
4 8 5
2 6 9
1 7 3

比特币工作原理

1 比特币钱包

2 比特币交易

3 区块链

4 挖矿共识

交易生命周期

交易是比特币系统中最重要的部分。系统中任何其他的部分都是为了确保比特币交易可以被生成、能在比特币网络中得以传播和通过验证，并最终添加入全球比特币交易总账簿。

01

创建交易

可以在线创建，可以离线创建。但是要有资金所有者的私钥进行签名

02

广播交易

被创建的交易会被广播到在线网络的所有比特币节点，这个动作由客户端发起

03

交易传播

收到交易信息的节点首先验证信息的正确性，验证通过则将此交易继续广播

04

记录交易

每一个收到该交易且通过了对其验证的节点均尝试将这交易记录到最终的账本中

交易结构体

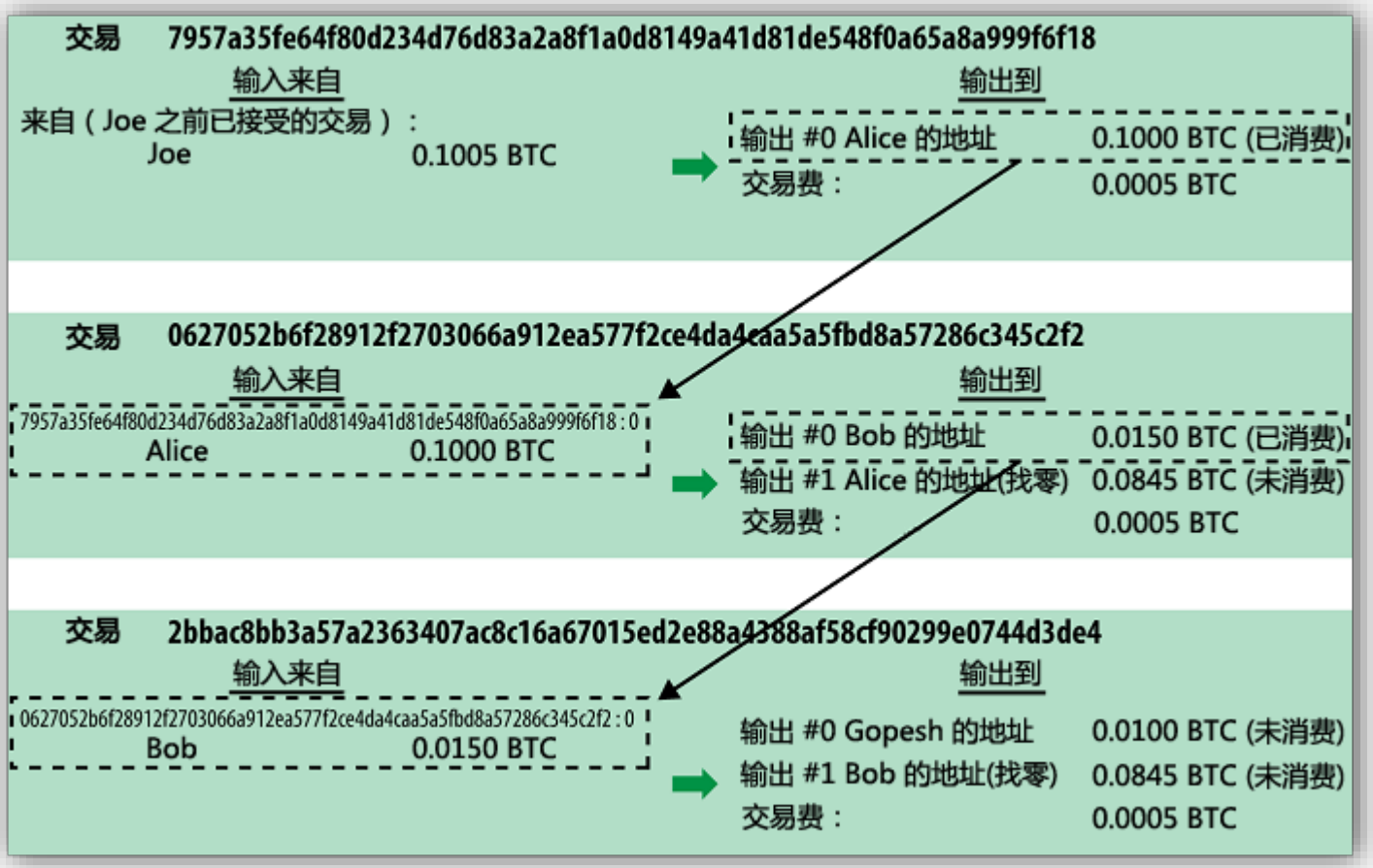
大小	字段	描述
4字节	版本	明确这笔交易参照的规则
1-9字节	输入数量	被包含的输入的数量
不定	**输入**	**一个或多个交易输入**
1-9字节	输出数量	被包含的输出的数量
不定	**输出**	**一个或多个交易输入**
4字节	时间戳或区块号	一个UNIX时间戳或区块号

比特币交易

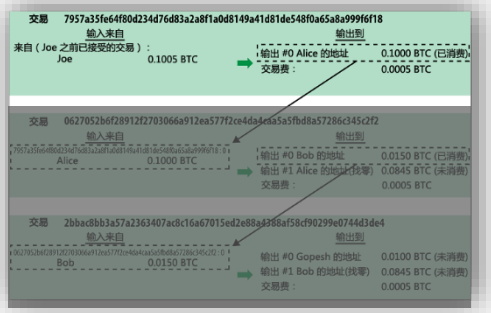
Alice 花 0.015 BTC 购买咖啡

A bitcoin address: "1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA"
The payment amount: "0.015"
A label for the recipient address: "Bob's Cafe"
A description for the payment: "Purchase at Bob's Cafe"

比特币交易


交易 7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18
输入来自
来自（Joe 之前已接受的交易）：
Joe 0.1005 BTC
输出到
输出 #0 Alice 的地址 0.1000 BTC (已消费)
交易费： 0.0005 BTC
交易 0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2
输入来自
7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18 : 0
Alice 0.1000 BTC
输出到
输出 #0 Bob 的地址 0.0150 BTC (已消费)
输出 #1 Alice 的地址(找零) 0.0845 BTC (未消费)
交易费： 0.0005 BTC
交易 2bbac8bb3a57a2363407ac8c16a67015ed2e88a4388af58cf90299e0744d3de4
输入来自
0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2 : 0
Bob 0.0150 BTC
输出到
输出 #0 Gopesh 的地址 0.0100 BTC (未消费)
输出 #1 Bob 的地址(找零) 0.0845 BTC (未消费)
交易费： 0.0005 BTC

构建交易



交易记录 浏览比特币交易的相关信息

7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18

18iWVBbk8tA9bbipS1evviVLP4eE5ga51P (0.1 BTC - 输出) → 1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK - (已使用) 0.1 BTC

0.1 BTC

概览	
大小	225 (字节)
接收时间	2013-12-27 18:25:41
在区块内	277298 (2013-12-27 21:35:27 + 190 时间)
确认	198850 确认
播报方IP地址	Blockchain.info
可视化	浏览树状图

输入与输出	
输入总额	0.1 BTC
输出总额	0.1 BTC
交易费	0 BTC
每字节费用	0 sat/B
预计比特币成交	0.1 BTC
脚本	隐藏交易脚本 & Coinbase信息

输入脚本

3046022100a59e516883459706ac2e6ed6a97ef9788942d3c96a0108f2699fa48d9a5725d1022100f9bb4434943e87901c0c96b5f3af4e7ba7b83e12c69b1edbfe6965f933fcd17d01
04e5a0b4de6c09bd9d3f730ce56ff42657da3a7ec4798c0ace2459fb007236bc3249f70170509ed663da0300023a5de700998bfec49d4da4c66288a58374626c8d 确认

输出脚本

OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG 确认

构建交易

交易记录 浏览比特币交易的相关信息

0627052b6f28912f2703066a912ea577f2ce4da4caa5a5fbd8a57286c345c2f2

1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK (0.1 BTC - 输出)

1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA - (未使用) 0.015 BTC
1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK - (未使用) 0.0845 BTC

0.0995 BTC

概览	
大小	258 (字节)
接收时间	2013-12-27 23:03:05
在区块内	277316 (2013-12-27 23:11:54 + 9 时间)
确认	198832 确认
播报方IP地址	Blockchain.info
可视化	浏览树状图

输入与输出	
输入总额	0.1 BTC
输出总额	0.0995 BTC
交易费	0.0005 BTC
每字节费用	193.798 sat/B
预计比特币成交	0.015 BTC
脚本	隐藏交易脚本 & Coinbase信息

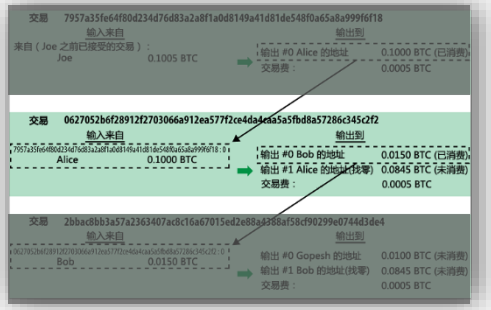
输入脚本

3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f039ff08df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e381301
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d172787ec3457eee41c04f4938de5cc17b4a10fa336a8d752adf 确认

