VICTORLAMP

区块链技术核心篇
之三：比特币区块链交易共识

e休，爱编程的葫芦娃：exiu@victorlamp.com

煤油灯科技公司 http://www.victorlamp.com

多媒体课程：《深入浅出区块链技术核心篇》

深入浅出区块链技术

课程介绍

《区块链技术基础篇之一：白话非对称加解密》
《区块链技术基础篇之二：白话哈希算法》
《区块链技术基础篇之三：白话P2P网络》
《区块链技术基础篇之四：白话拜占庭将军问题》
《区块链技术核心篇之一：比特币区块链起源及原理》
《区块链技术核心篇之二：比特币区块链密钥与地址》
《区块链技术核心篇之三：比特币区块链交易共识》
《区块链技术核心篇之四：比特币区块链核心架构》

讲师介绍

网名：一休

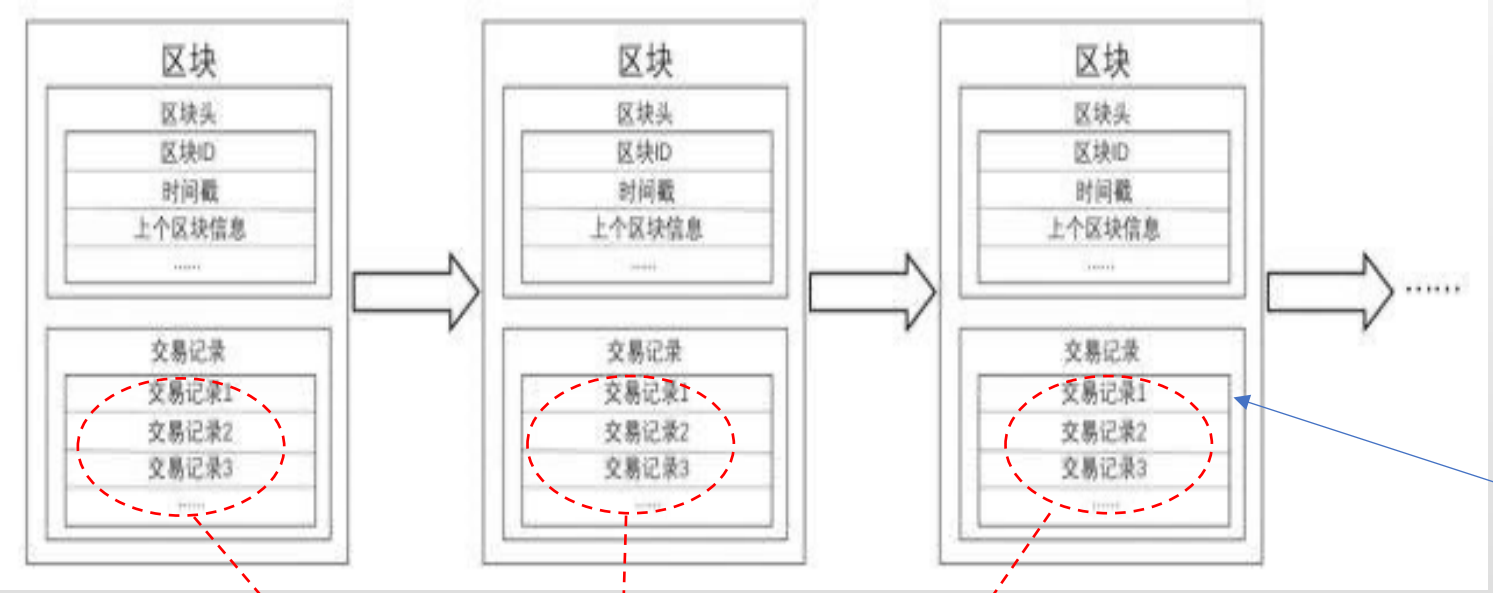
2015 年开始，从事区块链技术开发，先后成功的研发出：区块链技术的数字版权管理（DRM）系统、基于区块链IPFS的CDN产品开发。

是多项区块链技术专利发明人。

对于比特币区块链、IPFS项目源码非常熟悉。

本人负责的基于区块链的创新技术方案，获得了当年华为公司年度十大发明奖。

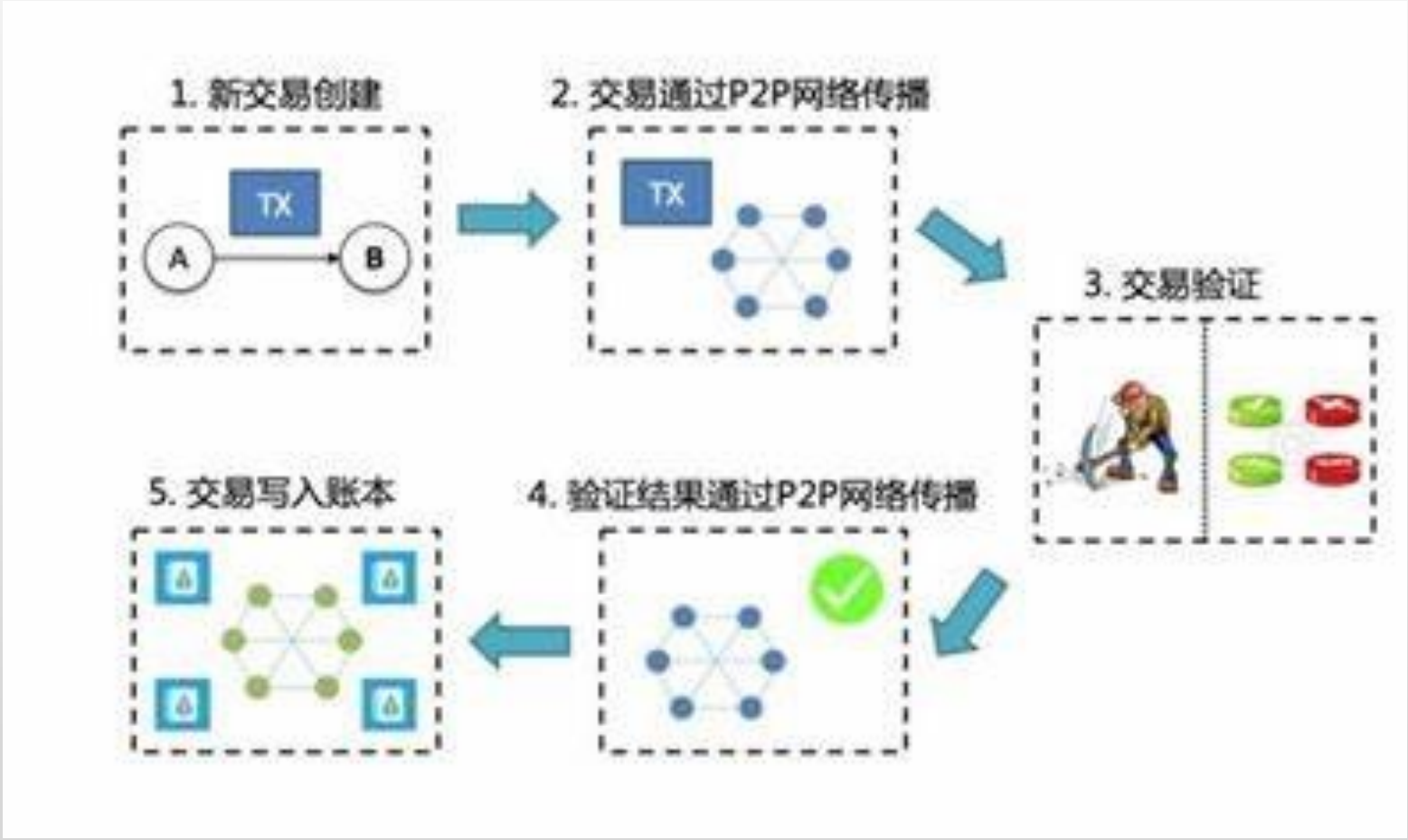
什么是区块链交易?



