SMN（Super Master Node）技术白皮书

摘要: 这是一款以中本聪所开发的比特币为基础，改进并添加了诸如双层奖励制网络---也称为主节点网络，等多项新功能的加密数字货币。其中还包含为提高可互换性的匿名支付（Darksend），和在不依赖中心权威下实现即时交易确认的即时支付功能（InstantX）。

介绍

2009 年，中本聪提出比特币的概念，自那以后，比特币已迅速在主流应用和商业用途中传播开来，成为首个吸引大量用户的数字货币，是数字货币史上的里程碑。不过从完成交易的角度来看比特币接收的情形，我们可以发现一个重要问题，就是比特币区块确认交易的时间过长，而传统的支付公司已找出使买卖双方实现比特币交易零确认的解决方案，但这一解决方案通常是要在协议之外采用可信赖的第三方完成交易。

比特币提供假名交易，实现发送者和接受者之间一对一交易的关系，

并能永远记录全网发生过的交易。比特币只提供低层次的隐私保护，这点在学术界众所周知，尽管有此不足，许多人仍然相信区块链记录的转账历史。

基于中本聪成果，SMN 以保护隐私为要旨的加密数字货币。我们在比特币概念的基础上进行了一系列的改进，由此诞生出一个去中心化的和具备良好匿名性的加密数字货币，它支持防篡改的即时交易，又有能为 SMN 网络提供服务奖励制的点对点次级网络。

超级主节点网络

X16R 全节点是运行在 p2p 网络上的服务器，让小节点使用它们来接受来自全网的动态变化。这些全节点需要显著的流量和要消耗大量成本的其它资源，由此在一段

时间内会观察到比特币网络上的这些节点数量呈现稳步下降的趋势，使区块广播

的时间需要额外增加。为解决这问题，提出了许多方案，例如引入微软研究的新奖励计划和 Bitnodes 激励计划。

这些节点对网络的健康而言十分重要，它们能让客户端同步和通过全网快速广播

信息。我们提议增加次级网络，名为SMN主节点网络。这些节点将具有高可用性，

而且在为网络提供符合一定要求的服务后能够得到主节点服务奖励。。

主节点奖励计划——成本和奖励

比特币网络全节点锐减的主要原因是缺乏对运行节点的奖励。随着时间的推移，全网接入的用户会更多，对带宽的需求会更高，对节点运行者的资金需求也更多，结果使运行全节点的成本提高。考虑到成本的上升，节点运行者必须要降低他们的运行成本或者运行轻客户端，但这样完全不利于网络健康。

正如比特币网络一样，主节点是全节点，但不同的是主节点必须对全网提供一定的服务，并需要一定量的押金才能加入。押金不会丢失，在主节点运行时也是安全的。这可让投资者为全网提供服务的同时，赚取一定的投资收益，减少了价格的波动性。

运行一个主节点，需要存储 10000SMN。当主节点生效时，它可为全网的客户端提供服务，并以利息的形式获取奖励。这就使得用户为这项服务投资，但同时得到一定的回报。主节点获取的收益是来自同一个矿池，大约有 60%的区块奖励纳入到这个计划中。

考虑到主节点奖励计划的奖励率是固定的百分比，还有主节点网络节点存在波动的事实，预计主节点奖励会根据当前生效的主节点总数作出变化。

SMN 的主节点奖励通过去中心化的确定性队列和概率性的选择来确定。

全球列表

全球列表囊括了所有的主节点。各个主节点在列表中的位置取决于它们在网络上进行最后一次支付的时间，而不是在区块链上的位置。在网络上新创建的主节点和刚收到支付的主节点将被列在列表的末尾。除此之外，使用远程过程调用命令‘masternode start’ 或 ‘masternode start-alias’ 来重启的活跃主节点也将被列在列表的末尾。如需避免这样的结果，用户可以使用远程过程调用命令‘masternode start-missing’ 。随着某些主节点移至末尾，其它主节点将会向前推移到列表顶端。一旦主节点移到列表的前 10%，那么它就有可能从候选池中被选中。

候选池

候选池指的是在主节点全球列表中的前 10%。它的规模取决于 SMN 主节点的总数量。举个例子，如果全球共有 4500 个活跃的主节点，那么候选池中共有 450 个主节点。一旦进入候选池， 那么主节点是否会得到奖励将由哈希熵来决定。前一百个区块的区块哈希值确定了哪一个主节点将得到奖励。前 100 个区块的工作量证明将会与双 SHA256 的交易哈希值及候选池中的所有主节点的索引进行比较。与区块哈希值最相近的主节点将被选作得到奖励的主节点。

