区块链技术简介

概艺区块链（Getarts.cn)

郑嘉文- 2020

区块链6层架构

技术角度

层级	功能描述	技术
数据层	区块链就是通过区块来存储数据，并以链表的形式串在一起。数据层最主要的功能是维护一个唯一的全局的账本，包括账户信息，交易信息。	主要用到传统计算机里的成熟技术，比如：不对称加/解密，键值（Key-Value）数据库，哈希函数，默克尔树，前缀树等。
网络层	采用P2P组网技术，以及相应的数据传输和验证技术，执行节点发现，节点间通信以及节点管理功能。P2P一般基于UDP协议，所以也要使用相应的网络管理和通信功能	P2P， Gossip， Kademlia DHT
共识层	实现共识算法，让系统中的各个节点就区块数据的有效性达成共识。为分布式系统提供统一的输出，尽量处理节点恶意或者故障的情况。	目前流行的有几十种共识算法，比较知名的有：工作量证明（PoW），权益证明（PoS），委托权益证明（DPoS）等。
激励层	通过提供激励机制，鼓励节点参与记账，而不要参与作恶。	通证经济模型，社会学，博弈论，概率
智能合约层	可以视为商业逻辑的中间层	Solidity，Web Assembly（WASM），脚本语言， Serpeng，LLL等
应用层	各种区块链应用和案例	Web3， gRPC，Restful API

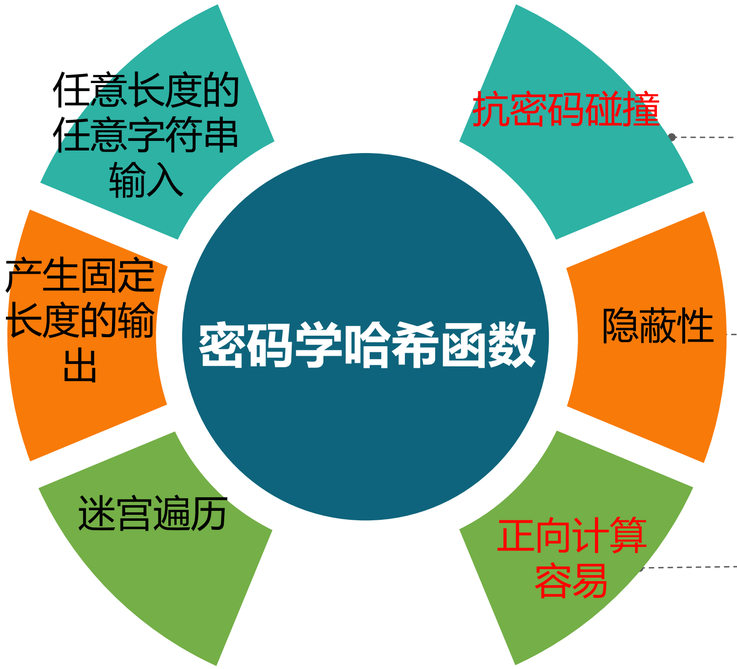
区块链4层架构

应用角度

层级	名字	功能
第一层	分布式账本应用	提供多方计算，多方存储的去中心化账本，作为信任机器的基础。来增强可信度，提高写作效率
第二层	价值传输网络	实现价值传输功能；可清算，可追溯，安全
第三层	通证激励体系	提供激励机制，充分调动关联各方参与而不作恶，优化资源分配
第四层	资产数字化	证券通证化，通证证券化，有价资产数字化，无价资产数字化

区块链密码学哈希函数

特性



不同的消息不会有同样的哈希值

很难从已知的哈希值反向计算出原文

很容易来计算任意数据的哈希值

雪崩效应是指输入数据任何的微小变化都会引起其哈希值的巨大变化。

区块链密码学算法

对称加密

对称加密

即加密和解密使用相同密钥的算法（加密Key=解密key）。其优点在于加密速度快，便于硬件实现和大规模生产通常，而且通常，对称加密算法比较简便高效，密钥简短，破译极其困难。其缺点在于需要保障密钥安全；无法用来签名和抗抵赖；

DES(Data Encryption Standard)

DES是最基本的对称加密算法，也是使用频率最高的一种算法，加密密钥与解密密钥相同。DES出身比较好，出自IBM之手，后被美国军方采纳，之后便广泛流传，但是近些年使用越来越少，因为DES使用56位密钥，以现代计算能力，24小时内即可被破解

PBE(Password-based encryption)

PBE算法（Password Based Encryption，基于口令加密）是一种基于口令的加密算法，其特点是使用口令代替了密钥，而口令由用户自己掌管，采用随机数杂凑多重加密等方法保证数据的安全性。PBE算法在加密过程中并不是直接使用口令来加密，而是加密的密钥由口令生成

AES(又称Rijndael加密法)

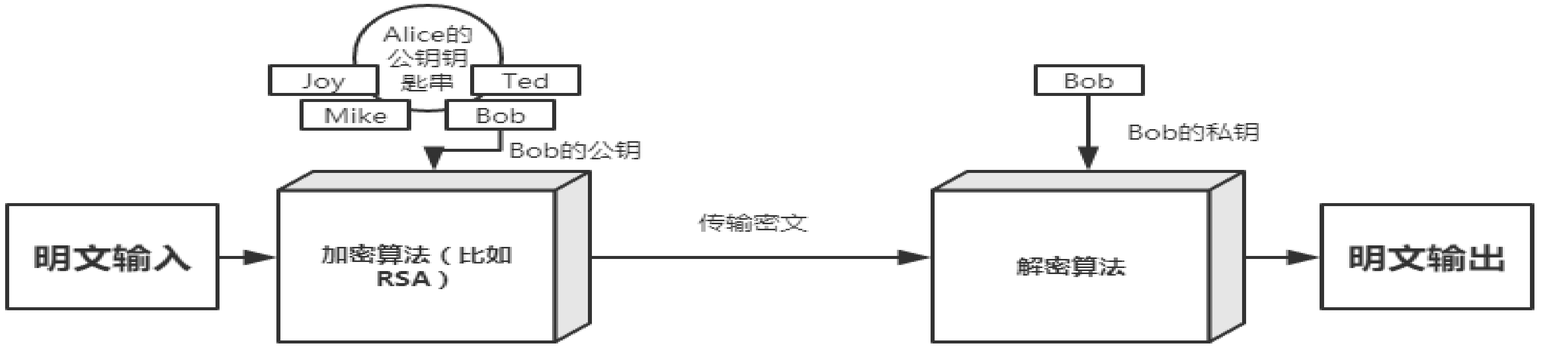
高级加密标准（英语：Advanced Encryption Standard，缩写：AES），在密码学中又称Rijndael加密法，是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的DES，已经被多方分析且广为全世界所使用。

