**高等计算机体系结构**

**第三次作业**

**使用GPU进行比特币挖矿的技术调研**



学院：信息工程学院

成员：高俊龙 1701213594

姜 波 1701213603

夏雅娴 1701213643

日期：2018年1月11日

# 比特币基本概念

根据比特币交易平台Bitstamp显示，截至1月10日15时，一个比特币的交易价格约为1.4万美元，折合人民币超过9万元。

值得注意的是，此前中国央行等七部委叫停了国内所有ICO活动，金融监管部门也下发通知关闭了各家比特币交易场所。比特币在国内的热潮一度退却，但在国际市场，其价格却经历了一路暴涨，甚至一度触及20000美元。2017年比特币话题赤手可热。从挖矿的到跳广场舞的似乎都在谈论这比特币。而作为比特币的技术和基础架构，区块链的受关注度也“水涨船高”，技术与“区块链”挂钩的公司股价飞涨，多家公司也顺势宣布布局区块链行业。比特币到底是什么?

## 什么是比特币(BitCoin)

比特币（英语：Bitcoin）是一种用去中心化、全球通用、不需第三方机构或个人，基于区块链作为支付技术的电子加密货币。比特币由中本聪（又译中本哲史）（化名）于2009年1月3日，基于无国界的对等网络，用共识主动性开源软件发明创立。比特币也是目前知名度与市场总值最高的加密货币。

与大多数货币不同，比特币不依靠特定货币机构发行，它依据特定算法，通过大量的计算产生，比特币经济使用整个P2P网络中众多节点构成的分布式数据库来确认并记录所有的交易行为，并使用密码学的设计来确保货币流通各个环节安全性。P2P的去中心化特性与算法本身可以确保无法通过大量制造比特币来人为操控币值。基于密码学的设计可以使比特币只能被真实的拥有者转移或支付。这同样确保了货币所有权与流通交易的匿名性。比特币与其他虚拟货币最大的不同，是其总数量非常有限，具有极强的稀缺性。该货币系统曾在4年内只有不超过1050万个，之后的总数量将被永久限制在2100万个。

## 区块与块链

区块链（Blockchain）是比特币的一个重要概念，该概念在中本聪的白皮书中提出，区块链是一串使用密码学方法相关系产生的数据块（称为“区块”，block）。新增的数据块总能链接到上一个区块，即整条区块链的尾部。比特币点对点网络将所有的交易历史都存储在“区块链”（blockchain）中，所以区块链可以看作记录着比特币交易的账本。区块链是一群分散的客户端节点，并由所有参与者组成的分布式数据库，是对所有比特币交易历史的记录。中本聪预计，当数据量增大之后，用户端希望这些数据并不全部存储自己的节点中。为了实现这一目标，引入散列函数机制。这样客户端将能够自动剔除掉那些自己永远用不到的部分，比方说极为早期的一些比特币交易记录。

表1-1一个区块的记录格式

|  |  |  |
| --- | --- | --- |
| 数据项 | 描述 | 长度 |
| Magicno魔术数 | 总是0xD9B4BEF9 | 4字节 |
| Blocksize块大小 | 到区块结束的字节长度 | 4字节 |
| Blockheader块头 | 正整数 VI=VarInt | 1-9字节 |
| Transactioncounter  交易数量 | 正整数 VI=VarInt | 1-9字节 |
| transactions交易 | 交易列表(非空) | <Transactioncounter>许多交易 |