概率

由于主节点是否得到奖励取决于区块哈希熵，因此我们无法预测奖励什么时候会产生。主节点运营者应当预见到支付间隔可能存在较大差异。一旦主节点进入候选池，那么就有可能得到奖励。在上述例子中，（4500 个活跃主节点中的）450 个主节点位于候选池当中，因此，这 450 个主节点中选并获得区块奖励的概率都是 1/450。

下表显示了主节点在特定时间段内被选中的可能性。举个例子，上述 450 个候选主节点在 12 个小时内中选的概率约为 46%。下表不会（也无法）告知用户主节点在特定时间段之后中选的概率。举个例子，如果某一个主节点在过去的 12 个小时内没有中选 - 据列表所示，这个情况发生的可能性为 54%- 那么，它在下一个区块中选的可能性不是 46%，而是 1/450。综合考虑这些计算，如果用户持有某个位于候选池中的主节点，并且在 48 个小时内都没有中选，那是很低概率的事情，这个概率还不到 10%。不过，即时这样的情况发生了，那么该主

节点在其它区块中选的几率仍然是 1/450。一旦某个主节点被选中，它将会被移至列表的末尾。在再次进入候选池之前，它都不会再被选中。

小时	区块	概率
1	23.07	5.00%
2	46.14	9.76%
3	69.21	14.27%
4	92.28	18.56%
6	138.42	26.50%
8	184.56	33.67%
10	230.70	40.14%
12	276.84	45.98%
18	415.26	60.30%
24	553.68	70.82%
30	692.10	78.56%
36	830.52	84.24%
42	968.94	88.42%
48	1107.36	91.49%
72	1661.04	97.52%
96	2214.72	99.28%

挖矿供应

SMN 采用另一种可降低挖矿引起的通胀的方法，就是每 43200(约每月)个区块的供应进行 10%的减产，这不同于其它数字货币的减半。另外，每个区块的供应量与全网的矿工数直接相关，更多矿工的参与意味着更少的挖矿奖励。
SMN 的开取计划最终在 2035 年左右挖矿才会停止。

总量固定在：294390000 枚 SMN。

为什么 SMN（Super Master Node）采用 X16R 算法。

背景

加密货币使用的哈希算法的演进历史，大致是：从比特币（Bitcoin）的 SHA256，到莱特币（Litecoin）的 Scrypt，然后是以太坊（Ethereum）的 Ethash，达世币（Dash）的 X11，以及后来的 X13，X15，X17。

X16R 是这个演变过程中的下一代算法。

改变算法的目的是为了减小专用硬件对挖矿生态的影响。

比特币最初的设计是希望全世界各地的普通电脑进行挖矿。

随着比特币价值的增加，专为并行化处理设计的硬件开始显示出优势。

于是，挖矿进入 GPU 时代。

当挖矿的经济效益进一步提升后，使用可编程硬件在经济上变得可行，例如比 CPU 和 GPU 更有优势的现场可编程门阵列（FPGA）。

再下一步，便是制造专门为挖矿定制的芯片。

专用集成电路（ASIC）成为挖矿的主宰，使用其他技术的挖矿设备将变得不切合实际。

最终，随着设备的不断升级挖矿进入了更快和能效比更高的 ASIC 时代。

不幸的是，这种转变带来了挖矿中心化的问题。

虽然任何人都可以订购这些 ASIC 设备，但是地理位置上更靠近生产商的人可以享受到物流时间短的好处。电力是挖矿成本的重要部分，能够获得便宜的电力是非常重要的。

中国拥有大量 ASIC 矿机生产商，并且在某些省份可以获得廉价电力，这必将导致挖矿集中在这里。

对抗

减少 ASIC 矿机影响的一个解决方案是使用内存密集型的哈希算法。莱特币的 Scrypt 和 ZCash 使用的 Equihash 就是采用了这种思路。虽然有矿工使用 ASIC 矿机运算 Scrypt，但相对于 SHA256 算法下 ASIC 对 GPU 的优势，这种优势并不大。目前还没有针对 Equihash 的 ASIC 矿机出现。