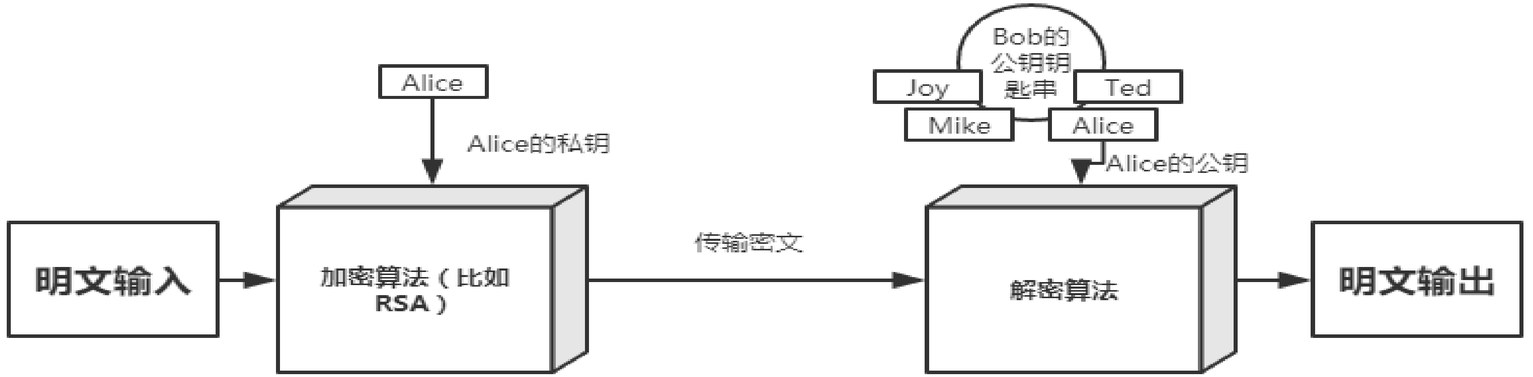
RC4（Rivest Cipher 4的首字母缩写）

RC4于1987年提出，和DES算法一样。是一种对称加密算法，也就是说使用的密钥为单钥（或称为私钥）。但不同于DES的是。RC4不是对明文进行分组处理，而是字节流的方式依次加密明文中的每个字节。解密的时候也是依次对密文中的每个字节进行解密

区块链密码学算法

不对称加密





区块链密码学算法

国密

算法名称	类型	简介	补充说明
SM1	对称加密	其加密强度与AES相当。该算法不公开，调用该算法时，需要通过加密芯片的接口进行调用。	由于SM1、SM4加解密的分组大小为128bit，故对消息进行加解密时，若消息长度过长，需要进行分组，要消息长度不足，则要进行填充。
SM2	非对称加密，基于ECC	该算法已公开。由于该算法基于ECC，故其签名速度与秘钥生成速度都快于RSA。ECC 256位（SM2采用的就是ECC 256位的一种）安全强度比RSA 2048位高，但运算速度快于RSA。 国家密码管理局公布的公钥算法，其加密强度为256位	公钥密码算法(RSA和SM2)、公开加密算法本身和公开公钥，保存私钥
SM3	非对称加密	消息摘要。可以用MD5作为对比理解。该算法已公开。校验结果为256位。	摘要算法(SM3，MD5)用于数字签名，消息认证，数据完整性，但是SM3安全度比MD5高
SM4	对称加密	无线局域网标准的分组数据算法，密钥长度和分组长度均为128位。	分组密码算法(DES和SM4)、将明文数据按固定长度进行分组，然后在同一密钥控制下逐组进行加密

区块链密码学算法

RSA算法

加密过程

1) 选两个很大的质数p，q
2) 计算n=p*q,且公开n
3) 再选一个数字e，需要保证e和（p-1）(q-1)没有除1之外的公因数。数字e和（p-1）(q-1)同样是公开的。
4) 如需加密信息X,计算X^e mod n

样例

1) 我们来假定p=11,q=17
2) n=11*17=187
3) 选择e=3，满足与160没有除1之外的公因数。
4) 假如我要加密的信息为X=MATH,，每个字母以它在字母表中的顺序赋值（A=1，B=2，C=3, 以此类推）
5) 于是我要加密信息就是M=13,A=1,T=20,H=8，加密后，