区块链的每个区块的Body部分就是一条条的交易组成的集合。

我们的每一次购物、转账都会产生一条交易纪录。

比特币区块链运行过程：就是不断的打包交易，挖矿，构建新的区块的过程



每条交易包含哪些东西?

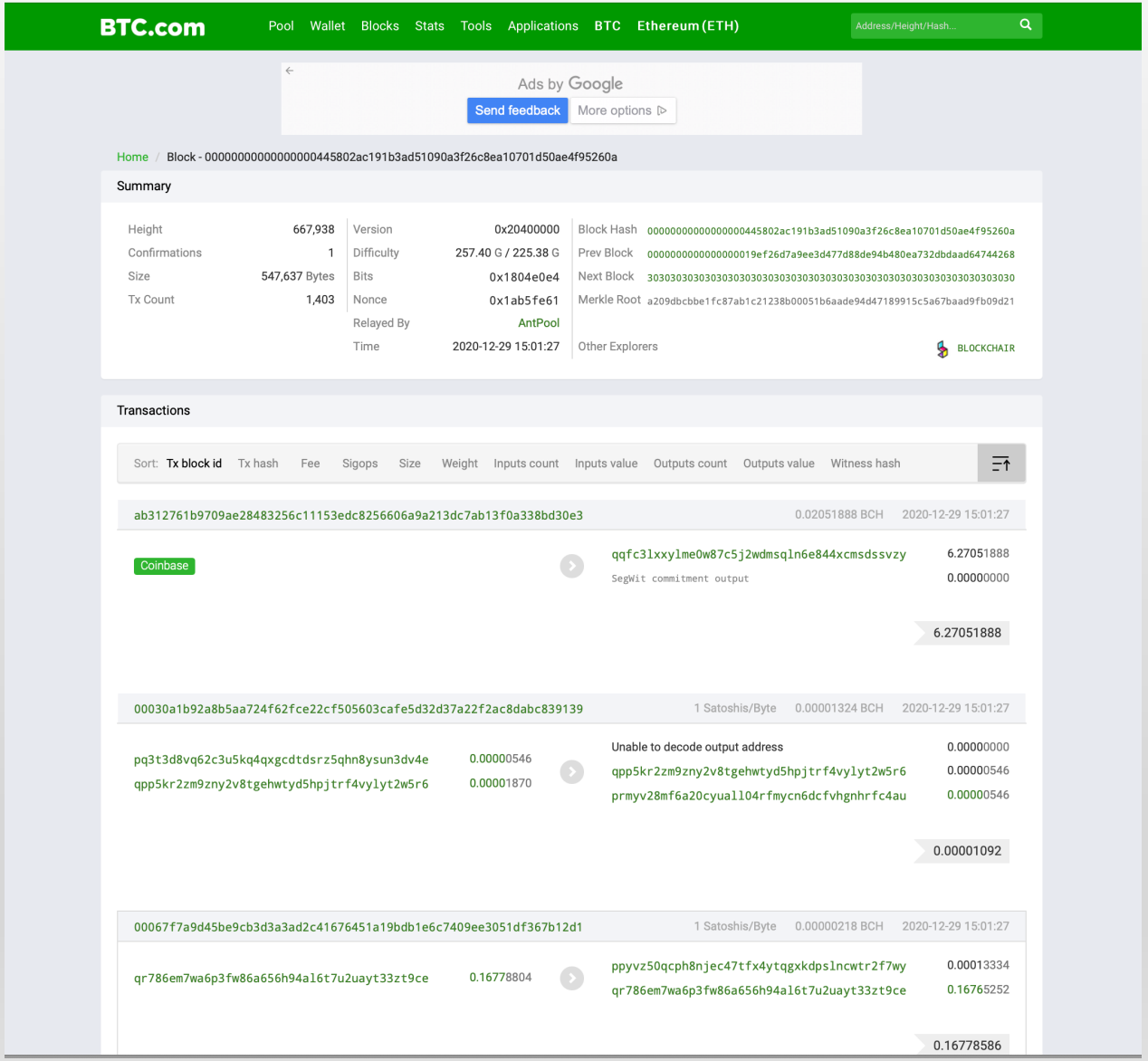
每条交易TX的字段

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

具体的一条交易里面的内容


```
{
 "version": 1,
 "locktime": 0,
 "vin": [
  {
   "txid":"7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
   "vout": 0,
   "scriptSig":
"3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f039ff08df09cbe9f6addac9
60298cad530a863ea8f53982c09db8f6e3813[ALL]
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d172787ec3457eee41c04f4938d
e5cc17b4a10fa336a8d752adf",
   "sequence": 4294967295
  }
 ],
 "vout": [
  {
   "value": 0.01500000,
   "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG"
  },
  {
   "value": 0.08450000,
   "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG",
  }
 ]
}
```