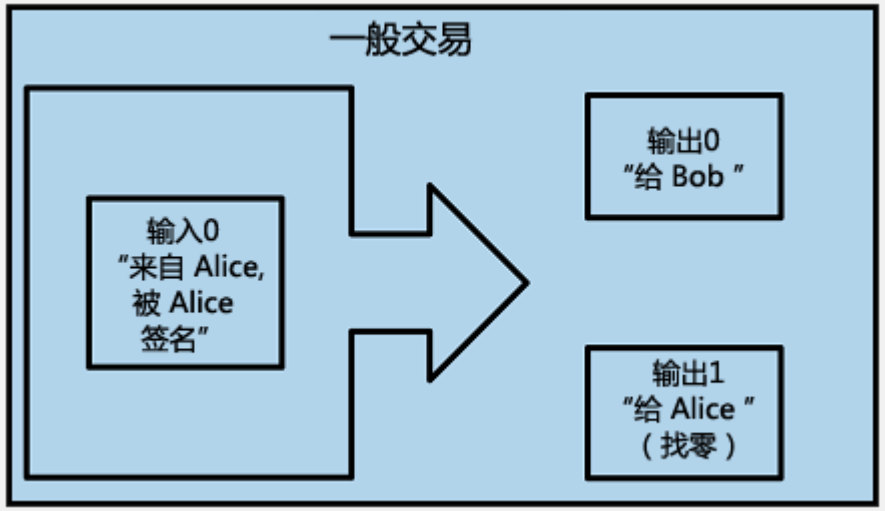
输出脚本

OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG 确认

OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG 确认



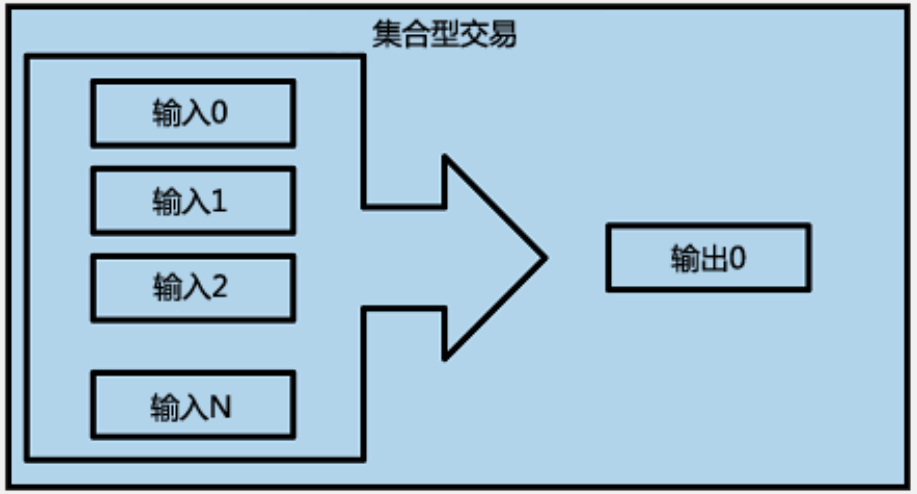
Alice 的交易类型

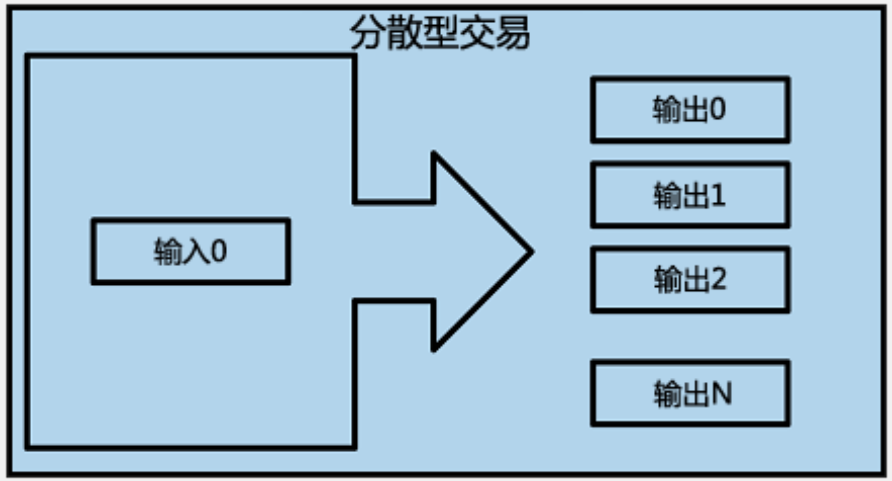


交易中有一个输入，即 Alice 的公钥地址上的一笔钱。

输出 0 是收款方 Bob。
输出1是 Alice 自己，用来实现这笔交易的找零钱。

其他交易类型





比特币交易

“

比特币交易的基本单位是未经使用的一个交易输出，简称 UTXO。当一个用户接收比特币时，金额被当作 UTXO 记录到区块链里。实际上，并不存在储存比特币地址或账户余额的地点，只有被所有者锁住的、分散的 UTXO。

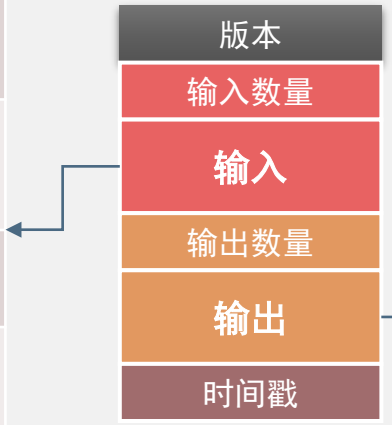
“**一个用户的比特币余额**”这个概念比特币钱包通过扫描区块链并聚合所有属于该用户的 UTXO 来计算该用户的余额。

”

交易的输入和输出

大小	字段	描述
32 字节	交易	指向交易包含的被花费的 UTXO 的哈希指针
4 字节	输出索引	被花费的 UTXO 的索引号，第一个是0
1–9 字节	解锁脚本长度	用字节表示的后面的解锁脚本长度
变长	**解锁脚本**	**一个达到 UTXO 锁定脚本中的条件的脚本**
4 字节	序列号	目前未被使用的交易替换功能，设成 0xFFFFFFFF

交易输入的结构



交易结构

大小	字段	描述
8 字节	总量	用聪表示的比特币值
1–9 字节	锁定脚本长度	用字节表示的后面的锁定脚本长度
变长	**锁定脚本**	**一个定义了支付输出所需条件的脚本**

交易输出的结构

```
{
   "txid":"9ca8f969bd3ef5ec2a8685660fdbf7a8bd365524c2e1fc66c309acbae2c14ae3",
   "version" : 1,
   "locktime" : 0,
   "vin" : [
      {
         "txid":"d3c7e022ea80c4808e64dd0a1dba009f3eaee2318a4ece562f8ef815952717d7",
         "vout" : 0,
         "scriptSig" : {
            "asm" :
"3045022100a4ebbeec83225dedead659bbde7da3d026c8b8e12e61a2df0dd0758e227383b302203301768ef878007e9ef7c304f70ffaf1f2c975b192d34c5b9b2ac1bd193dfba20104793ac8a58ea751f9710e39aad2e296cc14daa44fa59248be58ede65e4c4b↵884ac5b5b6dede05ba84727e34c8fd3ee1d6929d7a44b6e111d41cc79e05dbfe5cea",
            "hex":
"483045022100a4ebbeec83225dedead659bbde7da3d026c8b8e12e61a2df0dd0758e227383b302203301768ef878007e9ef7c304f70ffaf1f2c975b192d34c5b9b2ac1bd193dfba2014104793ac8a58ea751f9710e39aad2e296cc14daa44fa59248be58ede65e4c4b884ac5b5b6dede05ba84727e34c8fd3ee1d6929d7a44b6e111d41cc79e05dbfe5cea"
         },
         "sequence" : 4294967295
      }
   ],
   "vout" : [
      {
         "value" : 0.05000000,
         "n" : 0,
         "scriptPubKey" : {
            "asm" : "OP_DUP OP_HASH160 07bdb518fa2e6089fd810235cf1100c9c13d1fd2 OP_EQUALVERIFY OP_CHECKSIG",
            "hex" :"76a91407bdb518fa2e6089fd810235cf1100c9c13d1fd288ac",
            "reqSigs" : 1,
            "type" : "pubkeyhash",
            "addresses" : [
               "1hvzSofGwT8cjb8JU7nBsCSfEVQX5u9CL"
            ]
         }
      },
      {
         "value" : 1.03362847,
         "n" : 1,
         "scriptPubKey" : {
            "asm" : "OP_DUP OP_HASH160 107b7086b31518935c8d28703d66d09b36231343 OP_EQUALVERIFY OP_CHECKSIG",
            "hex" : "76a914107b7086b31518935c8d28703d66d09b3623134388ac",
            "reqSigs" : 1,
            "type" : "pubkeyhash",
            "addresses" : [
               "12W9goQ3P7Waw5JH8fRVs1e2rVAKoGnvoy"
            ]
         }
      }
   ]
}
```