块链是所有比特币节点共享的交易数据库，这些节点基于比特币协议参与到比特币网络中来。块链包含每一个曾在比特币系统执行过的交易。根据这个信息，人们可以找到任何时候任一个地址中的币数量，每个区块包含前一个区块的HASH值，这就使得从创世块（创世块是指块链的第一个块）到当前块形成了一条块链，每个区块必定按时间顺序跟随在前一个区块之后，因为如果不知道前一块区块的HASH值就没法生成当前区块。要改变一个已经在块链中存在一段时间的区块，从计算上来说是不可行的，因为如果它被改变，它之后的每个区块必须随之改变。这些特性使得双花比特币非常困难。“长度”是被计算成块链的所有联合难度，而不是区块数目，可以防止创建大量低难度区块故意使块链分叉，从而让网络接受它成为“最长”的块链。如果一个块链中的所有区块和交易有效，则该块链有效，并且要以创世块开头。对于块链中的任何区块来说，只有一条通向创世块的路径。然而，从创世块出发，却可能有分叉。块链的具体结构可以参考图1-1。

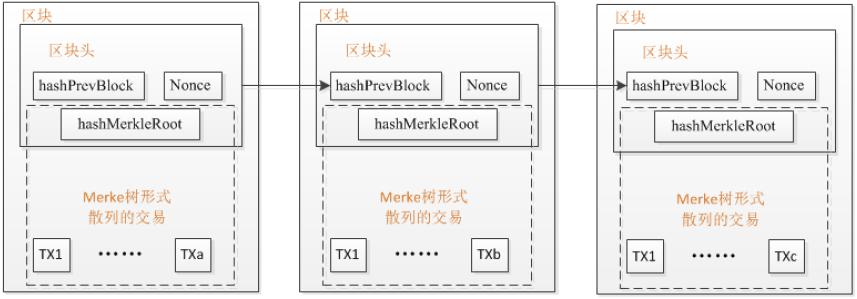


图1-1区块与块链结构

# 比特币技术原理

## 交易(Transactions)

每一位所有者（A）利用他的私钥对前一次交易单T1和下一位所有者（B）的公钥（即交易接收方的地址，用于说明当前交易的目标）签署一个随机散列的数字签名，A将此数据签名制作为交易单T2并广播全网。这样电子货币就发送给了下一位所有者。而收款人通过对签名进行检验，就能够验证该链条的所有者。比特币主要用ECDSA，即椭圆曲线签名算法，T1说明了该次交易的货币的来源（这部分货币是怎么到当前发起人这里来的）；数字签名则是发起方将前一次交易数据和接收方公钥连接起来并对其求Hash值x，再利用自己的私钥对x加密，便得到了这份数字签名。为防止双重支付，在每一笔交易结束后，这枚电子货币就要被造币厂回收，同时造币厂将发行一枚新的电子货币；而只有造币厂直接发行的电子货币，才有效。但该解决方案的问题在于，整个货币系统的命运完全依赖于运作造币厂的公司，因为每一笔交易都要经过该造币厂的确认，而该造币厂就好比是一家银行。

在造币厂模型里面，造币厂获悉所有的交易，并且决定了交易完成的先后顺序。为了排除第三方中介机构，系统内所有参与者都将拥有唯一公认的历史交易序列。收款人需要确保在交易期间绝大多数的节点都认同该交易是首次出现。

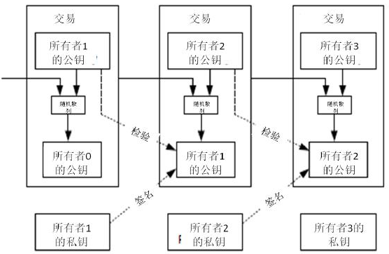


图2-1交易流程示意图

## 工作量证明（Proof-of-Work）

为了在点对点的基础上构建一组分散化的时间戳服务器，我们需要一个类似亚当•柏克提出的哈希现金。在进行随机散列运算时，工作量证明机制引入了对某一特定值的扫描工作。以SHA-256为例，随机散列值以一个或多个0开始。随着0的数目的上升,求解所需要的工作量将呈指数增长，而对结果进行检验则仅需一次随机散列运算。我们在区块中补增一个