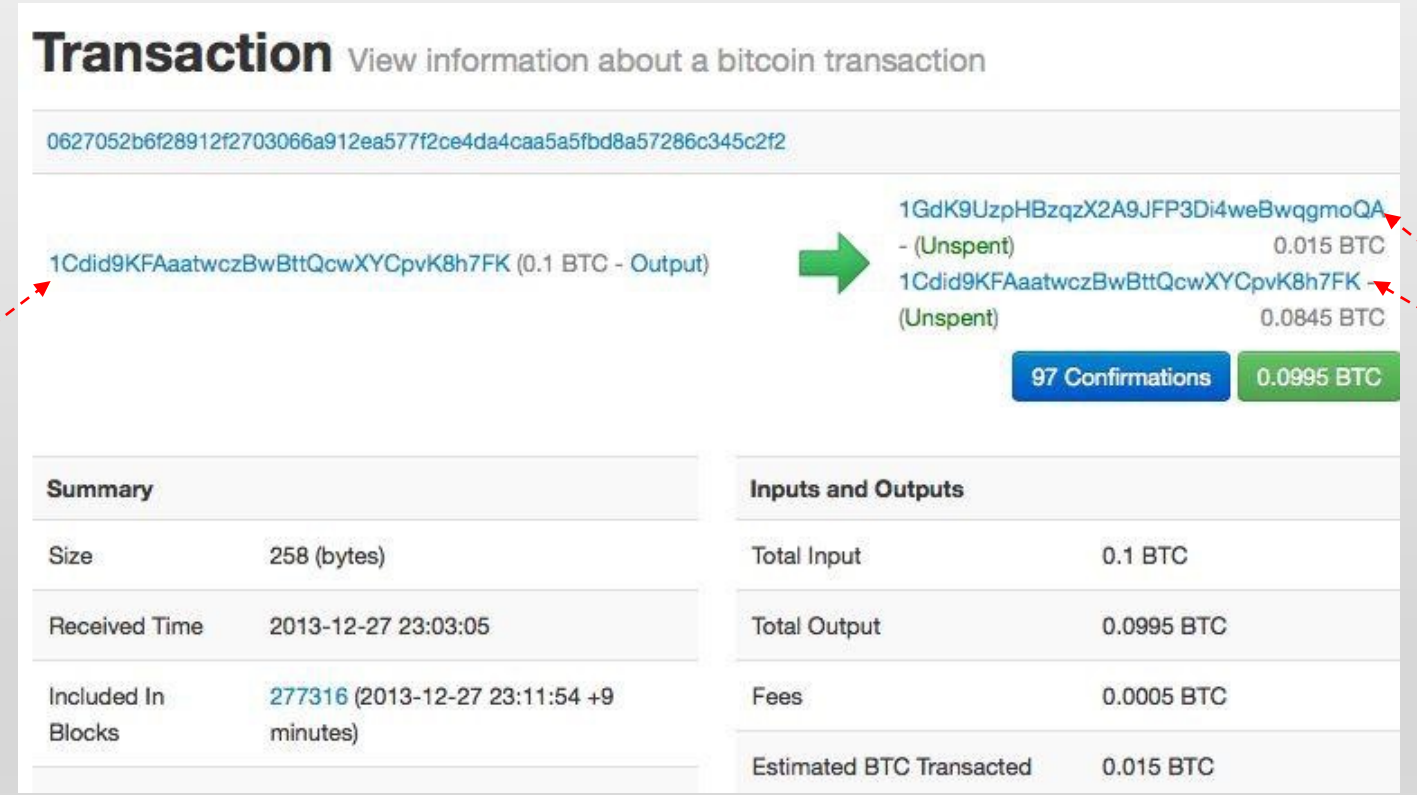
实时查看比特币区块链区块内容和交易详情　https://bch.btc.com/block



VICTORLAMP 煤油灯科技

比特币区块链交易

比特币交易是比特币系统中最重要的部分。根据比特币系统的设计原理，系统中任何其他的部分都是为了确保比特币交易可以被创建、在比特币网络中传播、通过验证，并最终添加入全球比特币交易总账簿（比特币区块链）。

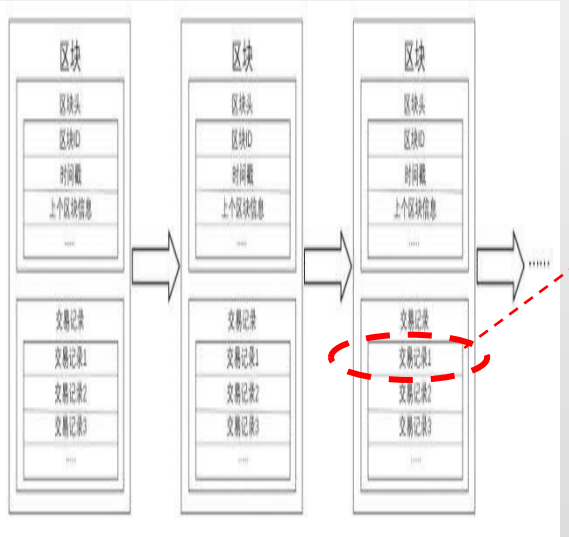


交易的输入

交易的输出

比特币区块链交易输入、输出字段

比特币区块链



一条交易TX的字段

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

交易的输入包含的字段

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

交易的输出包含的字段

字段	大小	描述
总量	8个字节	用聪表示的比特币值(10-8比特币)
锁定脚本尺寸	1–9个字节(可变整数)	用字节表示的后面的锁定脚本长度
锁定脚本	变长	一个定义了支付输出所需条件的脚本

什么是UTXO（Unspent Transaction Output）？

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。

比特币网络监测着以百万为单位的所有可用的(未花费的)UTXO。当 一个用戶接收比特币时，金额被当作UTXO记录到区块链里。

这样，一个用戶的比特币会被当作UTXO分散到数百个交易和数百个区块中。实际上，并不存在储存比特币地址或账戶余额的地点，只有被所有者锁住的、分散的UTXO。“一个用戶的比特 币余额”，这个概念是一个通过比特币钱包应用创建的派生之物。比特币钱包通过扫描区块链并聚合所有属于该用戶的UTXO 来计算该用戶的余额。