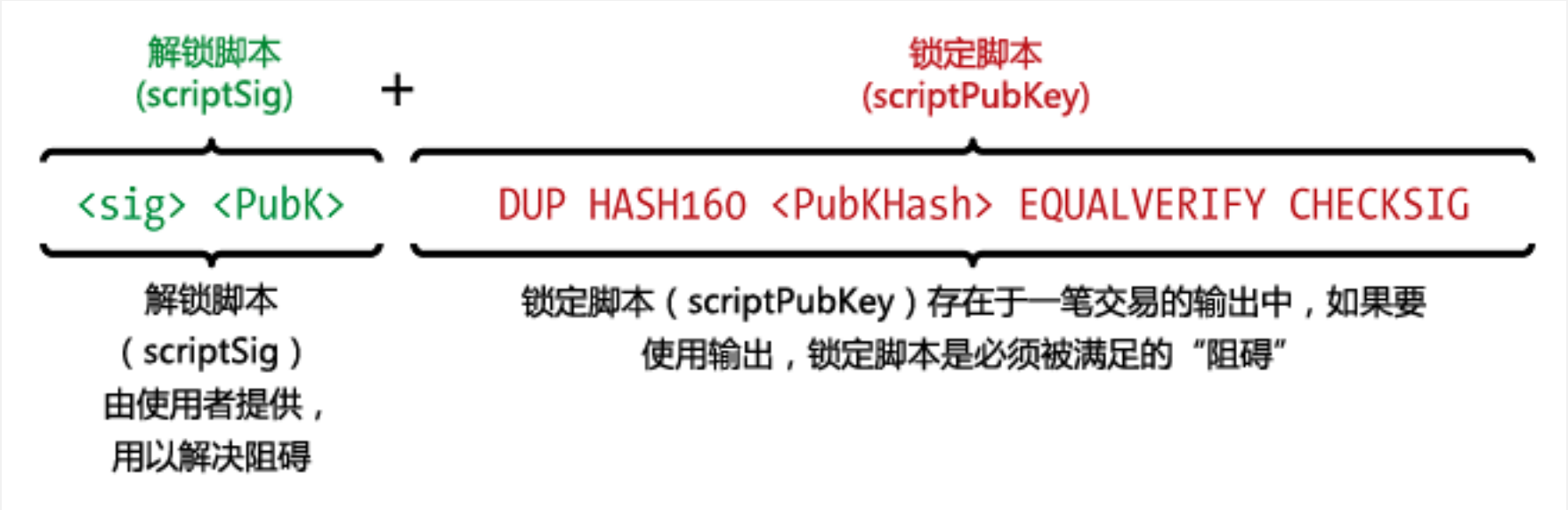

交易发起者自己公钥的签名和公钥

锁定脚本指定了收款人的地址哈希

锁定脚本指定了自己的公钥哈希

比特币交易脚本

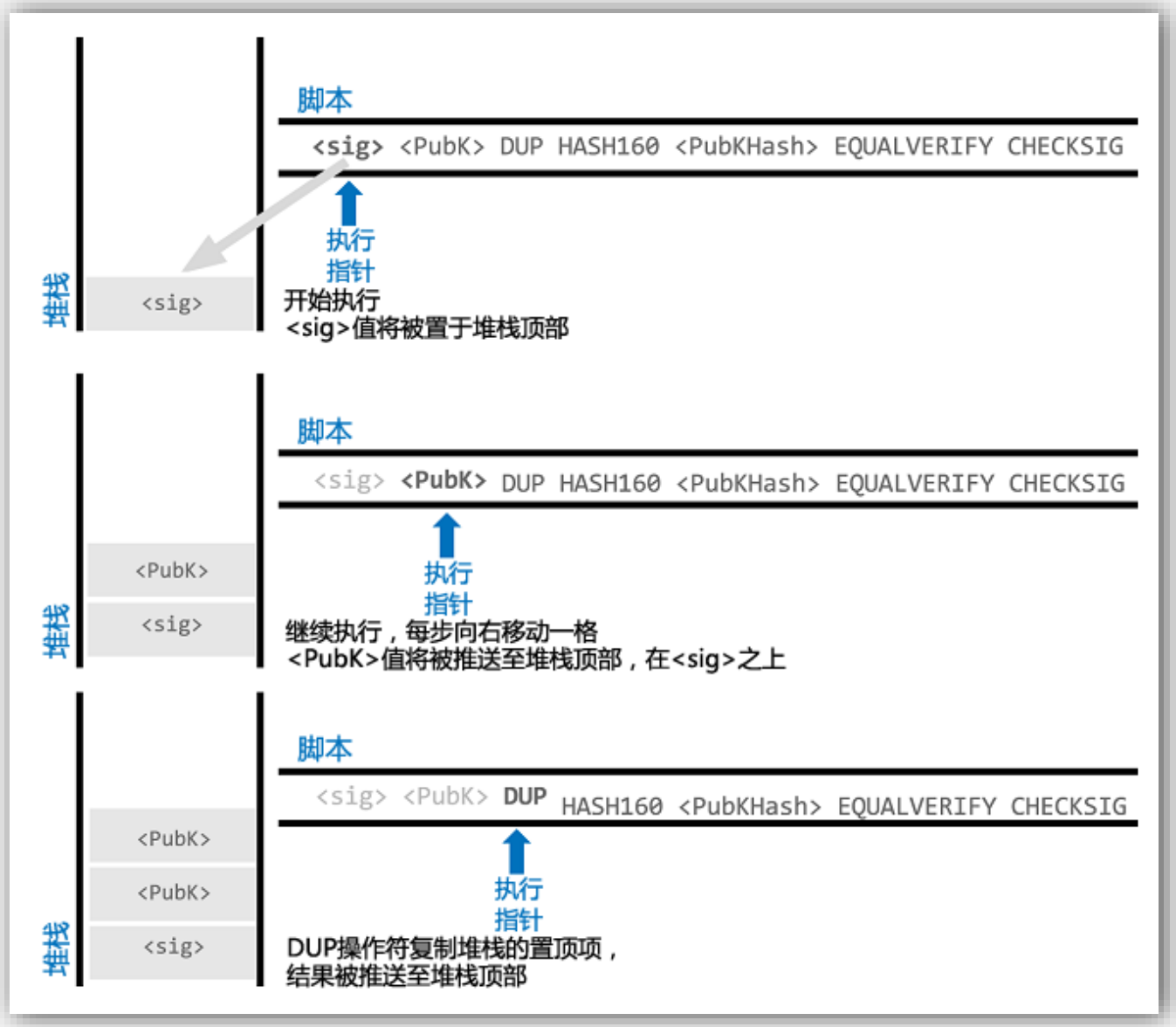
比特币的交易验证引擎依赖于两类脚本来验证比特币交易：一个锁定脚本和一个解锁脚本。