13^3 mod 187 =140

1^3 mod 187 =1

20^3 mod 187 =146

8^3 mod 187 =138

发出去的信息就成为了(140, 1, 146, 138)。

解密过程

1) 找到一个数字d，满足d*e=1 mod(p1-1)(q-1)
2) 解密信息Y,需要计算Yd,计算Yd mod n

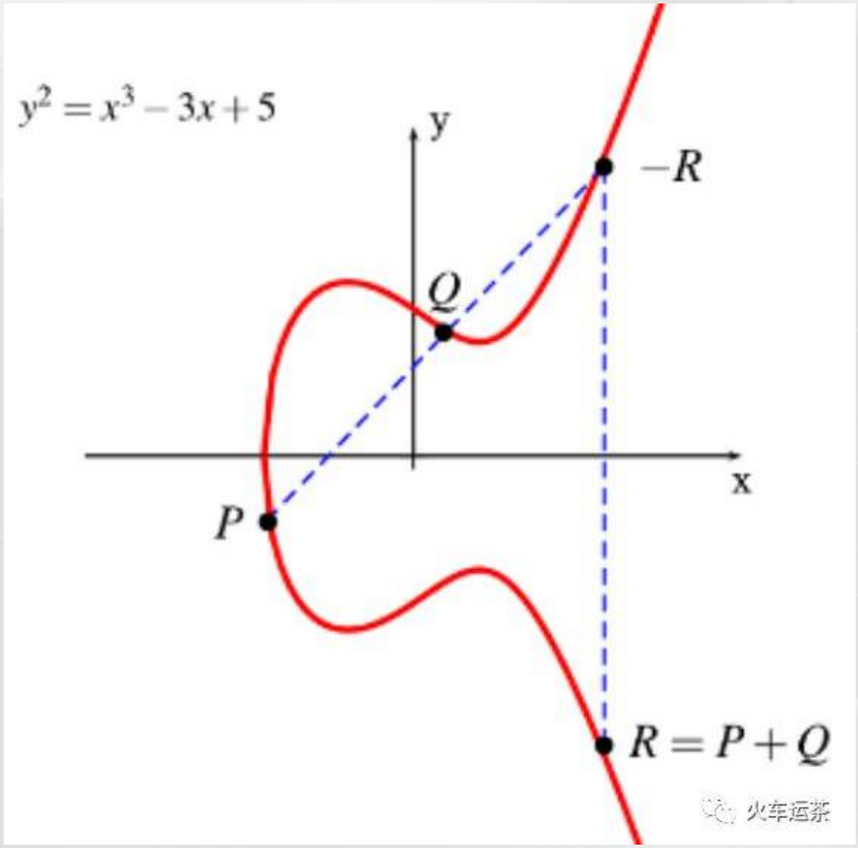
样例

解密上边发送过来的信息中的第一个数字，140。

1) 选择d=107，满足的d*e=321满足1 mod 160
2) Ydmod n=140107 mod 187=13.这样就得到发送的字母M
3) 用同样的方法可以最后得到13,1,20,8，对应字母表中的M,A,T,H。解密就完成了。

区块链密码学算法

魏尔斯特拉斯曲线



特点

- 曲线x轴对称；
- 一根直线跟一条椭圆曲线最多有三个交点

椭圆曲线

包括国际标准的NIST P-256，中国标准的SM2，比特币以及以太坊使用的secp256k1

y2 = x3 + ax + b； 注意需满足：$4a^3+27b^2 \neq 0$

无限点

这个无限点是椭圆曲线的初始点，就如同我们二维坐标系里的原点（x=0, y=0）

椭圆曲线加法

如果p1和p2都位于椭圆曲线上，那么连接p1和p2,与椭圆曲线的交点就是p3.

椭圆曲线加法满足交换律

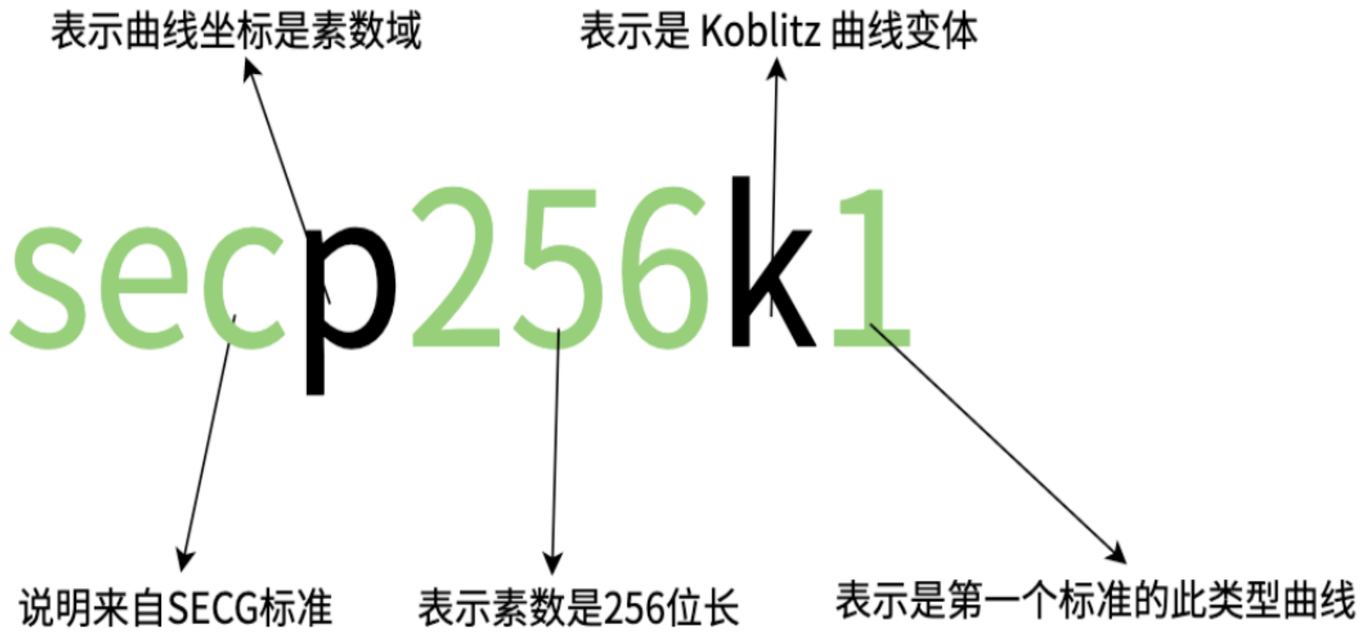
A + (B + C) = (A + B) + C= A + B + C

椭圆曲线乘法

k * G = G + G + ... + G (一共k个G)