我们通过反复尝试来找到随机数，进而就构建了一个工作量证明机制。该工作量证明机制解决了在集体投票表决时，谁是大多数的问题即：工作量证明机制的本质是一CPU一票。“大多数”的决定表达为最长的链，因为该链包含了最大的工作量。如果大多数的CPU为诚实的节点控制，则诚实的链条将以最快的速度延长，并超越其他的竞争链条。如果想对业已出现的区块进行修改，攻击者必须重新完成该区块和该区块之后所有区块的工作量，并最终赶上和超越诚实节点的工作量。

另一个问题是，随着硬件的运算速度在高速增长，而节点参与网络的程度则会有所起伏。为了解决这个问题，工作量证明的难度将采用移动平均目标的方法来确定，即令难度指向令每小时生成区块的速度为某一个预定的平均数。如果区块生成的速度过快，那么难度就会提高。

## 公开密钥密码体系

该体系简称公钥体系。在信息传递过程中，发送方通过一把密钥将信息加密，接收方在收到信息后，再通过配对的另一把密钥对信息进行解密，这就保证了信息传递过程的私密性与安全性。而密钥无非是一组数字，通过将原始信息与这组数字放在一起进行特定运算，就能够把信息转换为另外一种格式，从而实现加密。解密过程则刚好相反。在大多数情况下，一组密钥由公钥和私钥组成。私钥由自己保存，公钥则需要向其他人公开。在信息传递过程中，公钥和私钥相互配合，既能够对持有私钥的发信人进行身份验证，也能够确保发信人对自己发出的信息不能抵赖，还能够保证收发信息的完整性、防止中间环节被截获篡改。如果公钥丢失，还可以通过私钥进行恢复。但试图通过公钥反推出私钥的努力，从理论上来讲是基本不可行的，这就保证了私钥的私密性。

# 挖矿算法

如前面所述，挖矿就是在竞争对交易进行确认，在交易的确认操作，就是对当前的区块进行SHA-256解密。之所以选择SHA-256，是因为SHA-256的哈希值前面每增加一个0，寻找解的难度就增加了16倍，而且任何一个哈希值没有固定的解密算法，只能通过随机哈希碰撞策略。在SHA-256下，随机散列值以一个或多个0开始，我们在区块中补增一个随机数，这个随机数要使得该区块的随机散列值出现所需的那么多个0。挖矿的过程实际上就是找个某个随机数Nonce满足公式：



其中Version和hashPrevBlock是固定不变的；Timestamp是每隔几秒更新一次，bits是每产生2016个区块就根据实际挖矿速度和目标挖矿速度动态进行调整，bits的调整同样反映到Target中，从而来调整挖矿的难易程度。Nonce的范围是0~2^32，即挖矿的过程要从0到2^32不断尝试不同的Nonce，从而来找到一个有效解。

在比特币系统中，总共存在约2100万个比特币，每次挖到一个新的区块时，都会伴随着M个比特币的产生。而随着剩余比特币数目的减半，M的值也将减半，M的初始值为50。根据设计，系统会控制平均每10分钟产生一个新的区块，所以在比特币产生的前四年，每次挖到新的区块将得到50个比特币，4年内大约产生50\*（60/10）\*24\*365\*4，约1050万个比特币，而在这之后每个区块的奖励将变成25个比特币。按这速度，大约2040年后所有的比特币都将被挖出，此后就算有的新的区块产生也不会再有比特币产生，每个新区块的奖励变成零比特币。

前面提到挖矿会伴随着新的比特币的产生，这些比特币归属于得到这个解的用户，另外这个新的区块中所有交易产生的交易费也归属于挖矿者，作为对解决新区块的奖励。这种激励机制保证了即使随着时间的推移，挖矿的难度越来越大，挖矿者仍然能够通过新挖到的比特币和比特币交易费构成的激励维持继续进行挖矿。为了确认新挖到的比特币的归属，在新产生的每个区块中，第一笔交易成为生产交易（coinbase），该交易中记录的比特币地址是获得该比特币奖励的地址，这个交易记录也就确认了该比特币的归属问题。