最常见的 P2PKH（Pay-to-Public-Key-Hash）脚本

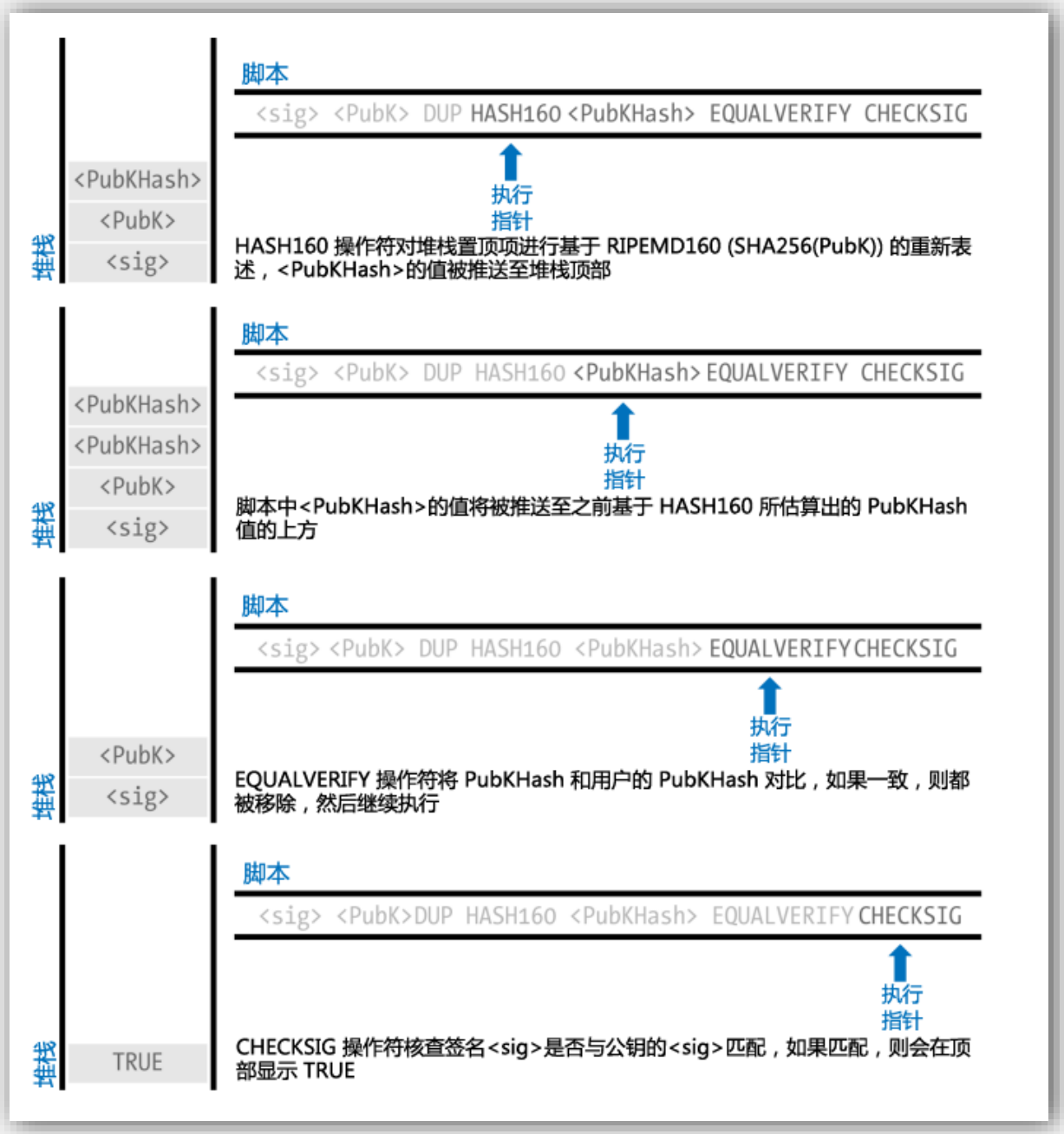
P2PKH 脚本

只有当解锁版脚本与锁定版脚本的设定条件相匹配时，执行组合有效脚本时才会显示结果为真



P2PKH 脚本

只有当解锁版脚本与锁定版脚本的设定条件相匹配时，执行组合有效脚本时才会显示结果为真



一种特殊的交易 Coinbase

“

Coinbase 创币交易——每个区块中的首个交易。交易中的比特币作为挖矿的奖励支付给“赢家”矿工。**创币交易没有输入**，不消耗 UTXO。它只包含一个被称作 coinbase 的输入，仅仅用来创建新的比特币。创币交易有一个输出，支付到这个矿工的比特币地址。

”

比特币工作原理

1 比特币钱包

2 比特币交易

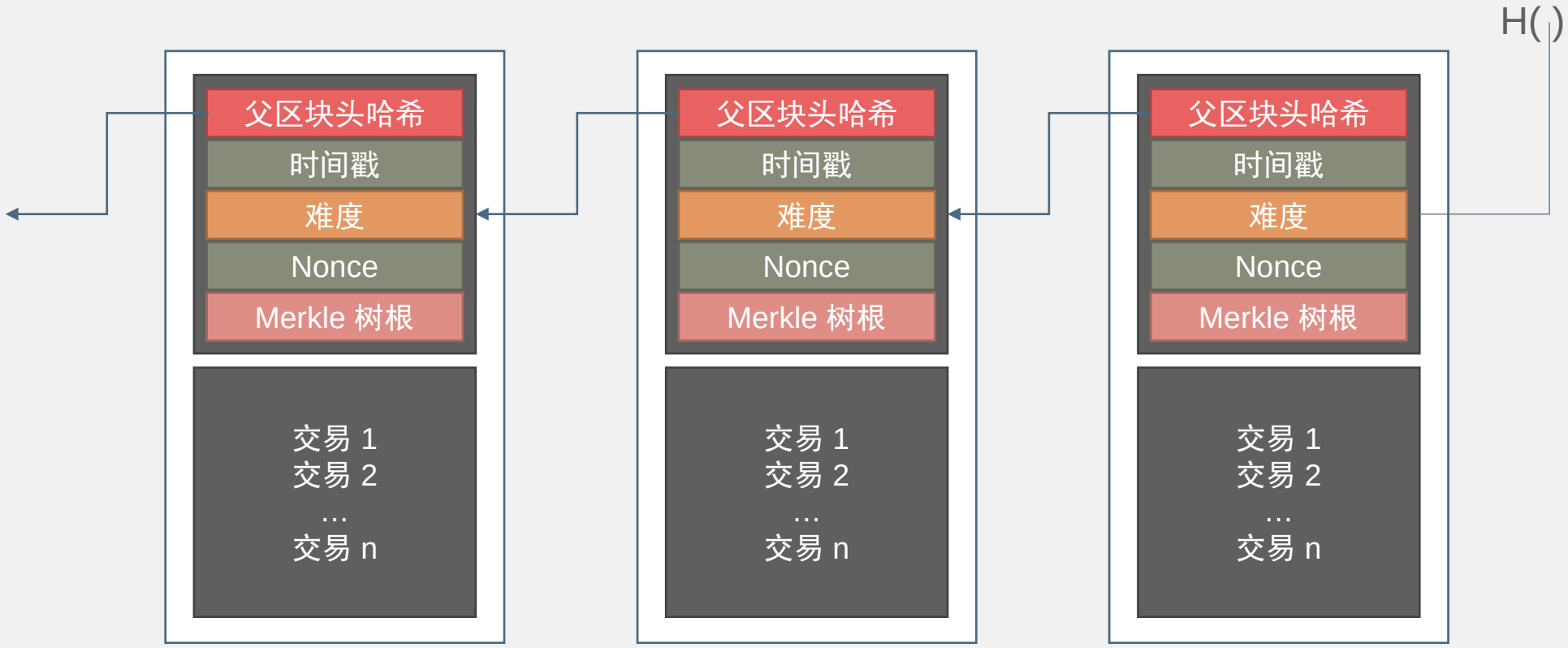
3 区块链

4 挖矿共识

区块链

比特币的账本是由包含交易信息的区块从后向前有序链接起来的块链式数据结构。

区块链



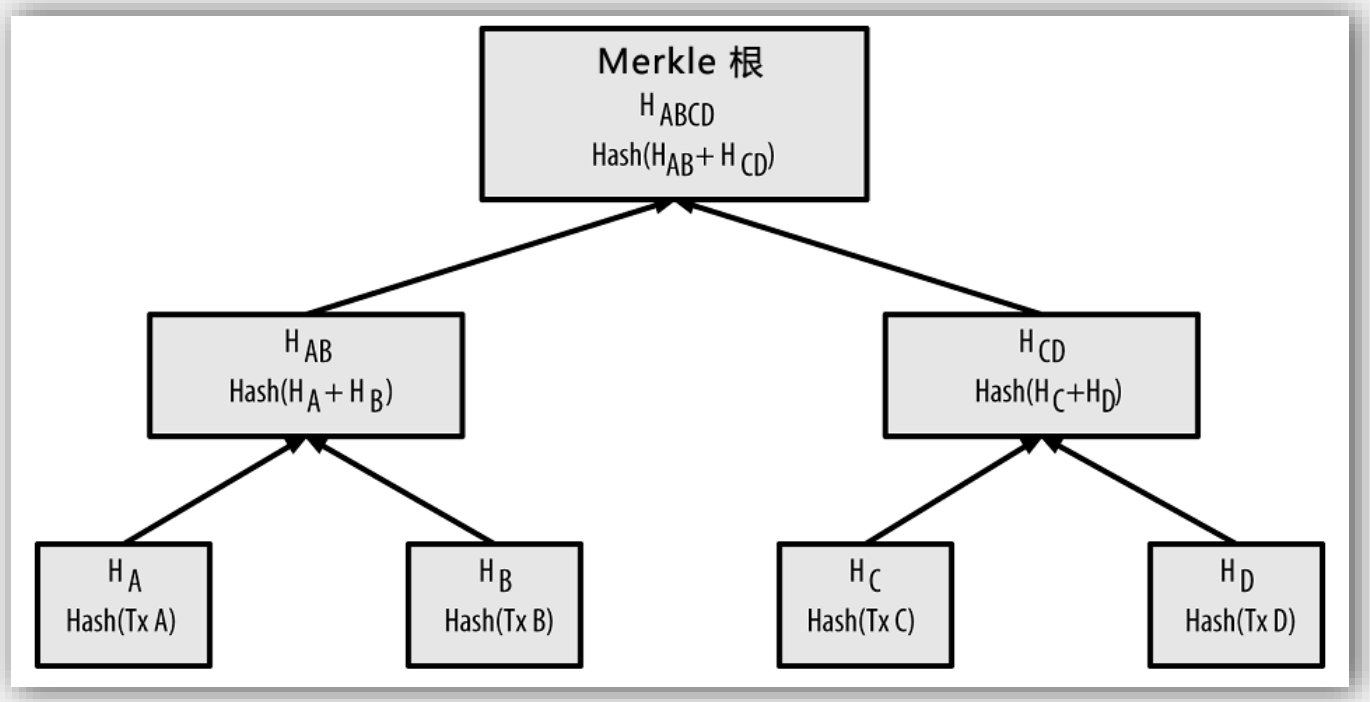
区块通过引用父区块的区块头哈希值的方式，以链条的形式进行相连

Merkle 树

Merkle 树是一种哈希二叉树，它是一种用作快速归纳和校验大规模数据完整性的数据结构。

H_A = SHA256(SHA256(Tx A))