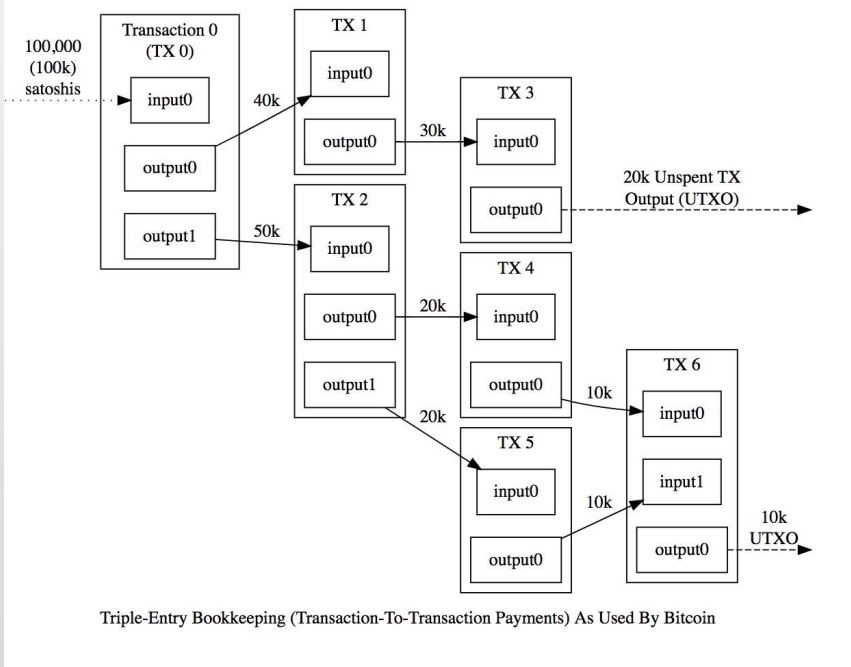
在比特币中，一笔交易的每一条输入和输出实际上都是 UTXO，输入 UTXO 就是以前交易剩下的， 更准确的说是以前交易的输出 UTXO。

除了 coinbase 交易（挖矿奖励）没有输入 UTXO 之外，其它交易都有输入和输出，都可以为多个。

可以简单理解UTXO就是一条账户变动记录，但是记账的方式与传统银行记账不一样。

比特币UTXO与传统银行账号记账模型对比

一直可以追溯到具体来自哪个coinbase的UTXO



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

比特币区块链UTXO模型

张三有6个比特币地址，放在一个钱包里面管理

UTXO1(比特币地址A：100BTC)

UTXO2(比特币地址B：50BTC)

UTXO3(比特币地址C：20BTC)

UTXO4(比特币地址D：10BTC)

UTXO5(比特币地址E：5BTC)

UTXO6(比特币地址F：1BTC)

- 比特币地址是钱包软件自动生成的，可以随时生成不同的比特币地址，可以多个，只有这个钱包软件知道是他生成的，其他人或软件无从知道来源于哪里。
- 比特币区块链系统只是纪录一个个的UTXO，每个UTXO会包含所属的比特币地址，这个UTXO属于这个特定的比特币地址。
- 你的钱包知道哪些地址是属于你的，对于你这个钱包的总比特币余额就是全网查找属于这个钱包管理所有特币地址的UTXO，进行累加就是比特币余额。

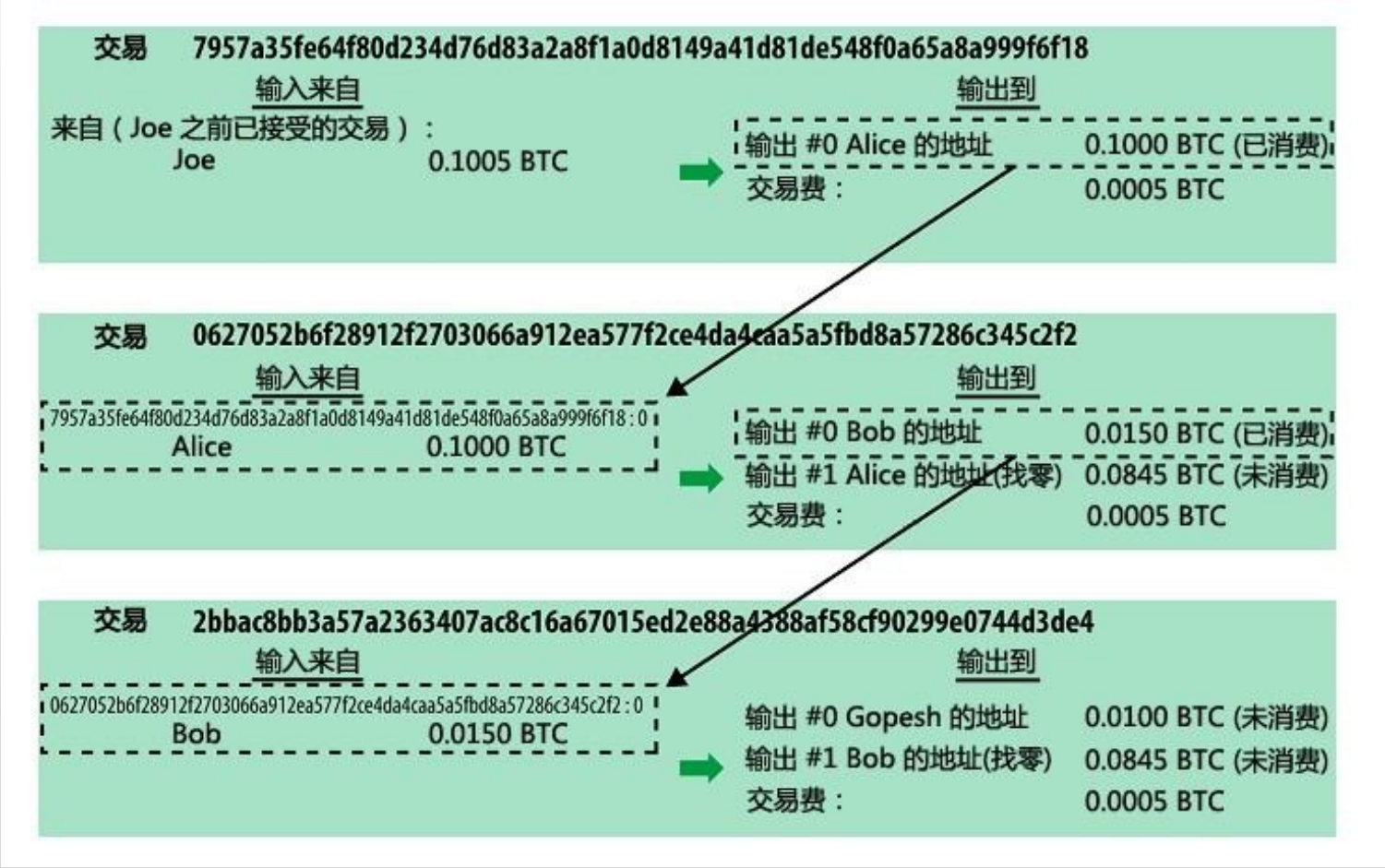
银行账户模型

张三有一个银行账号A

银行账号A：186元

- 银行给客户开账号，是预先确定的账号（例如60000123456789），通常还实名制，能够对应到身份证号。
- 账号纪录的是这个账号的输入和输出，进行加减算出余额。