# 使用GPU的高效哈希算法

## 基于GPU的哈希算法

其思想基于杜鹃哈希表（Cuckoohashing）。区别于其他方法建立哈希表，杜鹃哈希表由几个子表构成，并且每个子表拥有各自的哈希函数，这样每个数据可以通过各个子表的哈希函数映射到相应的子表中（如下图a，数据A可以通过哈希函数h1(x)、h2(x)、h3(x)分别映射到三个子表中），这样就可以保证每个数据可以映射到各个子表中的固定位置，当数据在各个子表中的位置都已被其他数据占据时（如下图b，数据A探测完3个子表发现都已有数据占据相应位置），就会回到第一个子表把占据位置的数据踢出，被踢出的数据执行同样操作（如下图c，数据A在第一个子表中踢出数据P后，P探测第二个子表后找到相应位置后插入）。

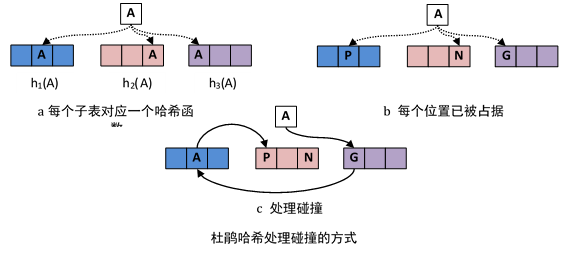


图4-1杜鹃哈希处理碰撞的方式

杜鹃哈希表可分为串行杜鹃算法构建哈希表和并行杜鹃算法构建哈希表。

## 并行杜鹃算法构建

在GPU上可以并行把数据插入杜鹃哈希子表中，在整个过程中每个线程负责管理一个数据，最终把数据存进哈希表中，主要问题是当发生碰撞时，确保能够正确处理好线程之间的竞争关系。如下图，所有数据开始通过第一个子表的哈希函数并行探测子表（如下图a，数据P、C、A、M、N通过第一个子表的哈希函数并行计算哈希值后探测子表），CUDA能确保发生碰撞的地址有且只有一个数据被成功插入子表之中，随后进行一次线程同步，线程会检查自身管理的数据是否插入当前子表之中，若没有插入成功，那么就移至下一个子表中继续同样的操作（如下图bc，数据并行探测完前两个子表后，数据P、N都没有插入子表，随之移到第三个子表继续探测），这是一个轮转调度的过程，当所有子表都遍历完后，没有在子表中找到存储位置的数据就回到第一个子表进行踢出操作，被踢出的数据再重复同样的过程，直到所以数据都找到了存储位置或达到一个最大迭代次数而停止，若是后者，那么每个子表会选择新的哈希函数后重新建表。

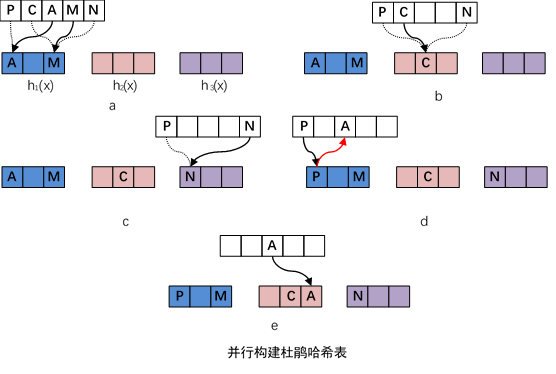


图4-2并行构建杜鹃哈希表

对于较小的数据集，在GPU上构建杜鹃哈希表的整个过程高效，但对于较大数据集，在GPU上运行的效率会降低，原因有两方面：一方面较大数据集只在一个哈希表中探测，意味着每次迭代引起GPU上全局存储器被多次访问，虽然全局存储器能够提供很高的带宽，但同时也具有较高的访问延迟；另一方面由于算法自身缺陷，会出现数据插入不成功的情况，最坏情况是当只有一个数据插入不成功时，都需要选取哈希函数重新建表，重新建表的代价消耗非常高。