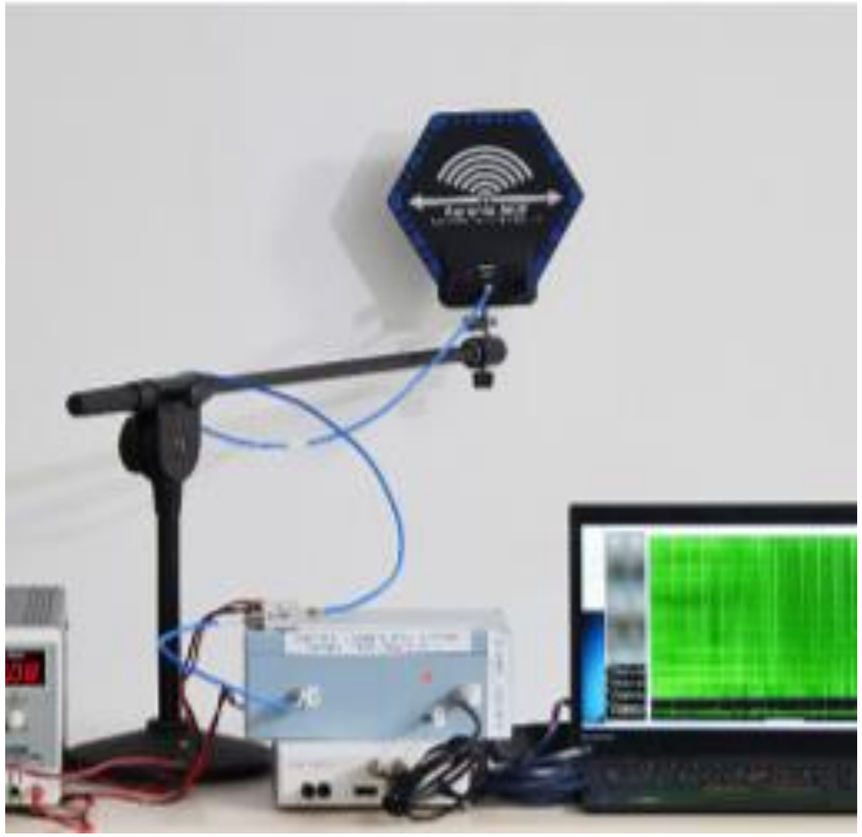
区块链密码学算法

比特币网络里使用的算法secp256k1



区块链密码学算法

Your great subtitle in this line



蒙哥马利曲线(ECSD25519)

伯恩斯坦发明的25519曲线。25519这个名字实际上是一个素数：2^{255} - 19
方程式为

$$y^2 = x^3 + 486662x^2 + x$$

爱德华兹曲线

448曲线得名于它使用的一个448位的素数：$p = 2^{448} – 2^{224} – 1$。又名“金发姑娘”（Goldilocks）。发明者说这是因为选择的素数在形式上具有“黄金比例（golden ratio）”。金发姑娘用的曲线名为爱德华兹曲线，方程式为

$$y^2 + x^2 = 1 - 39081x^2y^2$$

区块链共识算法

FLP & CAP定理

FLP定理

Fischer, Lynch 和 Paterson 在论文 "Impossibility of distributed consensus with one faulty process" (1985)证明, 在一个异步系统里面, 只要有一个错误的进程, 一致性就不可能达成. 这个也就是著名的"FLP"结论

CAP定理

分布式计算系统不可能同时确保一致性、可用性和分区容错性。这三者不可兼得。

- 一致性（Consistency）：所有节点在同一时刻能够看到同样的数据，即“强一致性”。
- 可用性（Availability）：确保每个请求都可以收到确定其是否成功的响应。
- 分区容错性（Partition Tolerance）：因为网络故障导致的系统分区不影响系统正常运行。

区块链技术 – 博弈论

目的

激励机制（Incentive）

比如，在拜占庭将军问题中给忠诚的将军以奖励。当背叛的将军发现背叛行为没有任何收益的时候，他们还有背叛的动机吗？这里我们引进了博弈论的概念：我们不再把节点或者说将军分成公正/恶意（忠诚/背叛）两方，我们认为每一个节点的行为是由激励机制决定了。就如2千年之前中国诸子百家热烈争论的话题：人之初，性本善焉，性本恶焉？我们认为人之初，性无善无恶。性的善恶由后天的激励机制决定。如果激励机制设置得当，考虑到每个节点都有最大化自己利益的倾向，大部分的节点都会遵守规则，成为公正的节点。

随机性（Randomness）

在拜占庭将军问题中，决定下一个行动需要将军们协调一致，确定统一的下一步计划。在存在背叛将军的条件下，忠诚的将军的判断可能被误导。在传统的中心化的系统中，由权威性大的将军作决定。比如现实世界里的政府，银行。在去中心化的系统中，研究者提出一种设想：是否可能在所有的将军中，随机的指定一名将军作决定呢？这个有点异想天开的设想为解决拜占庭将军问题打开了一扇门。根据什么规则制定作决定的将军呢？对应到金融系统里，就是如何决定谁有记账权。

1.**根据每个节点的计算力（Computing Power）来决定。** 谁的计算力强，解开某个谜题，就可以获得记账权（在拜占庭将军问题里是指挥权）。这是比特币里用的PoW共识协议。
2.**根据每个节点具有的资源（Stake）来决定。** 所用到的资源不能被垄断。谁投入的资源多，谁就可以获得记账权。这是PoS共识协议。

。

区块链共识算法

现有应用情况

共识算法	应用
PoW	比特币，莱特币，以太坊前三个阶段：即Frontier（前沿）、Homestead（家园）、Metropolis（大都会）
PoS	PeerCoin， NXT， 以太坊的第四个阶段，即Serenity（宁静）
DPos	BitShare，EOS
Paxos	Google Chubby， ZooKeeper
PBFT	Hyperledger Fabric
Raft	Etcd

区块链共识算法

工作量证明（Proof of Work， PoW）

算法描述

工作量证明，就是矿工们竞争做数学题。第一个解开数学谜题的就会获得奖励：一些代币。每一个节点就会更新新的块。每个矿工都会想赢得下一块的打包权，因而被激励去不停的解题。这就达成了一个全网的共识。