比特币区块链交易中金额的流转



比特币UTXO与传统银行账号转账对比

比特币区块链UTXO模型

任务：
王先生要给张三转账10BTC

UTXO1(比特币地址A：100BTC)

UTXO2(比特币地址B：50BTC)

UTXO3(比特币地址C：20BTC)

UTXO4(比特币地址D：10BTC)

UTXO5(比特币地址E：5BTC)

UTXO6(比特币地址F：1BTC)

UTXO7(比特币地址G：10BTC)

- 张三通过钱包软件新生成一个比特币地址G，把这个地址告诉王先生
- 王先生把自己账户里面UTXO作为输入，把比特币地址G作为输出UTXO的地址
- 发送转账，经过比特币区块链网络的确认之后交易成功。比特币钱包计算的时候知道比特币地址G是自己的，所以加上10BTC，总数196BTC。

银行账户模型

张三有一个
银行账号A

银行账号A：196元

任务：
王先生要给张三转账10元

- 张三把自己的银行账号告诉王先生
- 王先生通过银行，向张安的账号转账，交易成功，账户余额变成196元。

接下来的问题就是：那我怎么知道哪个UTXO是我的呢？我的UTXO会不会被别人冒充领走了呢？

比特币区块链的交易验证：锁定脚本和解锁脚本

交易的输入包含的字段

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

交易的输出包含的字段

字段	大小	描述
总量	8个字节	用聪表示的比特币值(10-8比特币)
锁定脚本尺寸	1–9个字节(可变整数)	用字节表示的后面的锁定脚本长度
锁定脚本	**变长**	**一个定义了支付输出所需条件的脚本**

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。


```
{
 "version": 1,
 "locktime": 0,
 "vin": [
  {
   "txid":"7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
   "vout": 0,
   "scriptSig":
"3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204b9f039ff0
8df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e3813[ALL]
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d172787ec34
57eee41c04f4938de5cc17b4a10fa336a8d752adf",
   "sequence": 4294967295
  }
 ],
 "vout": [
  {
   "value": 0.01500000,
   "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7
OP_EQUALVERIFY OP_CHECKSIG"
  },
  {
   "value": 0.08450000,
   "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8
OP_EQUALVERIFY OP_CHECKSIG",
  }
 ]
}
```

锁定脚本和解锁脚本里面包含的内容是什么？---针对P2PKH(Pay-to-Public-Key-Hash)

比特币网络上的大多数交易都是P2PKH交易，此类交易都含有一个锁定脚本，该脚本由公钥哈希实现阻止输出功能，公钥哈希即为广为人知的比特币地址。由P2PKH脚本锁定的输出可以通过键入公钥和由相应私钥创设的数字签名得以解锁。

例如，我们可以再次回顾一下Alice向Bob咖啡馆支付的案例。
Alice下达了向Bob咖啡馆的比特币地址支付0.015比特币的支付指令，该笔交易的输出内容为以下形式的锁定脚本:

OP_DUP OP_HASH160 <Cafe Public Key Hash也就是咖啡馆的比特币地址> OP_EQUAL OP_CHECKSIG

脚本中的 Cafe Public Key Hash 即为咖啡馆的比特币地址，但这个地址不是基于Base58Check编码的。事实上，大多数比特币地址都显示为十六进制码，而不是大家所熟知的以1开头的基于Bsase58Check编码的比特币地址。
锁定脚本的解锁版脚本是:

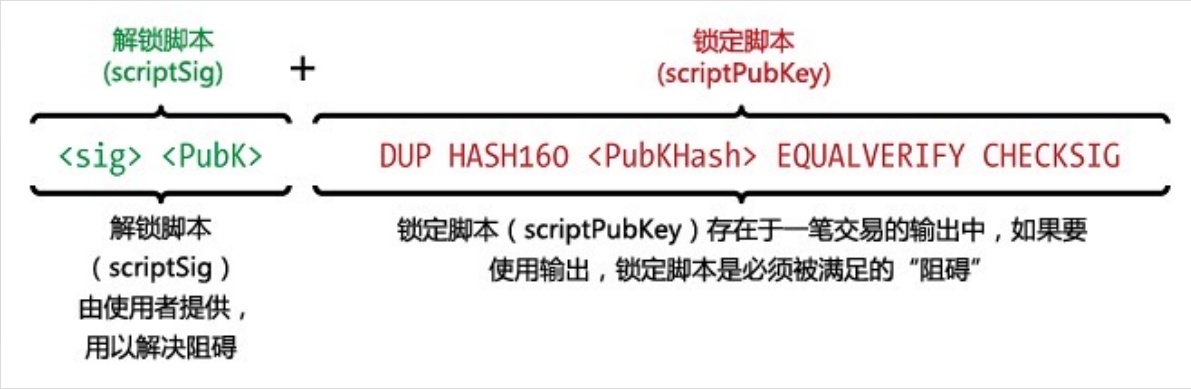
<Cafe Signature> <Cafe Public Key>

将两个脚本结合起来可以形成如下有效组合脚本:

<Cafe Signature> <Cafe Public Key> OP_DUP OP_HASH160 <Cafe Public Key Hash> OP_EQUAL OP_CHECKSIG

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(True)。即只有当解锁脚本得到了咖啡馆的有效签名，交易执行结果才会被通过(结果为真)，该有效签名是从与公钥哈希相匹配的咖啡馆的私钥中所 获取的。

比特币区块链的交易验证逻辑



接下来的问题就是：这么一串脚本怎么知道是啥意思？

比特币区块链脚本语言与脚本引擎

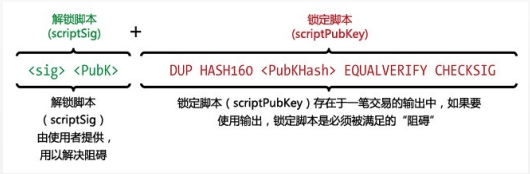
- 比特币交易脚本语言，也称为脚本，是一种基于逆波兰表示法的基于堆栈的执行语言。如果这让您听起来似乎在胡言乱语， 很有可能是您没学习过1960年的编程语言的缘故。脚本是一种非常简单的语言，这种语言被设计为能在有限的硬件上执行， 这些硬件类似简单的嵌入式设备，如手持计算器。它仅需最少的处理即可，而且不能做许多现代编程语言可以做的事情。当 涉及可编程的钱时，这是它的一个基于深思熟虑的安全特性。