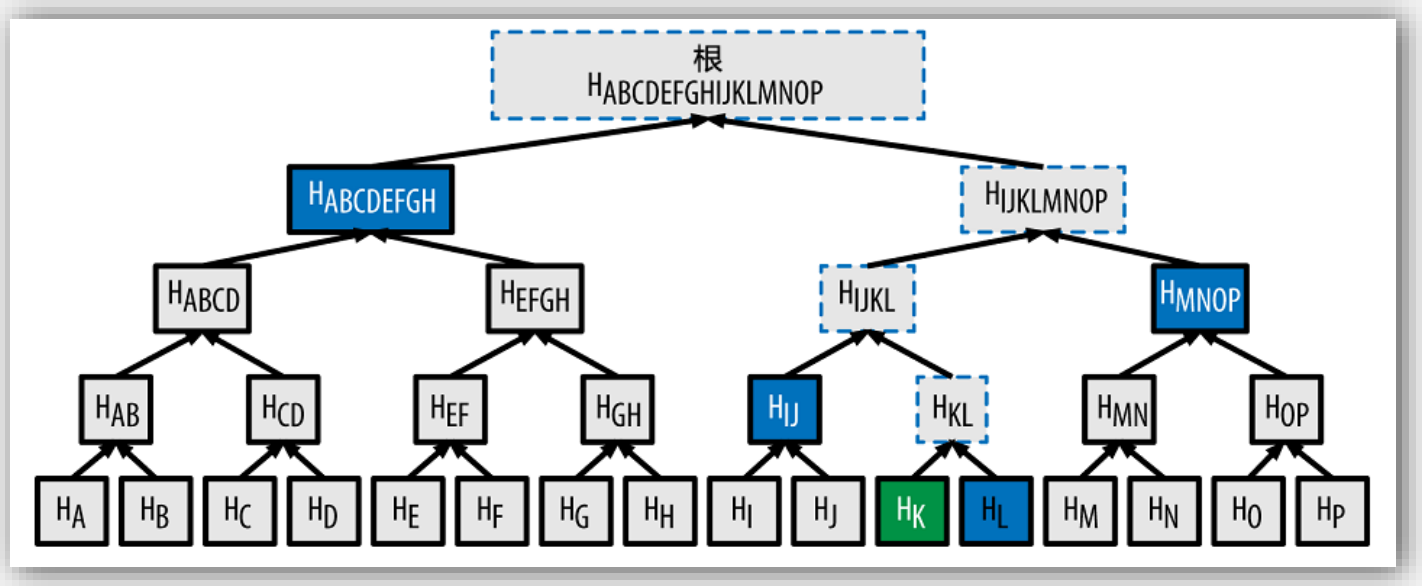
H_{AB}= SHA256(SHA256(H_A + H_B))



Merkle 树

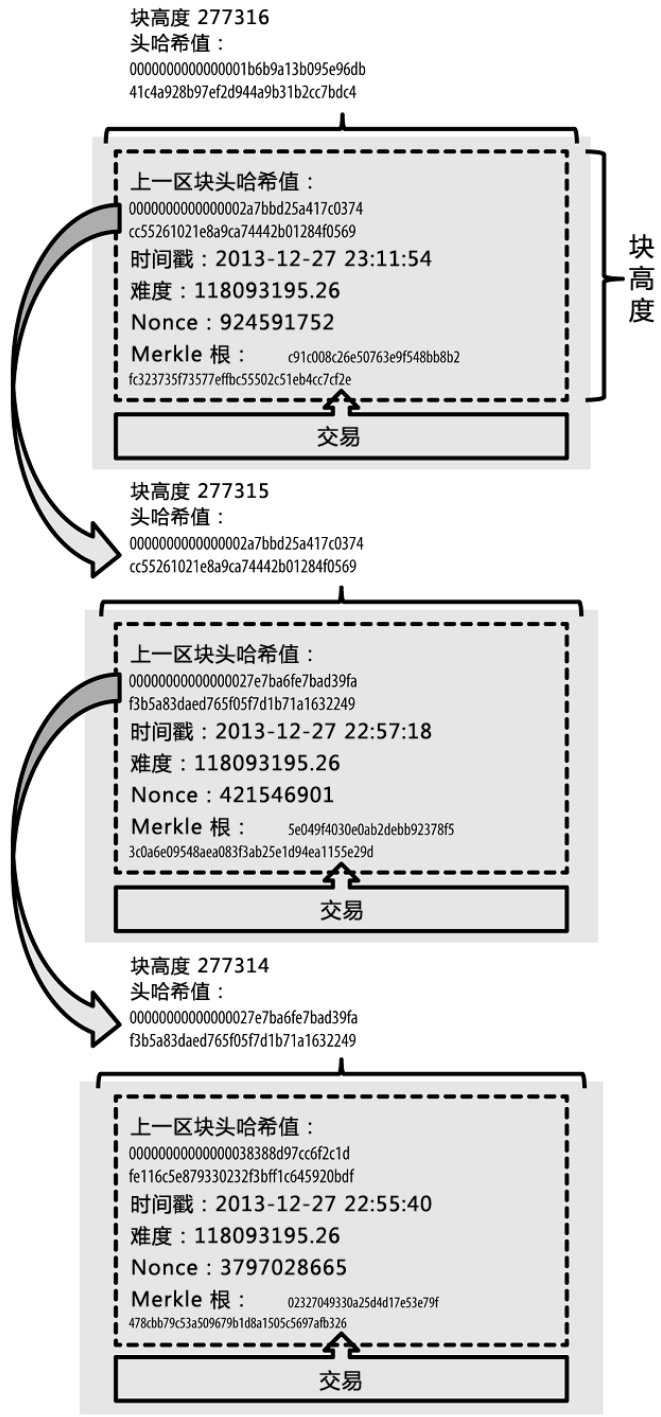
为了证明区块中存在某个特定的交易，一个节点只需要计算 $\log_2(N)$ 个 32 字节的哈希值，形成一条从特定交易到树根的认证路径或者 Merkle 路径即可。

由这 4 个哈希值产生的认证路径，再通过计算另外四对哈希值 H_{KL}、H_{IJKL}、$H_{IJKLMNOP}$ 和 Merkle 树根，任何节点都能证明 H_K 包含在 Merkle 根中。



区块链

把区块链想象成地质构造中的地质层，越往深处，地质层就变得越稳定。
区块链里，最新的六个区块就像几英寸深的表土层。
超过这六块后，区块在区块链中的位置越深，被改变的可能性就越小。
在100个区块以后，区块链已经足够稳定。
几千个区块（一个月）后的区块链将变成确定的历史，永远不会改变。



比特币工作原理

1 比特币钱包

2 比特币交易

3 区块链

4 挖矿共识

挖矿共识

竞争记账和激励机制

在比特币系统中，记账节点（矿工）争相完成一种**基于加密哈希算法的数学难题**，这些难题的答案包括在新区块中，作为矿工的计算工作量的证明，被称为“工作量证明”。通过对解决难题的竞争，获胜者有权在区块链上创建交易的区块，从而获得一定数量的比特币奖励。大约每 10 分钟就会有一个新的区块被“挖掘”出来，每个区块里包含着从上一个区块产生到目前这段时间内发生的所有交易。

挖矿共识

数学难题？

```
I am Satoshi Nakamoto0 => a80a81401765c8eddee25df36728d732...
I am Satoshi Nakamoto1 => f7bc9a6304a4647bb41241a677b5345f...
I am Satoshi Nakamoto2 => ea758a8134b115298a1583ffb80ae629...
I am Satoshi Nakamoto3 => bfa9779618ff072c903d773de30c99bd...
I am Satoshi Nakamoto4 => bce8564de9a83c18c31944a66bde992f...
I am Satoshi Nakamoto5 => eb362c3cf3479be0a97a20163589038e...
I am Satoshi Nakamoto6 => 4a2fd48e3be420d0d28e202360cfbaba...
I am Satoshi Nakamoto7 => 790b5a1349a5f2b909bf74d0d166b17a...
I am Satoshi Nakamoto8 => 702c45e5b15aa54b625d68dd947f1597...
I am Satoshi Nakamoto9 => 7007cf7dd40f5e933cd89fff5b791ff0...
I am Satoshi Nakamoto10 => c2f38c81992f4614206a21537bd634a...
I am Satoshi Nakamoto11 => 7045da6ed8a914690f087690e1e8d66...
I am Satoshi Nakamoto12 => 60f01db30c1a0d4cbce2b4b22e88b9b...
I am Satoshi Nakamoto13 => 0ebc56d59a34f5082aaef3d66b37a66...
I am Satoshi Nakamoto14 => 27ead1ca85da66981fd9da01a8c6816...
I am Satoshi Nakamoto15 => 394809fb809c5f83ce97ab554a2812c...
I am Satoshi Nakamoto16 => 8fa4992219df33f50834465d3047429...
I am Satoshi Nakamoto17 => dca9b8b4f8d8e1521fa4eaa46f4f0cd...
I am Satoshi Nakamoto18 => 9989a401b2a3a318b01e9ca9a22b0f3...
I am Satoshi Nakamoto19 => cda56022ecb5b67b2bc93a2d764e75f...
```