基于以上原因并结合GPU上多层存储空间，解决以上弊端的方法是使用双层结构，目的是分而治之，第一层利用一个哈希函数把大数据集分成多个部分（如下图，把数据集被分成两个部分），每部分会分配给CUDA不同的线程块，由于线程块之间处理数据相互独立，这样可以在线程块对应的共享存储器中利用杜鹃哈希方法并行建造哈希表，这样减少了在建表过程中线程访问全局存储器的次数，共享存储器是GPU片内的高速存储器，只能被同一线程块中的所有线程访问，并且线程对共享存储器访问速度强于全局存储器。由于线程块之间是相互独立地并行执行，所以当一个线程块建造杜鹃哈希表失败时不会影响其他线程块，最坏情况当只有一个数据插入不成功时，也只是针对当前线程块选取哈希函数重新建表。当所有线程块成功地完成对应哈希表构建后，再把这些哈希表拷贝回全局存储器合成一个大的哈希表，这样就完成整个哈希表的构建。

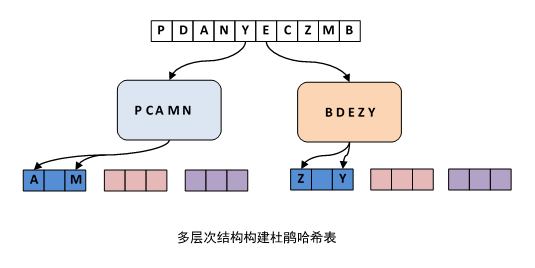


图4-3多层次结构构建哈希表

若要在GPU上快速建立杜鹃哈希表需要两个阶段：一是把大量数据分成几个块，二是针对每一部分分别建立哈希表，最后把每部分合并成一个大的哈希表。

**构建表的第一阶段**：

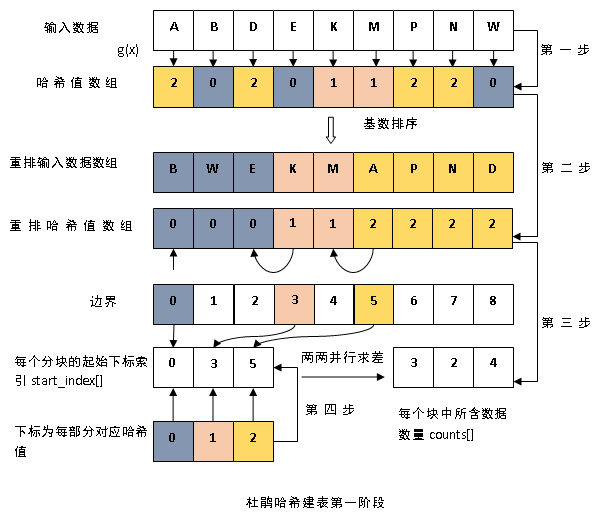


图4-4杜鹃哈希建表第一阶段

如上图，输入的数据集在GPU端经过哈希函数g(x)并行计算出对应的哈希值（如上图第一步），选择一个简单随机形成的哈希函数g(x)进行第一阶段分类，如公式1：



其中，a和b是16bit随机形成整数，p为1900813（大素数），B为第一阶段所分块数；当并行完成哈希值计算后，输入数据与哈希值数组基数排序，这样就把哈希值相同的数据聚类在一起（如上图第二步）；在重排哈希值数组中，线程并行计算出块与块之间的边界，各个哈希值对应的线程并行比较在它前一个的哈希值，若相同则不做处理，若不相同，则把下标记录在下标索引数组start\_index[]中，特别地，重排哈希值数组中第一个元素下标需记录在数组start\_index[]中，在生成后的下标索引数组start\_index[]中，并行计算其中两两值之差放于数组counts[]中，counts[]反映的是第一阶段分的每个块大小，若counts[]中一个值大于设定的C值，需要再次选择a和b重新执行第一阶段的整个过程（如上图第三步）。