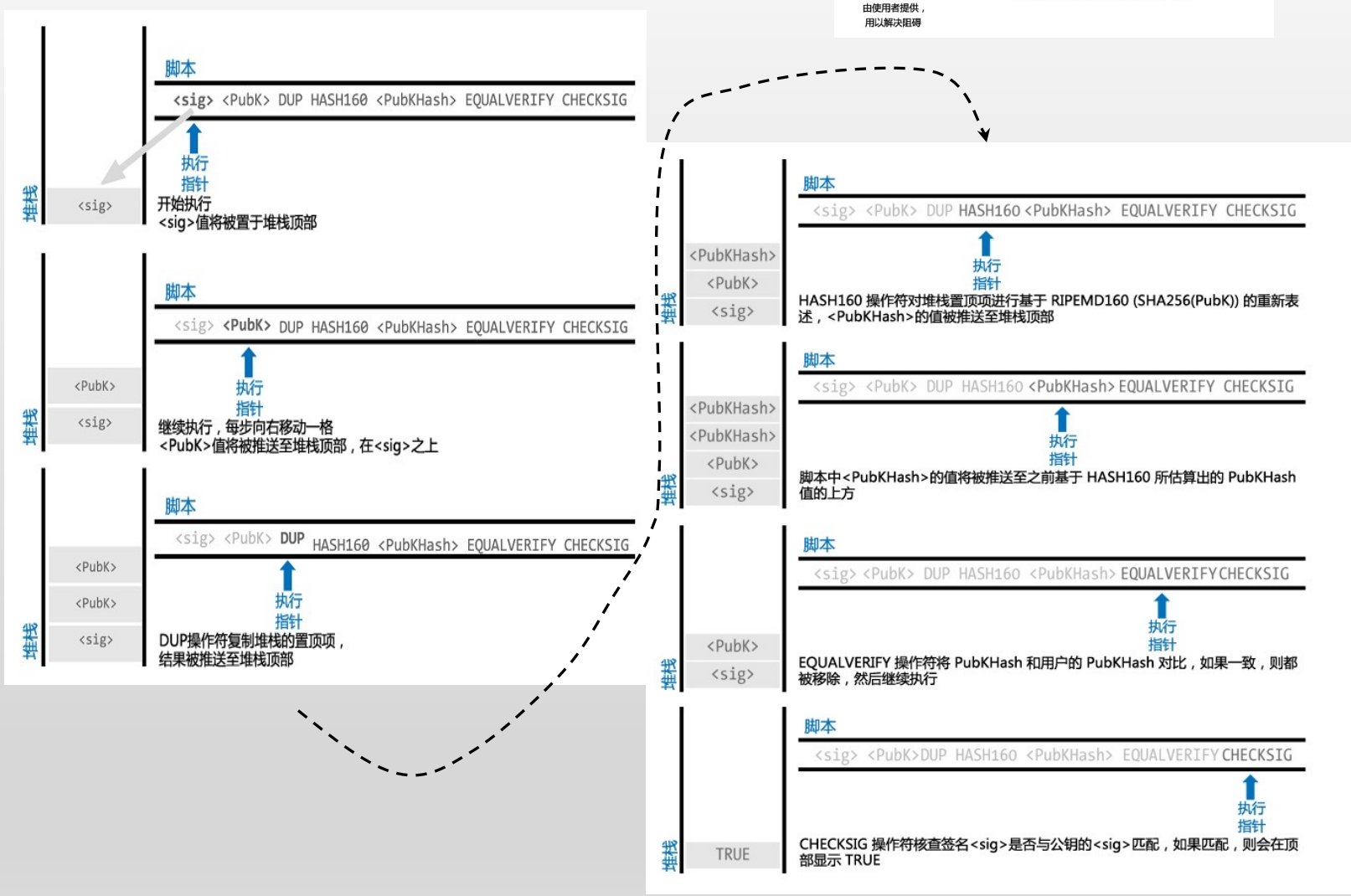
- 比特币脚本语言被称为基于栈语言，因为它使用的数据结构被称为栈。栈是一个非常简单的数据结构，它可以被理解成为一 堆卡片。栈允许两类操作:入栈和出栈。入栈是在栈顶部增加一个项目，出栈则是从栈顶部移除一个项目。

- 脚本语言通过从左至右地处理每个项目的方式执行脚本。数字(常数)被推送至堆栈，操作符向堆栈推送(或移除)一个或 多个参数，对它们进行处理，甚至可能会向堆栈推送一个结果。例如，OP_ADD将从堆栈移除两个项目，将二者相加，然后 再将二者相加之和推送到堆栈。

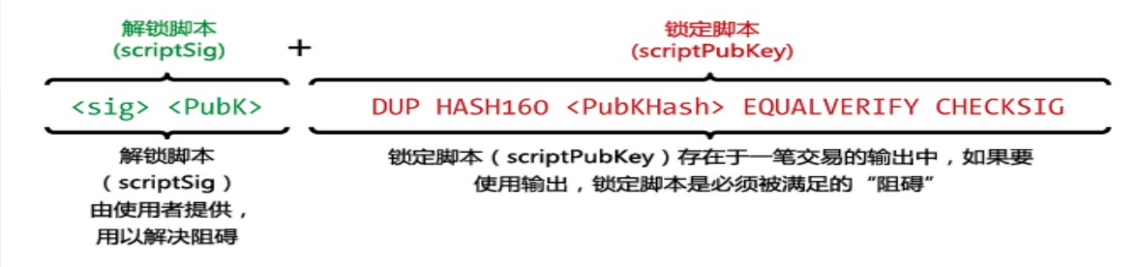
- 条件操作符评估一项条件，产生一个真或假的结果。例如，OP_EQUAL从堆栈移除两个项目，假如二者相等则推送真(表示 为1)，假如二者不等则推送为假(表示为0)。比特币交易脚本常含条件操作符，当一笔交易有效时，就会产生真的结果。