最长链原则

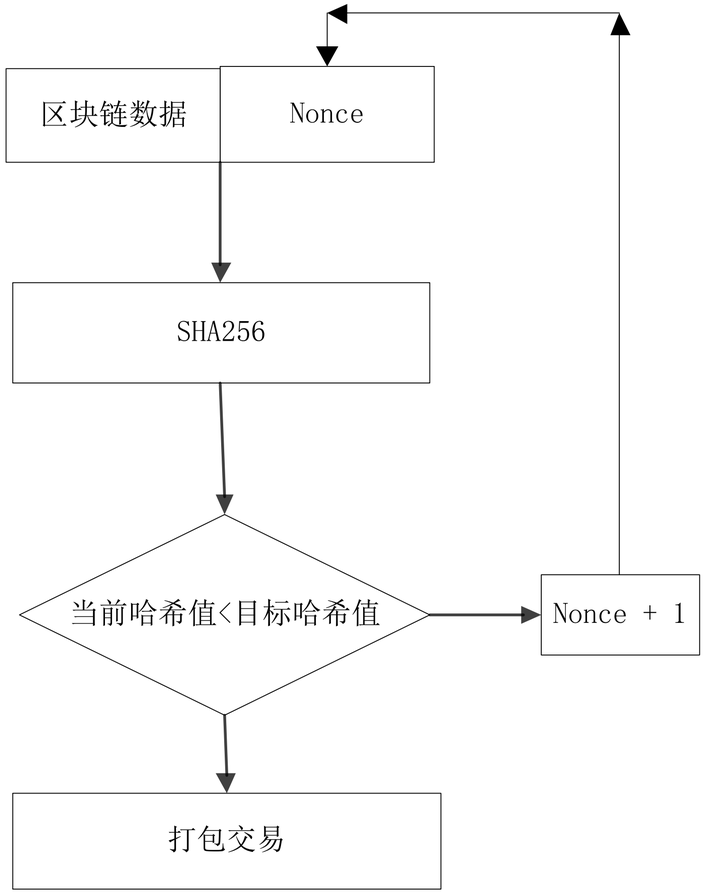
主要的统计方法两个主要两个主要进行数据分析可以自己替换。

优点

完全去中心化（任何人都可以加入）；节点自由进出，容易实现；破坏系统花费的成本巨大；

缺点

对节点的性能网络环境要求高；浪费资源；



区块链共识算法

权益证明（Proof of Stake， PoS）

股权权益证明（Proof of Stack 简称PoS）现在已经有了很多变种。最基本的概念就是选择生成新的区块的机会应和股权的大小成比例。股权可以是投入的资金，也可以是预先投入的其他资源。

PoS运作的机制大致如下

- 加入PoS机制的都是持币人，成为验证者（Validator）
- PoS算法在这些验证者里挑一个给予权利生成新的区块。挑选的顺序依据于持币的多少
- 如果在一定时间内，没有生成区块，POS则挑选下一个验证者，给予生成新区块的权利

以此类推以区块链中最长的链为准

PoS不需要为了生成新区块而大量的消耗电力。也一定程度的缩短了共识达成的时间。但是，缺点是PoS还是需要挖矿。

区块链共识算法

委托权益证明（Delegate Proof of Stake， DPoS）

委托权益人证明机制（Delegated Proof of Stake，DPOS）是PoS算法的改进。DPoS算法中使用见证人机制（witness）解决中心化问题。总共有N个见证人对区块进行签名。DPoS消除了交易需要等待一定数量区块被非信任节点验证的时间消耗。通过减少确认的要求，DPoS算法大大提高了交易的速度。

见证人机制

- 权益所有人为了见证人尽量长时间的在线，要付给见证人一定的报酬。
- 见证人必须保证尽量在线。如果见证人错过了签署区块链，就要被踢出董　　事会。不能担任见证人的工作。
- 如果权益所有人不喜欢选出来的见证人，可以选择卖出权益退场。

见证人机制有以下特点：

- 见证人的数量由权益所有者确定，至少需要确保11个见证人。
- 见证人必须尽量长时间的在线，以便做出响应。
- 见证人代表权益所有人签署和广播新的区块链。
- 见证人如果无法签署区块链，就将失去资格，也将失去这一部分的收入。
- 见证人无法签署无效的交易，因为交易需要所有见证人都确认。

区块链技术 – 博弈论

纳什均衡（Nash Equilibrium）

假设有n个人参与博弈，每个人选择自己的最优战略，所有参与人选择的战略一起构成一个战略组合。纳什均衡（Nash Equilibrium）指的是这样一种战略组合，它由所有参与人的最优战略组成。也就是说，给定别人战略的情况下，没有任何单个参与人有积极性选择其他战略，也就是说没有任何人有积极性打破这种均衡

区块链技术 – 博弈论

公地悲剧

这一问题最早由美国环境学家**格雷特·哈定**(Garrett Hardin, 1915-2003)在1968年发表在《科学》杂志上的论文《**公地的悲剧**》指出。公地悲剧问题，是一个纳什均衡。公地悲剧问题说的是，有一块公共的草地，每个人都能上去放羊，如果所有人都不加节制地放牧，很快这地就被啃光了，再也长不出草来了，大家就失去了一块草地。但是在相互竞争的情况下，每个牧羊人都出于自身利益最大化的动机，选择不加节制的放牧。在现代工业社会，把草地换成湖泊、河流，把牧羊人换成工厂企业家，把放牧换成排放废料、污水，这个问题是等价的。对于公共资源的使用，都存在公地悲剧问题，