第一阶段任务完成以后，当数据需查询被分到哪个块时，经过g(x)计算哈希值后，把哈希值作为下标在start\_index[]中查找，得到在重排哈希值数组中的起始位置，结合数组counts[]限定查找范围即可找到（如上图第四步）。

**构建表的第二阶段：**

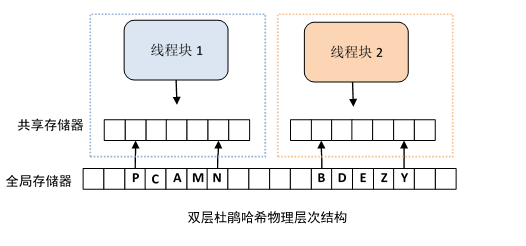
****

图4-5双层杜鹃哈希物理层次结构

第一阶段把分好的块从全局存储器拷贝到对应线程块中的共享存储器后，每个线程块之间各自独立地建立杜鹃哈希表，如上图为双层杜鹃哈希对应的物理层次结构。

第二阶段每个线程块选择的哈希函数与第一阶段选择哈希函数类似，都是简单的哈希函数，如公式2：

其中，S为杜鹃哈希子表的大小，每个子表都对应一个哈希函数hi(x)，hi(x)对应一对ai和bi，而ai和bi都是通过两个随机16bit整数异或操作产生。

在每个线程块中，每个数据对应的线程并行探测第一个哈希子表，探测完并做块内线程同步后，再回到子表中检查数据是否成功插入（在竞争过程中是不知道哪个线程竞争成功），若成功则不做处理，若不成功则移到下个子表中继续探测。当经过第一轮迭代操作之后无数据剩下，则表明此线程块中杜鹃哈希表建立成功，若还有数据没有插入哈希表，那么就会回到第一个子表踢出相应数据，被踢出的数据再重复同样的操作。若达到迭代次数A还没建立成功，则每个子表重新选择设置a和b后，再重新建表直至成功，并记下此时建表成功时选择的a和b，供查询数据使用。

当每个线程块中的哈希表建立成功后，把每个共享存储器中的杜鹃哈希表拷贝回全局存储器，当需要查找某个数据时，按照建表成功时选择的哈希函数计算相应哈希值，最终找到相应的数据。

## GPU上的MD5计算

MD5(Message-DigestAlgorithm5)算法是一种通过散列生成信息摘要的算法，属于分组加密算法，生成一个128-bit的哈希值。对于消息m产生长度为128bit的消息摘要h，可表示为h=MD5(m)。

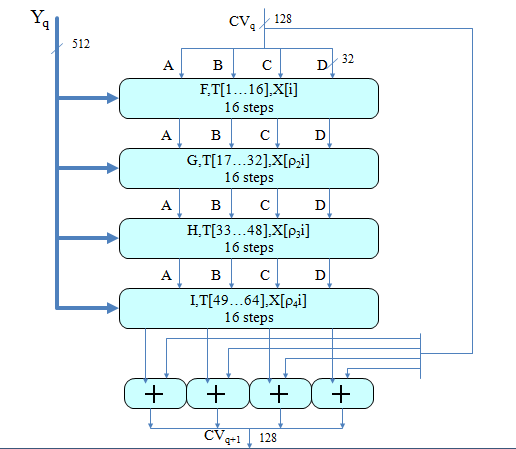


图4-6CUDA并行计算MD5

MD5进行四轮循环运算，每一轮16步。在对m进行SHA-256操作时，主要步骤为：附加填充比特。对m进行填充，使其位数与448mod512同余；附加64位的长度值。使得其总位数是512的倍数；初始化4个128bit的缓存，A=0x01234567，B=0x89abcdef，C=0xfedcba98，D=0x76543210；将处理后的m进行分组成512位的M{i}，然后对每个M{i}按上图进行MD5操作，由4轮共64步迭代运算组成；对于MD5算法的最后一个分组的输出便是128bit的哈希值。