用概率的角度来看，我们可以期望每 16 次得到一个以 0 开头的哈希值。从数字的角度来看，我们要找的是小于 0x1000000000000000000000000000000000000000000000000000000000000000 的哈希值。我们称这个为目标阀值，我们的目的是找到一个小于这个目标的哈希值。如果我们减小这个目标值，那找到一个小于它的哈希值会越来越难。

谁能先算出比前 N 位都是 0 的哈希值还小的数谁就赢了！

挖矿共识

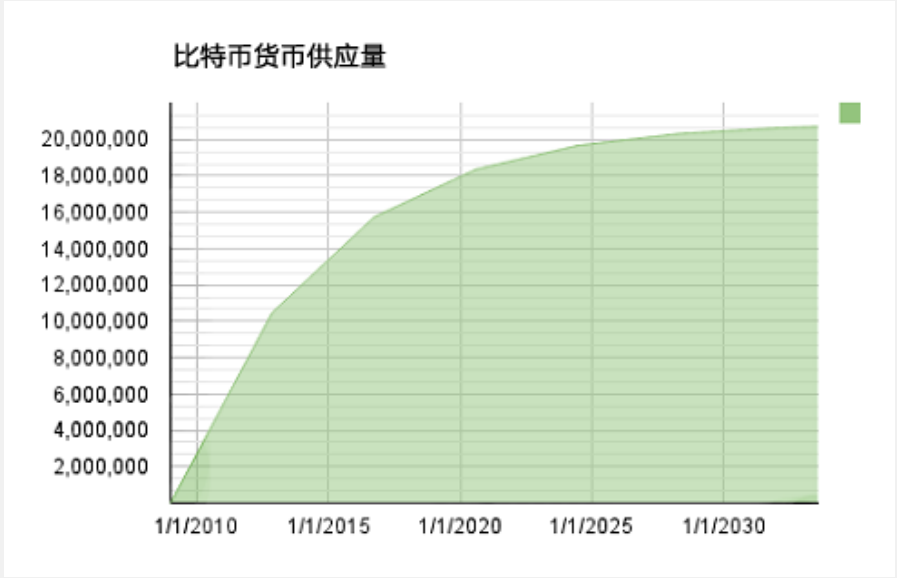
挖矿过程

1. 矿工根据当前区块链末端的数据块计算新数据块的头部 prev_hash；
2. 矿工生成随机数 nonce，并将新数据块的头部 prev_hash、收集到的交易的 merkle 树根、随机数 nonce 作为哈希函数H( )的输入，计算H(prev_hash || merkle || nonce)，若 H( ) 的函数值小于某一个预设的阈值，则 nonce 合法，否则重新生成 nonce，继续计算；
3. 矿工在找到合法的 nonce 后迅速进行 p2p 广播，其他矿工在收到该消息后停止挖矿并进行验证，验证通过后，认为新的区块已产生（达成共识）；
4. 矿工基于新产生的区块继续挖矿。

难度示例：0x0000000000000003A30C0000000000000000000000000000000000000000000
算力 1 THash/s 的矿工平均每 59 天才能挖到一个区块

挖矿共识

比特币的奖励机制被设计为速度递减模式，类似于贵重金属的挖矿过程，因此被形象地比喻成 **挖矿**



矿工通过创造一个新区块得到的比特币数量每 210,000 个块减少一半。2009 年 1 月每个区块奖励 50 个比特币，2012 年 11 月减半为每个区块奖励 25 个比特币，2016 年 7 月再次减半为每个新区块奖励 12.5 个比特币。到 2140 年，所有的比特币（20,999,999.9769）全部发行完毕。

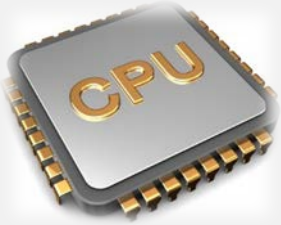
挖矿共识

挖矿的奖励

矿工们在挖矿过程中会得到两种类型的奖励：创建新区块的新币奖励，以及区块中所含交易的交易费。

矿工们的“工具”

CPU



20MHash/s

GPU

400MHash/s

FPGA

25GHash/s

ASIC

3.5THash/s +

挖矿共识

挖矿既是比特币发行的过程，也是支撑比特币安全的去中心化的共识机制。

挖矿共识

拜占庭将军问题

Leslie Lamport 1982 年提出用来解释一致性问题的一个虚构模型。拜占庭是古代东罗马帝国的首都，由于地域宽广，守卫边境的多个将军（系统中的多个节点）需要通过信使来传递消息，达成某些一致的决定。但由于将军中可能存在叛徒（系统中节点出错），这些叛徒将努力向不同的将军发送不同的消息，试图会干扰一致性的达成。

拜占庭问题即为在此情况下，如何让忠诚的将军们能达成行动的一致。

挖矿共识

“

比特币账本的同步更新也面临同样的问题。系统采用工作量证明机制，限制一段时间内整个网络中出现提案的个数，同时放宽对最终一致性确认的需求，约定好大家都确认并沿着已知最长的链进行拓宽。

”