区块链技术 – 博弈论

囚徒困境

	A		
B		沉默	揭发
	沉默	（1，1）	（0，10）
	揭发	（10，0）	（8，8）

1)如果两个人都不招供，那么两人将被判入狱1年。

2)如果两个中的一个揭发一个不揭发，那么揭发的人将因立功表现而被释放，而另一方获得10年刑期。

3)如果两个都揭发那么两个都会得到8年刑期。

区块链技术 – 博弈论

在区块链里的应用

矿工在区块链系统中拥有很大的权力，如果他们选择为自己的个人利益作弊，他们可能会对系统造成严重破坏。为了解决这个问题，区块链使用博弈理论机制来保持系统的健壮性。

挖矿中的纳什均衡与惩罚制度

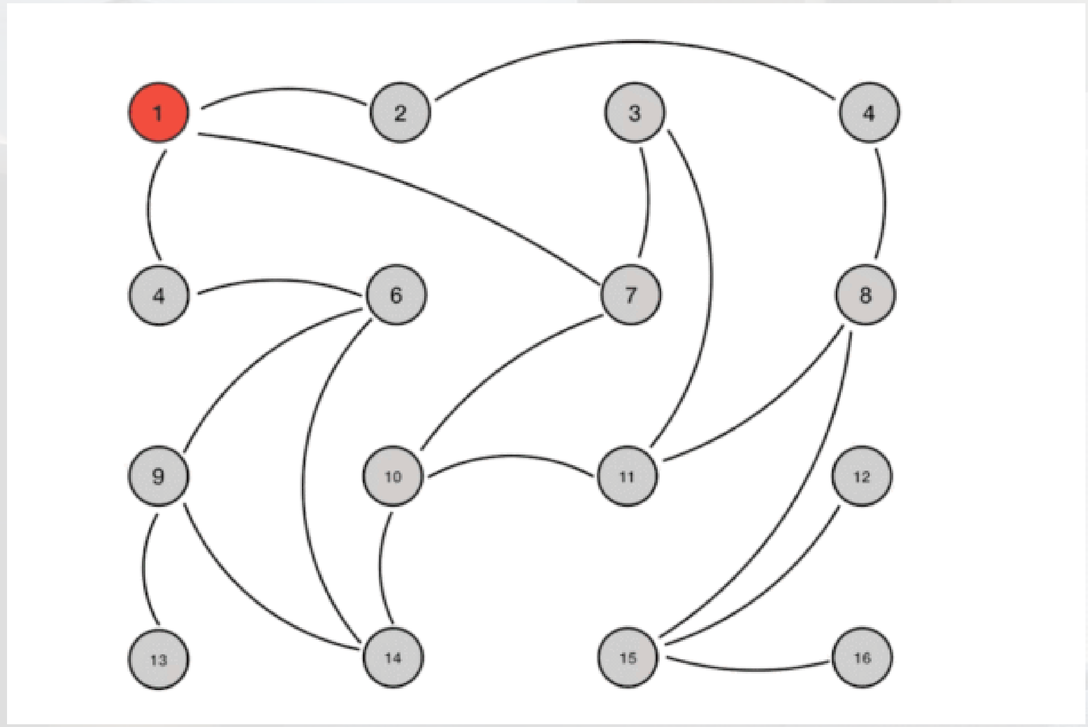
如果矿工创建了无效区块
那么由于区块链机制中已定义的规则，其他人不会在其上挖掘。在无效块之上挖掘的任何块都将变为无效块。使用此规则，矿工将简单地忽略无效区块并继续在主链顶部挖掘。

类似的逻辑也可以适用于“次长”，或者“第n长”（n不等于1）的分叉链
由于比特币认的是最长的链，在任何非最长链上挖矿都是无效且无用的。所以，矿工也会在最长链上挖矿，而不会工作在“次长”，或者“第n长”（n不等于1）的分叉链。因为矿工们作为一个群体将选择最稳定的状态，即具有纳什均衡的状态。显然，你可以让所有的矿工在分叉链上开采并使其成为新的主链，然而，矿工的数量是如此巨大，以至于这样的事件根本无法协调。正如协调问题博弈所述，如果群体中的大多数人没有改变他们的状态，少数人将没有任何动力留在分叉的状态。如果明知这是徒劳的，为什么一个矿工会浪费他们的计算能力和冒巨大风险?

。

区块链P2P算法

Gossip算法



- 节点 A 周期性的选择相邻的 k 个节点，并且向这 K 个节点发送自身存储的数据；
- K 个节点接收到 A 发送过来的数据后，发现自身没有则存储下来，如果有则丢掉，并且重复节点 A 的过程。

Gossip算法

Gossip 协议也叫 Epidemic Protocol （流行病协议），实际上它还有很多别名，比如：“流言算法”、“疫情传播算法”等。Gossip协议的主要用途就是信息传播和扩散：即把一些发生的事件传播到全世界。它们也被用于数据库复制，信息扩散，集群成员身份确认，故障探测等