组合脚本一步步检验交易有效性的过程

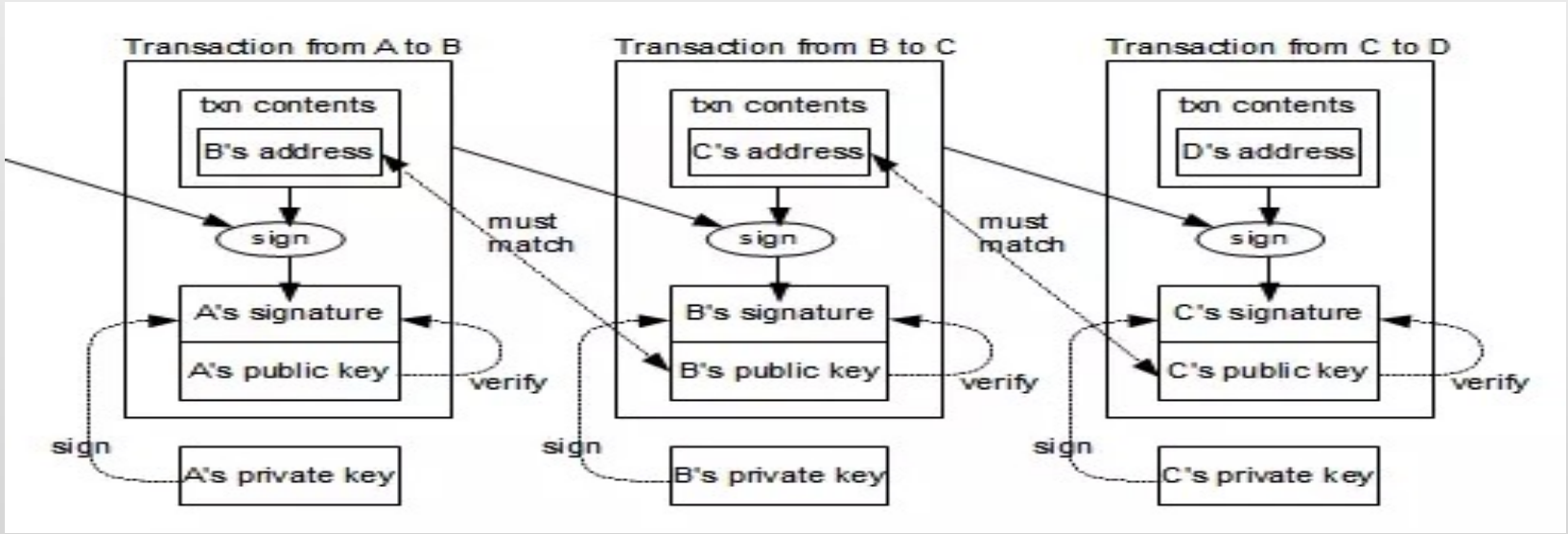




比特币区块链交易身份验证也是形成一个链式结构，保证交易的不可更改和安全性



交易输入的vin：UTXO　　交易输入的vin：UTXO　　交易输入的vin：UTXO



每条UTXO都是由脚本加密锁定的，只有交易的UTXO接收地址的属主才能够解锁和使用这条UTXO里面的比特币。

Coinbase交易（或Generation 交易，或生成交易）是怎么样

coinbase 交易中包含的各项数据。
交易中包含一个 input 和一个 output ，这里 input 就是 coinbase 。output 指向矿工的地址，总金额等于 coinbase 加上区块中全部交易的手续费。
创币交易不包含“解锁脚本”(又称作 scriptSig)字段，这个字段被coinbase数据替代，长度最小2字节，最大100字节。除了开始的几个字节外，矿工可以任意使用coinbase的其他部分，随意填充任何数据。 以创世块为例，中本聪在coinbase中填入了这样的数据“The Times 03/Jan/ 2009 Chancellor on brink of second bailout for banks”(泰晤士报 2009年1月3日 财政大臣将再次对银行施以援手)，表示对日期的证明，同时也表达了对银行系统的不信 任。

创世块里面的交易


```
{
 "version": 1,
"locktime": 0,
"vin": [
{
“coinbase”:“04ffff001d0104455468652054696d65732030332f4a616e2f32303039204
368616e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757
420666f722062616e6b73”,
"sequence": 4294967295
}
],

"vout": [
{
 "value": 50.00000000,
"n": 0,
"scriptPubKey":
{
"asm":
"04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e
51ec112de5c384df7ba0b8d578a4c702b6bf11d5f
OP_CHECKSIG",
"hex":
"4104678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f3550
4e51ec112de5c384df7ba0b8d578a4c702b6bf11d5fac",
"type": "pubkey"
} } ],}
```


普通交易tx里面的内容


```
{
 "version": 1,
 "locktime": 0,
 "vin": [
  {

"txid":"7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
    "vout": 0,
    "scriptSig":
"3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae
24cb02204b9f039ff08df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e3813[
ALL]
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336
376789d172787ec3457eee41c04f4938de5cc17b4a10fa336a8d752adf",
    "sequence": 4294967295
  }
 ],
  "vout": [
  {
    "value": 0.01500000,
    "scriptPubKey": "OP_DUP OP_HASH160 ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY
OP_CHECKSIG"
  },
  {
    "value": 0.08450000,
    "scriptPubKey": "OP_DUP OP_HASH160 7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY
OP_CHECKSIG",
  } ]}
```

Coinbase交易的收入：创币奖励和交易费

挖矿是增加比特币货币供应的一个过程。挖矿同时还保护着比特币系统的安全，防止欺诈交易，避免“双重支付”，“双重支付”是指多次花费同一笔比特币。矿工们通过为比特币网络提供算力来换取获得比特币奖励的机会。