由于MD5算法属于分组加密算法，各组数据之间的运算时相互独立的，不存在相关性，故MD5本身数据上具有很大的并行性，很适合使用GPU进行运算。

# 使用GPU进行高性能比特币挖矿的技术讨论

## 比特币挖矿的两种方式

比特币挖矿，早期一般都是solo方式:单枪匹马淘金，随着挖矿设备性能的指数级提高，无法保障收入，现在一般都是采用加入矿池的方式。

### solo 方式

solo矿工一般通过bitcoind来网络上获得新的交易。挖矿软件(比如50Miner,GUIMiner之类)周期性的调用RPC方法(getblocktemplate)来从bitcoind取得新交易，这个方法返回新交易的一览+coinbase交易。

挖矿软件用得到的交易构筑一个新的区块，并创建一个80字节的区块头部。然后，挖矿软件把区块头部和难度设定传给挖矿硬件(比如CPU，GPU，ASIC等；挖矿硬件穷举计算所有可能的头部nouce值，来寻找是否有符合难度设定的hash值。如果没有，那么挖矿软件会在coinbase中追加额外的nouce值，再交给挖矿硬件来计算，否则挖矿硬件把带有合适nouce值的区块头部返回给挖矿软件。挖矿软件把区块头部+区块做成完成的区块传递给bitcoind。bitcoind把区块广播到P2P网络，等待网络确认。

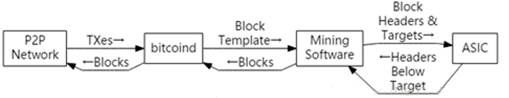


图5-4 solo方式比特币挖矿

### 矿池方式

挖矿计算量越来越大，矿工得到收益的概率不但在降低，而且越来越不稳定。所以出现了矿池方式，大家一起挖，挖到了以后收益按照出力的多少来分配。

基本做法是：矿池把目标难度降低几个数量级，然后分配给各个矿工，矿工可能找到了很多符合矿池目标值的header，这个被称为shares。其中，某个header碰巧符合目标难度值，则被矿池提交给P2P网络，而收益按照各个矿工提交的shares多少来分配。可以和上图5-4比较辅助理解挖矿过程。

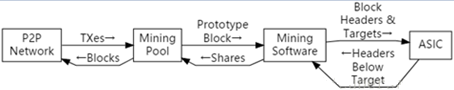


图5-5 solo方式比特币挖矿

## 整体流程概述

基于以上概念，比特币的工作流程如下：

1. 全网广播新的交易，交易通过公钥数字签名确认交易的双方和比特币的归属；
2. 每个矿工节点都将收到交易信息加入到正在解决的区块中；
3. 矿工节点都尝试不同的随机数，进行hash计算，试图找一个具有足够难度的工作量证明（workofproof），此过程是一个交易确认过程；
4. 当一个节点找到了一个工作量证明，新的区块产生，同时新区块中的生产交易确认了新产生的比特币的归属；
5. 当且仅当新区块中的交易都是之前未被前面区块记录的，该区块才能加入块链，避免了重复的交易。

# 参考文献

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. 2008.

[2] Alcantara D A, Volkov V, Sengupta S, et al. Building an efficient hash table on the GPU[J]. GPU Computing Gems, 2011, 2: 39-53.

[3 ]Dev J A. Bitcoin mining acceleration and performance quantification[C]//Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on. IEEE, 2014: 1-6.

[4] Lewenberg Y, Bachrach Y, Sompolinsky Y, et al. Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis[C]// International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2015:919-927.

[5] Plohmann D, Gerhards-Padilla E. Case study of the Miner Botnet[C]// International Conference on Cyber Conflict. IEEE, 2012:1-16.