可扩展性（Scalable）

gossip 协议是可扩展的，一般需要 O(logN) 轮就可以将信息传播到所有的节点，其中 N 代表节点的个数。

容错（Fault-tolerance）

网络中任何节点的重启或者宕机都不会影响 gossip 协议的运行

健壮性（Robust）

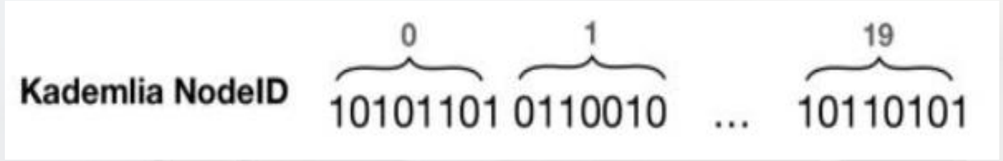
任何节点都可以随时加入或离开，而不会影响系统的整体服务质量（QOS）

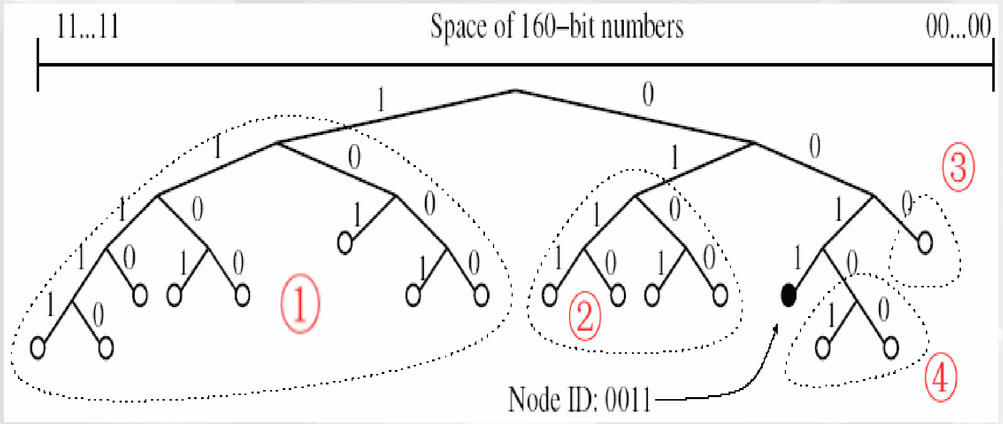
最终一致性（Convergent consistency）

Gossip 协议实现信息指数级的快速传播，因此在有新信息需要传播时，消息可以快速地发送到全局节点，在有限的时间内能够做到所有节点都拥有最新的数据

区块链P2P算法

Kademlia算法





	0	0	1	1	基础节点
1	1	X	X	X	前缀为1的所有子树
2	0	1	X	X	前缀为01的所有子树
3	0	0	1	X	前缀为001的所有子树
4	0	0	1	0	前缀为0010的所有子树

Kademlia

Kademlia 协议是美国纽约大学的 P. Maymounkov 和 D. Mazieres 在2002年发布的一项研究结果 Kademlia: A peer-to-peer information system based on the XOR metric。

Node ID

Kademlia标准的Node ID是160位，也就是20个字节

K

Kademlia协议对每个桶内维护的节点数设置了一个上限K，一旦桶内节点数超过，便根据淘汰算法进行更新。

α

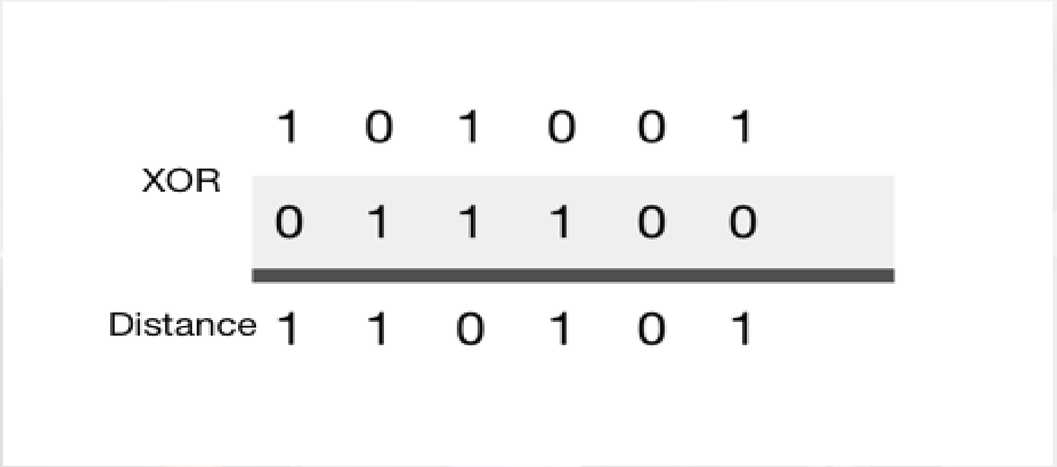
每次向其他node请求查找某个node时，会向α个node发出请求

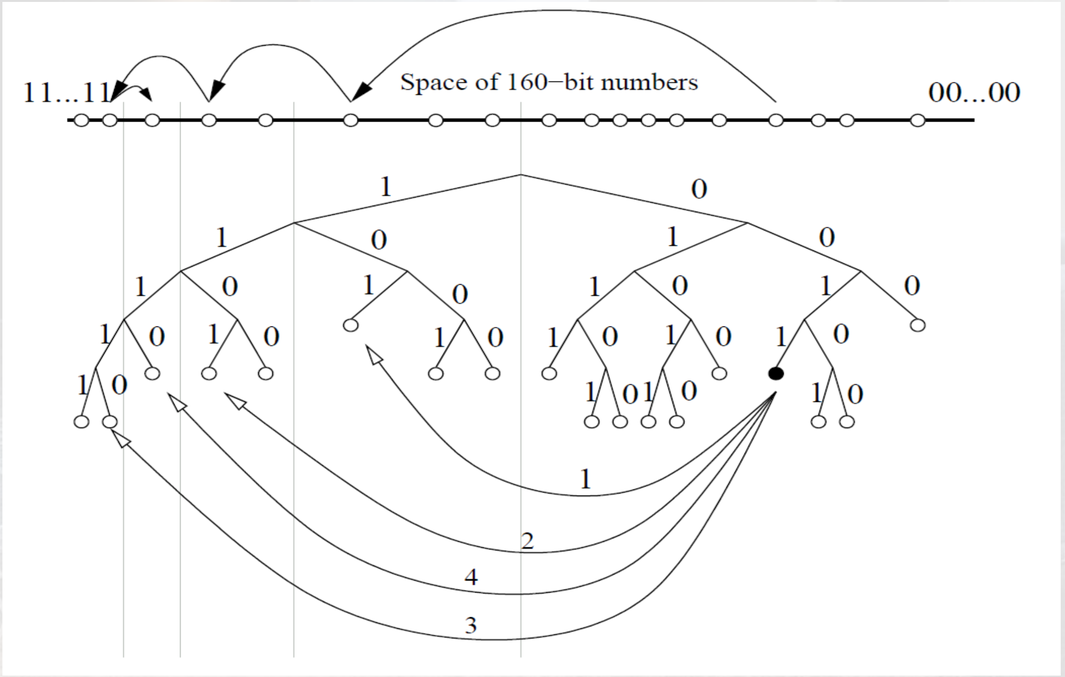
节点状态

Kademlia 中，根据当前节点的 Node ID 与它保存的其他 peer 节点 Node ID 的匹配的最多的前缀 bit 个数来构建一颗二叉树（Binary Tree），这里前缀匹配的 bit 数也叫 LCP（Longest Common Prefix）