另一种方法是将多种哈希算法串联起来，一个哈希算法的输出将作为下一个哈希算法的输入。达世币（最早被称为 DarkCoin），采用了这种思路，设计了 X11 算法。X11 把 11 种哈希算法串联起来使用，来加大 ASIC 开发的难度。

这种方法一度阻止了 ASIC 的出现，但是，目前也有了多家厂商在生产 X11 算法的矿机。

与 X11 类似，一些其他币种使用了 X13，X15，甚至是 X17，X17 串联了 17 种哈希算法。

固定顺序串联哈希算法的做法，依然可以设计出 ASIC。串联更多的哈希算法，可以增加开发 ASIC 的难度。像 X11 一样，X13、X15、X17 都是使用固定的哈希算法串联顺序，只是改变了哈希算法的数量和种类。

这类算法很可能越来越快的被突破，ASIC 制造商只需要逐个实现每一种哈希算法，并根据需要组合它们。

曙光

X16R 算法使用不断打乱哈希算法串联顺序的方法来解决这个问题。

X16R 选用的哈希算法是 X15 中已经经过验证的 15 种，外加 SHA512。但是 16 种哈希算法的串联顺序，是基于前一个区块的哈希值动态改变的。

这种动态改变顺序的做法，并不能使设计 ASIC 成为不可能。但是，这需要 ASIC 对额外的输入做更多适配。这些操作对 CPU 和 GPU 来说，可以轻松完成。动态改变顺序的做法也可以防止 ASIC 生产商通过简单的拓展 X11 和 X15 矿机的方法，生产出 X16R 矿机。

X16R 的 16 种哈希算法的串联次序，由前一区块哈希值的最后 8 字节（十六进制表示的 16 位）决定。

涉及的哈希算法如下：

- 0=blake
- 1=bmw
- 2=groestl
- 3=jh
- 4=keccak
- 5=skein
- 6=luffa
- 7=cubehash
- 8=shavite
- 9=simd
- A=echo
- B=hamsi
- C=fugue
- D=shabal
- E=whirlpool

- F=sha512

例如：

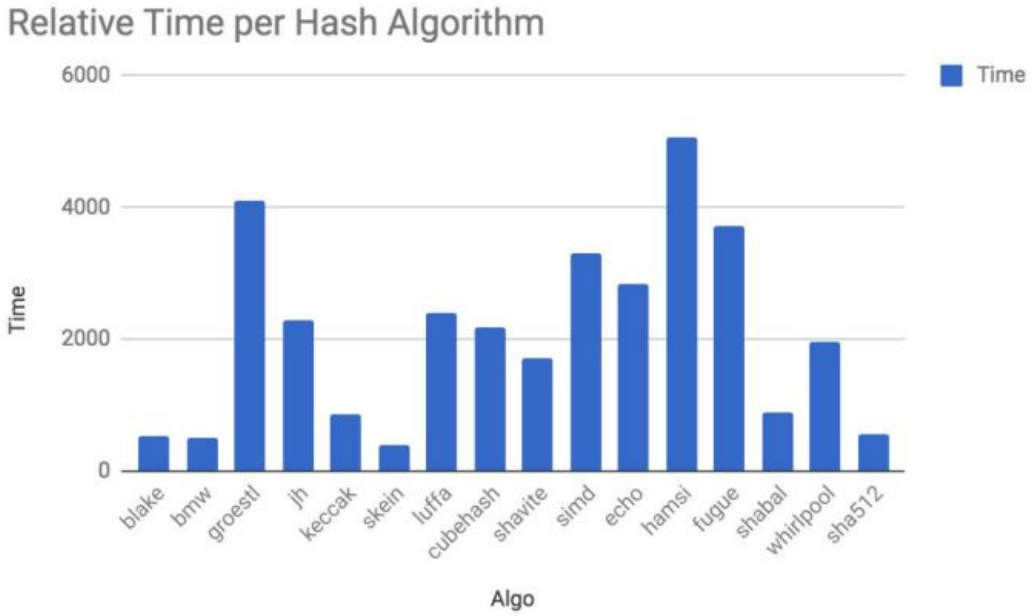
前一区块的哈希值为：

0000000000000000007e8a29f052ac2870045ae3970270f9**7da00919b8e86287**

最后 8 字节是：**0x7da00919b8e86287**

每一个十六进制位决定一种哈希算法，下一区块 X16R 的哈希算法排序为：

1. cubehash(7)
2. shabal(d)
3. echo(a)
4. blake(0)
5. blake(0)
6. simd(9)
7. bmw(1)
8. simd(9)
9. hamsi(b)
10. shavite(8)
11. whirlpool(e)
12. shavite(8)
13. luffa(6)
14. groestl(2)
15. shavite(8)
16. cubehash(7)