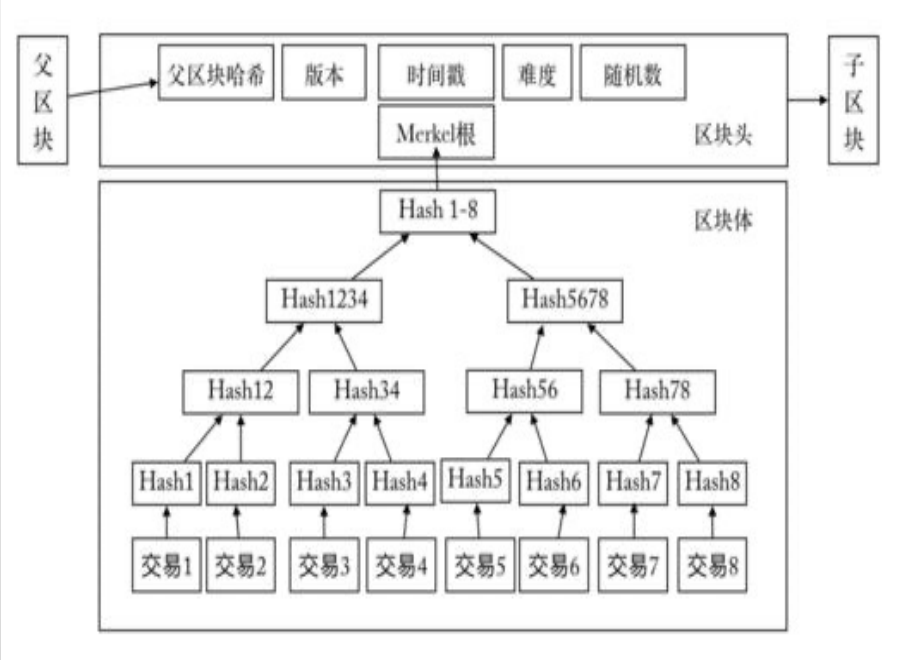
矿工们验证每笔新的交易并把它们记录在总帐簿上。每10分钟就会有一个新的区块被“挖掘”出来，每个区块里包含着从上一 个区块产生到目前这段时间内发生的所有交易，这些交易被依次添加到区块链中。我们把包含在区块内且被添加到区块链上的交易称为“确认”交易，交易经过“确认”之后，新的拥有者才能够花费他在交易中得到的比特币。

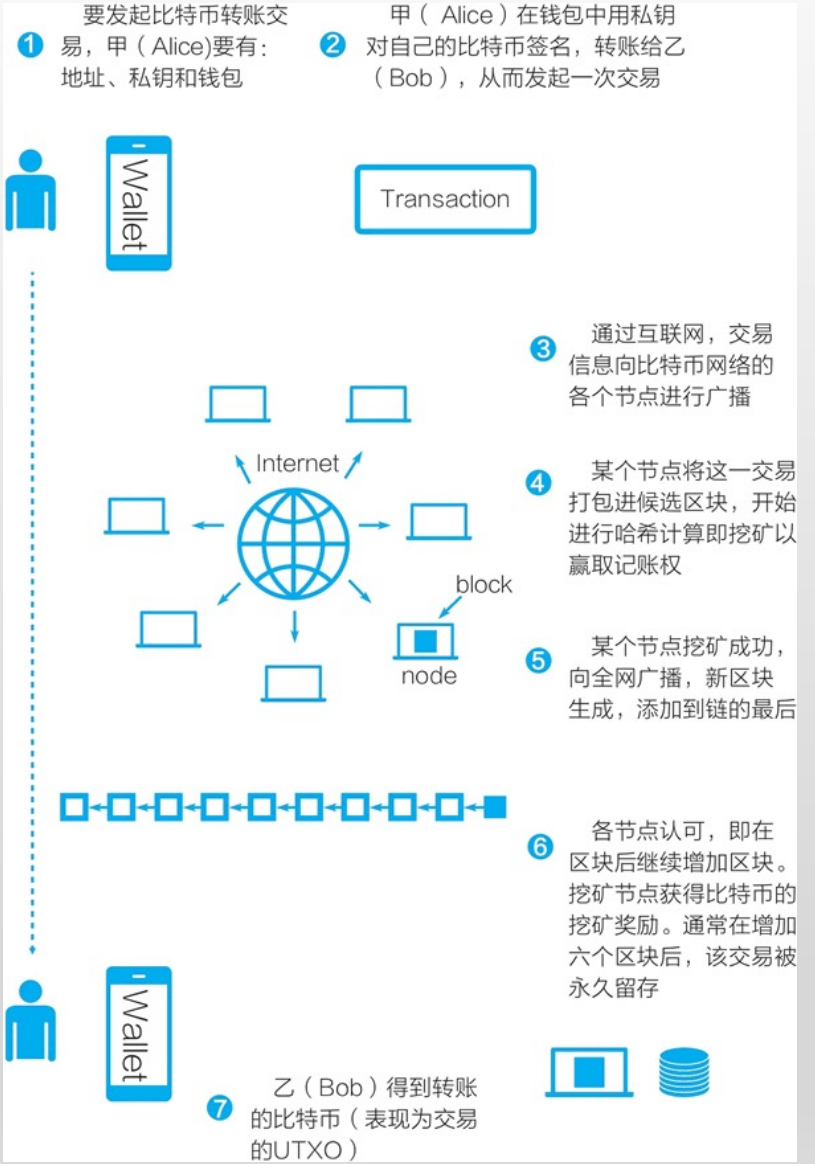
矿工们在挖矿过程中会得到两种类型的奖励:创建新区块的新币奖励，以及区块中所含交易的交易费。为了得到这些奖励， 矿工们争相完成一种基于加密哈希算法的数学难题，这些难题的答案包括在新区块中，作为矿工的计算工作量的证明，被称为“”工作量证明”。该算法的竞争的机制以及获胜者有权在区块链上进行交易记录的机制，这二者是比特币安全的基石。

新比特币的生成过程被称为挖矿是因为它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。比特币的货币是通过挖矿发行的，类似于中央银行通过印刷银行纸币来发行货币。矿工通过创造一个新区块得到的比特币数量大约每四年 (或准确说是每210,000个块)减少一半。开始时为2009年1月每个区块奖励50个比特币，然后到2012年11月减半为每个区 块奖励25个比特币。之后将在2016年的某个时刻再次减半为每个新区块奖励12.5个比特币。基于这个公式，比特币挖矿奖励 以指数方式递减，直到2140年。届时所有的比特币(20,999,999.98)全部发行完毕。换句话说在2140年之后，不会再有新的比特币产生。

矿工们同时也会获取交易费。每笔交易都可能包含一笔交易费，交易费是每笔交易记录的输入和输出的差额。在挖矿过程中成功“挖出”新区块的矿工可以得到该区块中包含的所有交易“小费”。目前，这笔费用占矿工收入的0.5%或更少，大部分收益仍来自挖矿所得的比特币奖励。然而随着挖矿奖励的递减，以及每个区块中包含的交易数量增加，交易费在矿工收益中所占 的比重将会逐渐增加。在2140年之后，所有的矿工收益都将由交易费构成。

回顾





谢 谢!

VICTORLAMP

Shenzhen VictorLamp Technologies CO. Ltd.
http://www.victorlamp.com

多媒体课程：《深入浅出区块链技术核心篇》

深入浅出区块链技术