挖矿共识

矿工守则

1.每个全节点依据综合标准对每个交易进行**独立验证**

2.通过完成工作量证明算法的验算，挖矿节点将交易记录**独立打包进新区块**

3.每个节点**独立**的对新区块进行**校验并组装**进区块链

4.每个节点对区块链进行**独立选择**，在工作量证明机制下选择**累计工作量最大的区块链**

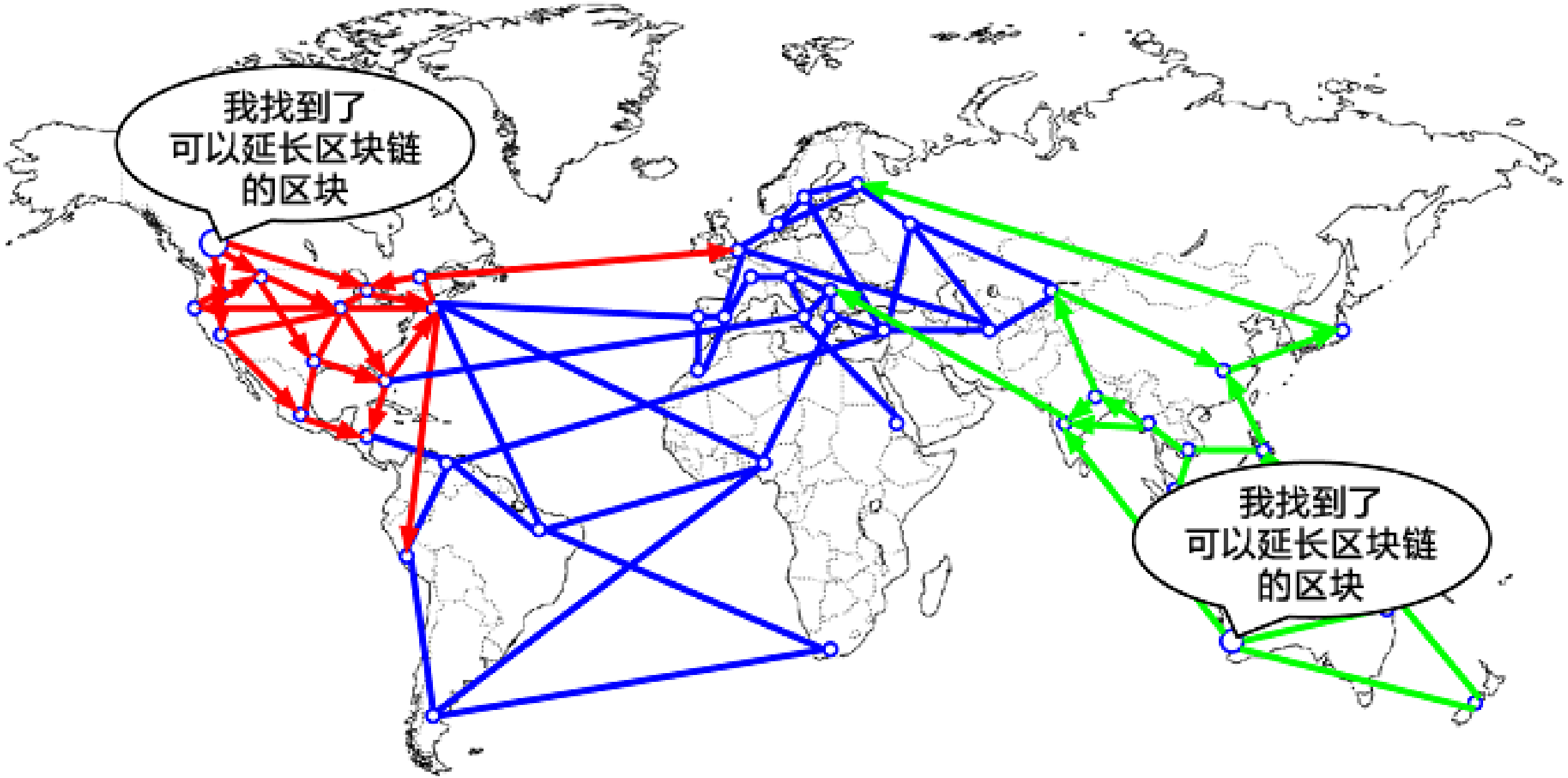
区块链分叉

“

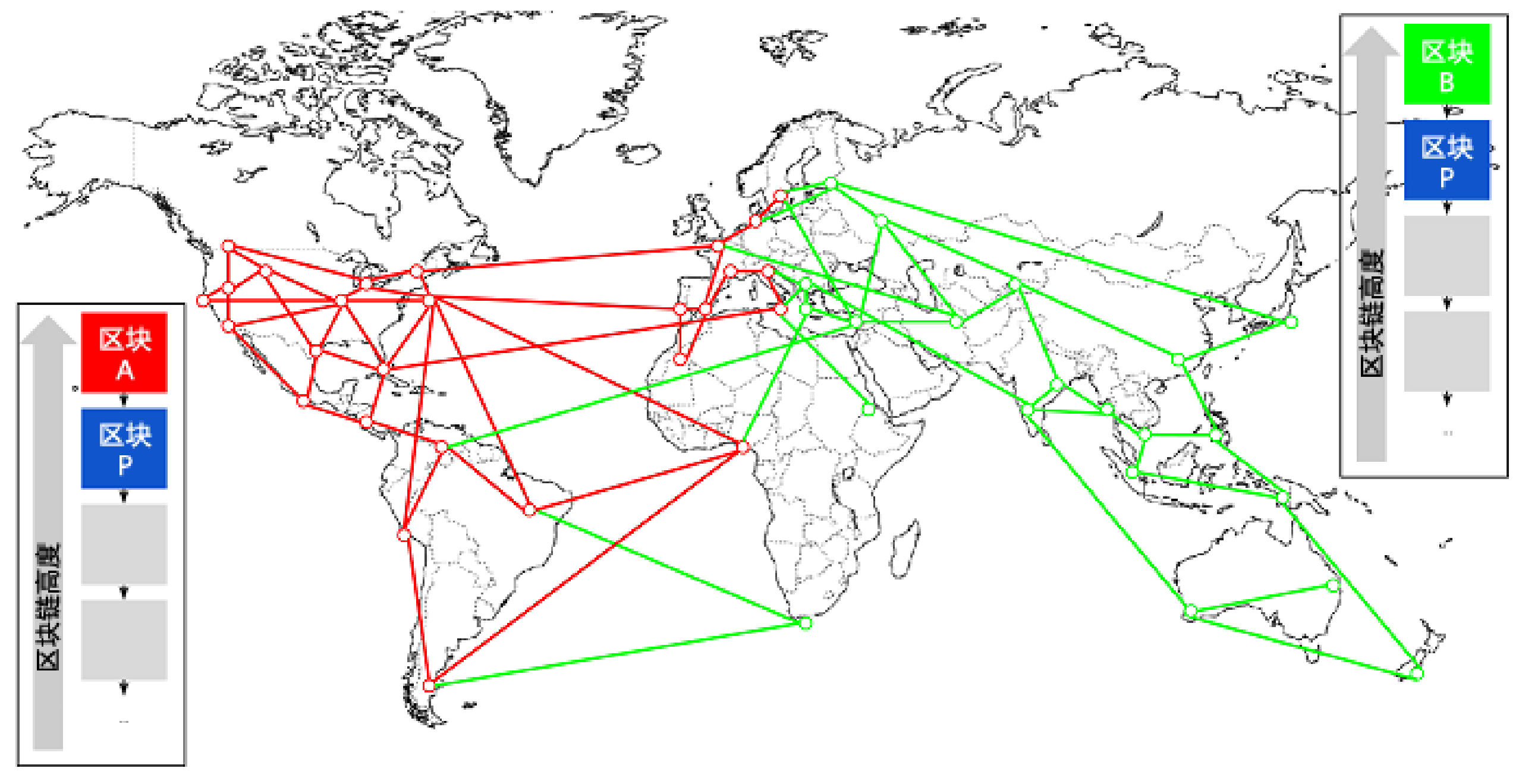
在无中心化场景中，不同节点之间的账本不能总是保持一致。区块有可能在不同时间到达不同节点，导致节点有不同的区块链视角。解决的办法是，每一个节点总是选择并尝试延长代表累计了最大工作量证明的区块链，也就是最长的或最大累计难度的链。

”

形象化的区块链分叉事件——分叉之前

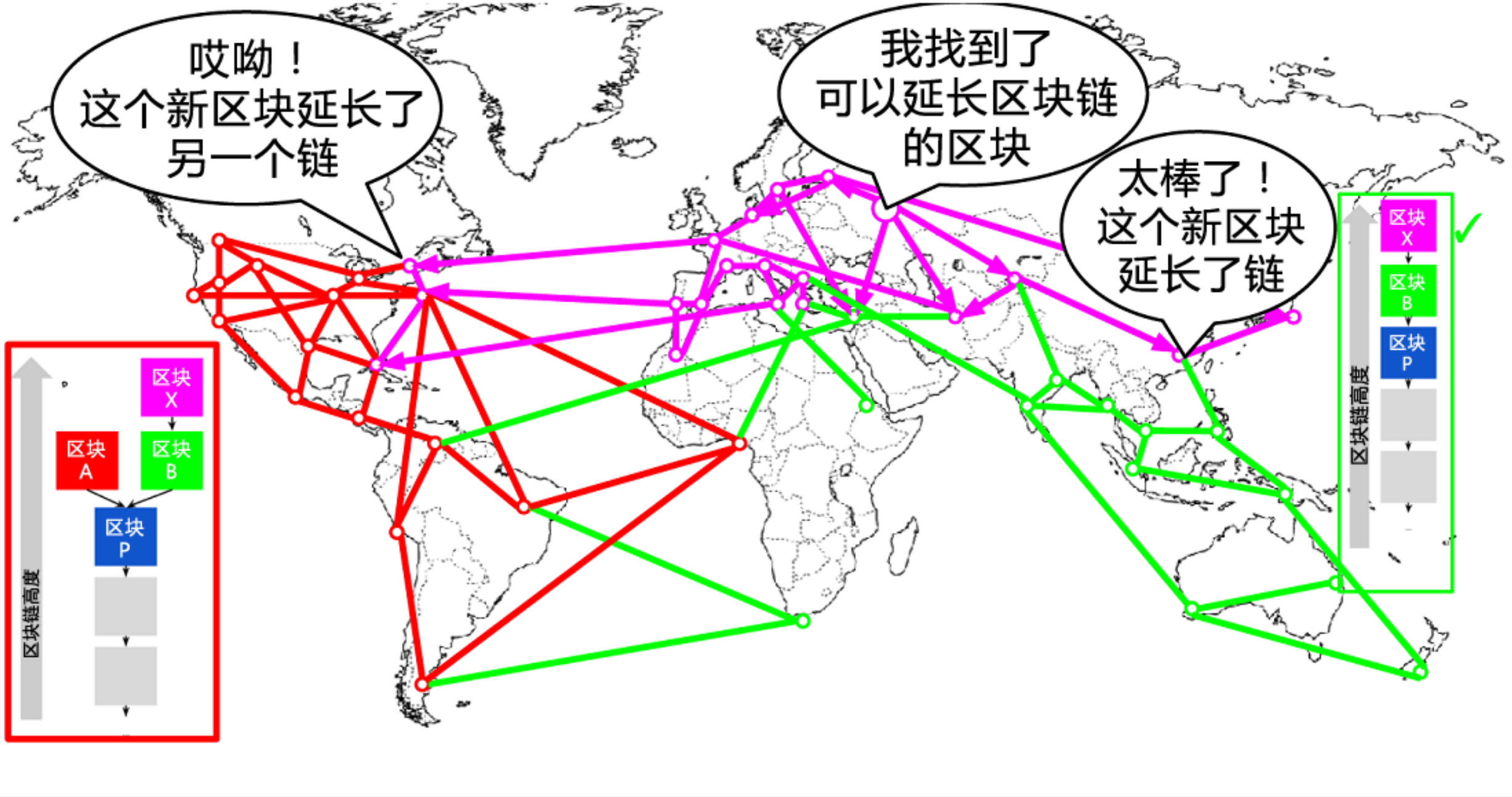


形象化的区块链分叉事件：同时发现两个区块



形象化的区块链分叉事件：两个区块的传播将网络分裂了

形象化的区块链分叉事件：新区块延长了分支



形象化的区块链分叉事件：全网在最长链上重新共识

目录

CONTENTS

04

比特币安全性分析

偷取比特币、篡改区块、拒绝服务、共识攻击

偷取比特币？

只要私钥不泄露，别人永远不可能从你的钱包地址中偷到比特币。

自行修改区块奖励？

根据矿工守则，其他诚实矿工是不会接受这样的区块的。

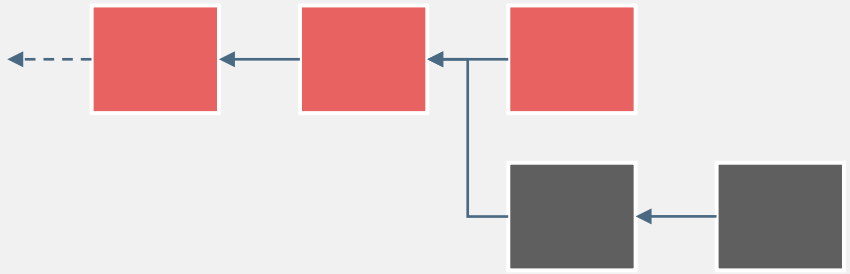
“拒绝服务”攻击

假设 Evil 是一个矿工，而 Bob 是他的死对头。Evil 在构造区块的时候，拒绝接受所有 Bob 相关的交易。能不能完全封锁 Bob 的比特币交易呢？

“Evil 阻止不了其他节点接收 Bob 的交易，也阻止不了其他节点接受记录了 Bob 交易的区块。**除非？**”