一些哈希算法比其他哈希算法需要更长的计算时间，如上表所示。

计算时间上的差异，将在挖掘区块时，通过算法的随机搭配而被平均。

启示

SMN（Super Master Node）是 X16R 的参考实现，定义了哈希算法的数量、种类、顺序、以及前一区块中，用于决定哈希算法顺序所使用的字节。

在比特币的基础上，SMN 修改了发行规则、区块时间和 PoW +POS算法。

X16R 的思路可以拓展到包括 Scrypt、Equihash 和其他 ASIC 抵抗算法，以确保允许任何人使用现成的空闲计算机挖矿。对每种加密货币来说，哈希算法的顺序、种类、数量都可以非常容易的改变，以阻止 ASIC 厂商像 X11 算法一样，设计出可以挖掘一类加密货币的矿机。

思考

为什么 X16R 这种动态改变哈希算法顺序的做法可以抵抗 ASIC 呢？

我们前面提到，X11 这种靠堆算法数量的做法最终被 ASIC 攻克，因为只要币价足够高，厂商就会投入人力物力去把 11 种哈希算法逐个用 ASIC 实现，最后再组合起来。

运行过程中，每个哈希算法模块，可以流水线运行。每个时刻，都可以做到 100%的芯片利用率，就像下面这样。

对 11 种哈希算法的 X11 来说，ASIC 可以提供 11 种哈希函数的电路，每时每刻，这 11 个哈希函数的电路都在工作，同时处理 11 个任务的不同部分。

而 X16R 在每一个区块进行 PoW 计算时，增加了很多挑战：

- 每种哈希算法被调用的次数不确定
- 哈希函数的排序不确定

对于预先设计好的 ASIC 矿机来说，一旦芯片流片，包含了多少种哈希函数，每种哈希函数有多少计算资源就是确定的。

对 X16R 来说，由于前一区块 Hash 值的后 8 字节，可以认为是完全随机的，我们可以计算出对每个区块，涉及哈希函数种类数和概率如下：

```
涉及哈希函数种类 概率
1 8.673617379884035e-19
2 4.263126310299903e-13
3 1.3008292880367645e-09
4 4.0680219586149147e-07
5 3.114800294252984e-05
6 0.0008548354163772504
7 0.010257933523243057
8 0.060249156768226245
9 0.1847133772998888
10 0.30522511853905976
11 0.273508450268678
12 0.13029987021900835
13 0.03130843802212624
14 0.0034140224047796153
15 0.00013610720550616406
16 1.1342267125513672e-06
```

加权平均下，每个区块运算时，平均涉及 10.3 种哈希函数，假设每种哈希函数的硬件实现需要的芯片面积相同，则芯片利用率约为 64.4%，这是平均利用率的上限，也就是说，无论何时，芯片上都会有 36%左右的电路是被浪费掉的。

同时，由于每种哈希函数在运算中被使用的次数不确定，顺序也不确定，想设计出高效的流水线就更加困难。

如果想更好的增加流水线的并行度，就要有更多冗余的运算单元，芯片利用率更低。

如果想提高芯片利用率，势必就会加剧资源争用，降低并行度，影响性能。

这一切只是增加了难度，当币价足够高时，ASIC 还是可以获得一定的优势。

然而，只要相对于 GPU，这种优势不是十分巨大，对减轻算力中心化的趋势，还是有帮助的。

集成电路其实是由晶体管构成若干门电路，门电路通过组合实现各种功能。

有没有人想到，如果有一种器件，可以根据需要，动态的组合这些门电路，来实现各种功能，不就可以每时每刻都充分利用整个芯片了吗？

FPGA 就属于这种器件。

X16R 虽然可以抵抗 ASIC，但是对 FPGA 几乎无能为力。每当上一个区块产生，下一个区块使用的哈希函数种类，每种哈希函数调用的次数，哈希函数调用的次序也就确定了。

FPGA 可以快速重新编程，为每种哈希函数分配好使用的门数量，便可以充分利用芯片资源，进行运算，但是其算力较为低下，和 GPU 相比优势不明显。

只要还有利可图、有人在乎设备平等，算法和 ASIC 矿机之间的较量就还会持续下去。