51% 共识攻击

想象这么一个场景，一群矿工控制了整个比特币网络 51% 的算力，他们联合起来打算攻击整个比特币系统。由于这群矿工可以生成绝大多数的块，他们就可以通过故意制造块链分叉来实现“双重支付”或者通过拒绝服务的方式来阻止特定的交易。算力充足情况下，攻击者可以一次性篡改最近的 6 个或者更多的区块，从而使得这些区块本应该包含的交易消失。



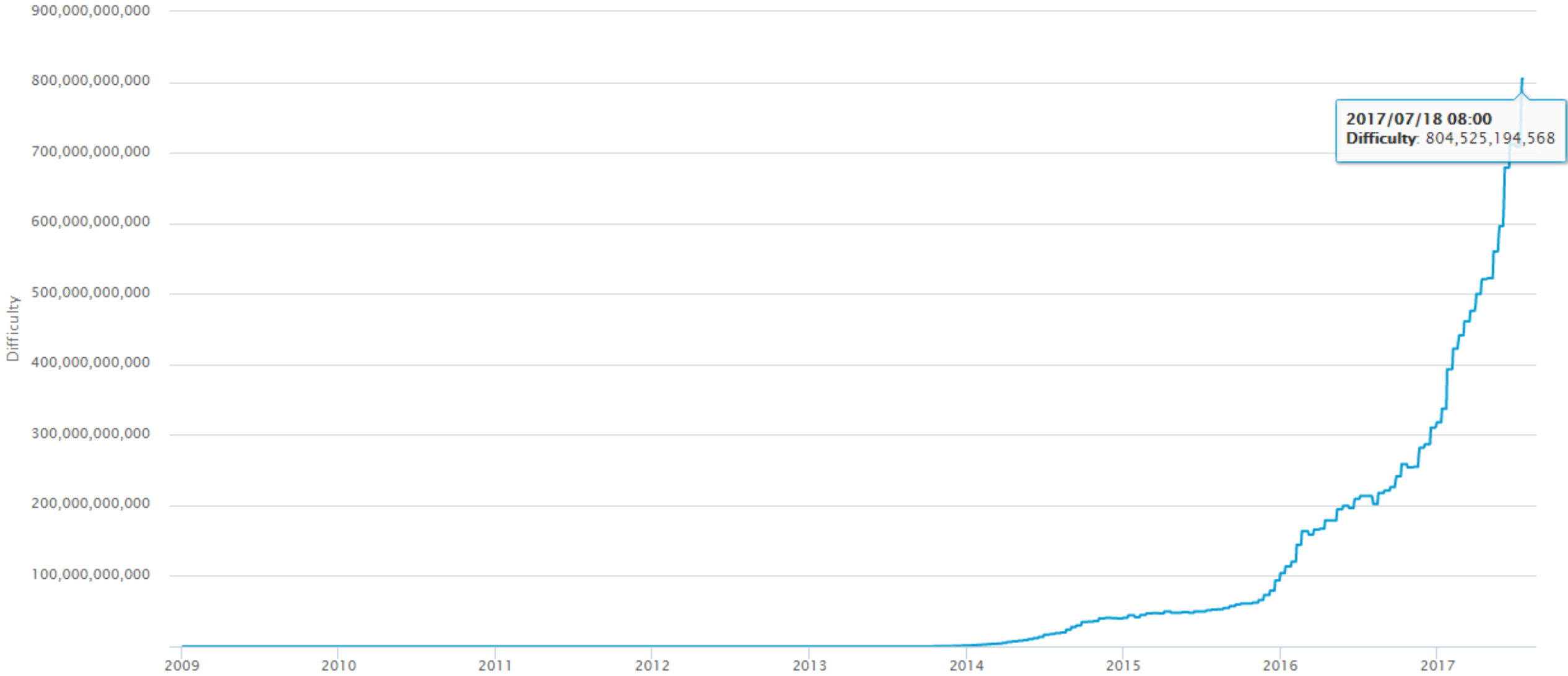
51% 共识攻击

限制条件：双重支付只能在攻击者拥有的钱包所发生的交易上进行，因为只有钱包的拥有者才能生成一个合法的签名用于双重支付交易。攻击者只能在自己的交易上进行双重支付攻击，但当这笔交易对应的是不可逆转的购买行为的时候，这种攻击就是有利可图的。

Difficulty

A relative measure of how difficult it is to find a new block. The difficulty is adjusted periodically as a function of how much hashing power has been deployed by the network of miners.

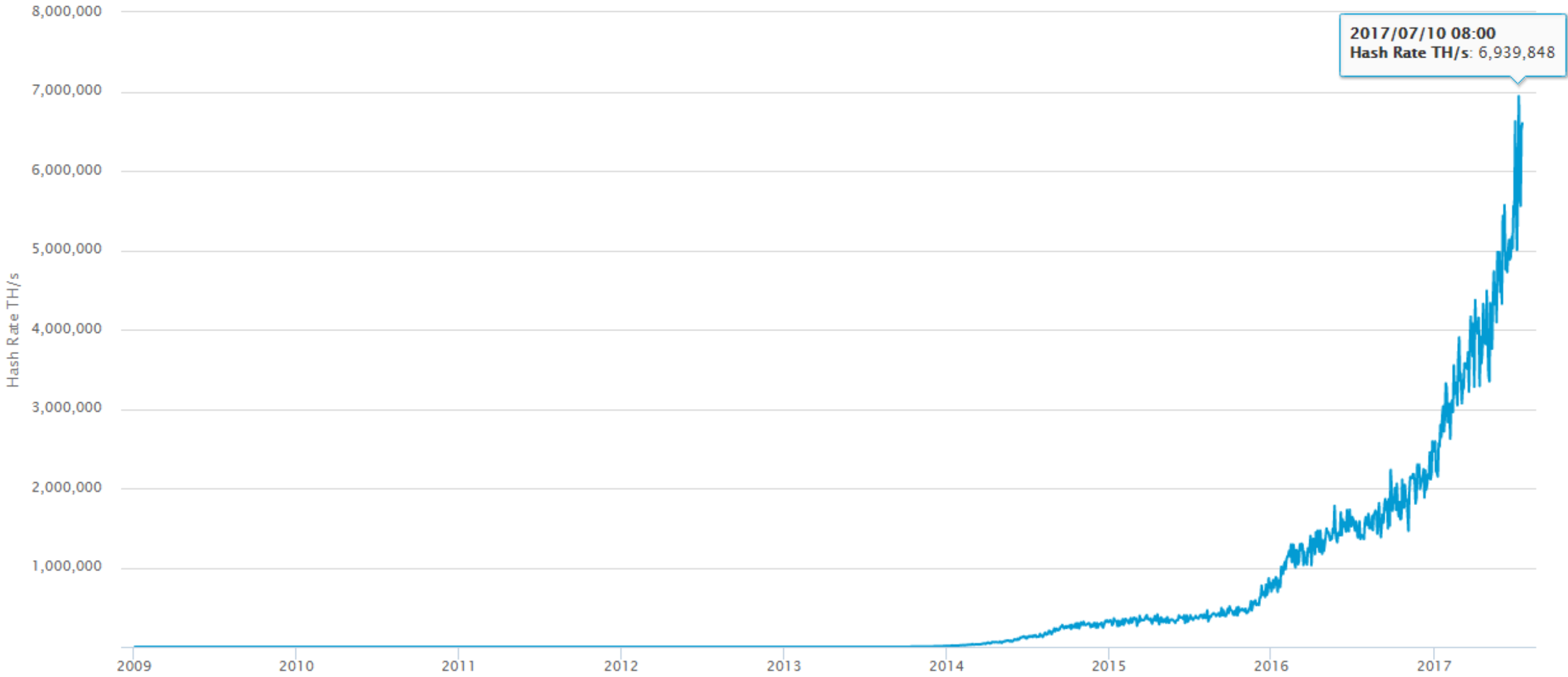
Source: blockchain.info



Hash Rate

The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing.

Source: blockchain.info



参考链接

- 精通比特币 http://zhibimo.com/books/wang-miao/mastering-bitcoin
- 区块链技术指南 https://yeasy.gitbooks.io/blockchain_guide/
- BLOCKCHAIN https://blockchain.info/
- 更多内容 https://github.com/gymgle/blockchain-reference

关于我

https://github.com/gymgle

@Gymgle

https://g2ex.github.io

ymgongcn#gmail.com

Thanks