区块链P2P算法

Kademlia算法





Kademlia的路由查询

节点 x 要查找 ID 值为 t 的节点

XOR距离

这个无限点是椭圆曲线的初始点，就如同我们二维坐标系里的原点（x=0, y=0）

查询过程

1) 计算到 t 的距离： $d(x,y)=x \oplus y$

2) 从 x 的第 [logd] 个 K 桶中取出 α 个节点的信息（“[” “]”是取整符号），同时进行 FIND_NODE 操作。如果这个 K 桶中的信息少于α个，则从附近多个桶中选择距离最接近 d 的总共α个节点。

3) 对接受到查询操作的每个节点，如果发现自己就是 t，则回答自己是最接近 t 的；否则测量自己和 t 的距离，并从自己对应的 K 桶中选择α个节点的信息给 x。

4) X 对新接受到的每个节点都再次执行 FIND_NODE 操作，此过程不断重复执行，直到每一个分支都有节点响应自己是最接近 t 的。

5) 通过上述查找操作，x 得到了 k 个最接近 t 的节点信息。

数据结构

默克尔树线性简化图

第一页：

Stamp1=
Hash(Content1)

第二页：

Stamp2=
Hash(Stamp1+
Content2)

第三页：

Stamp3=
Hash(Content3+
Stamp2)

...

...

第n页：

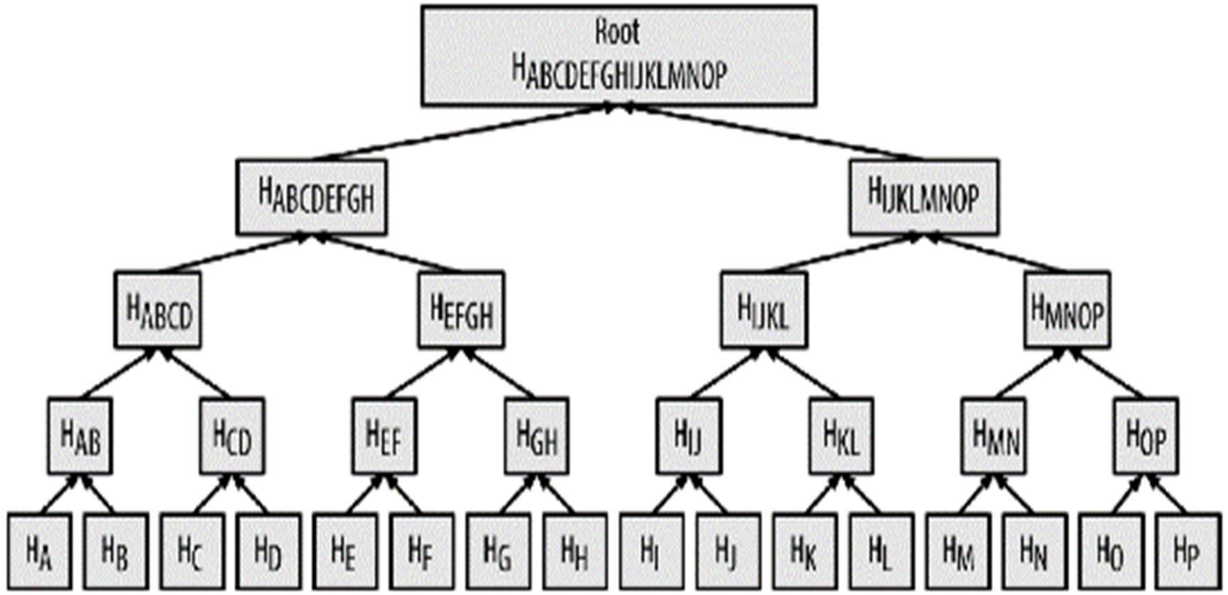
StampN=
Hash(ContentN
+StampN-1)

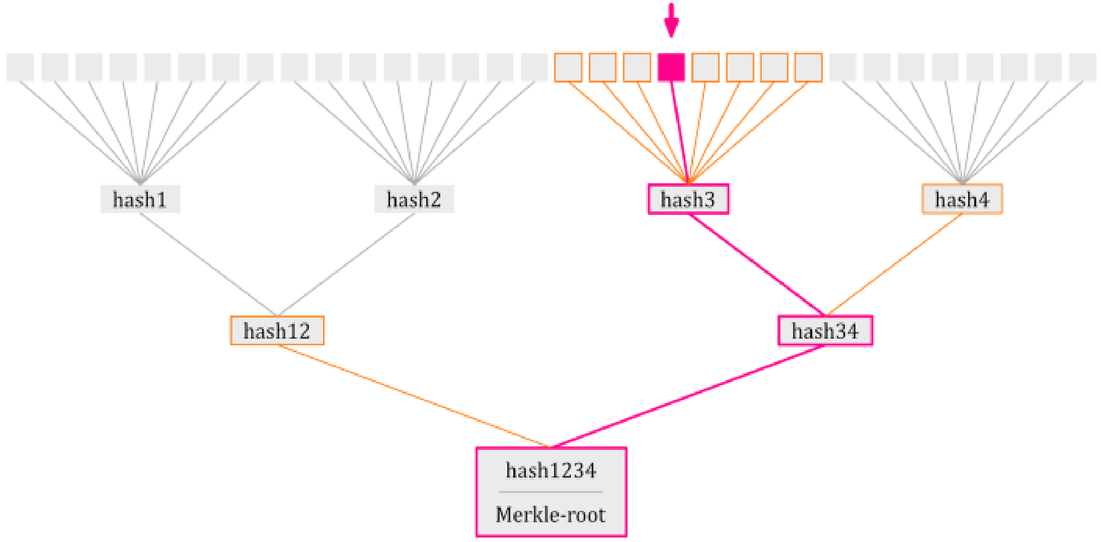
知道内容是否被篡改

知道第几页的内容被篡改

区块链关键数据结构

默克尔树





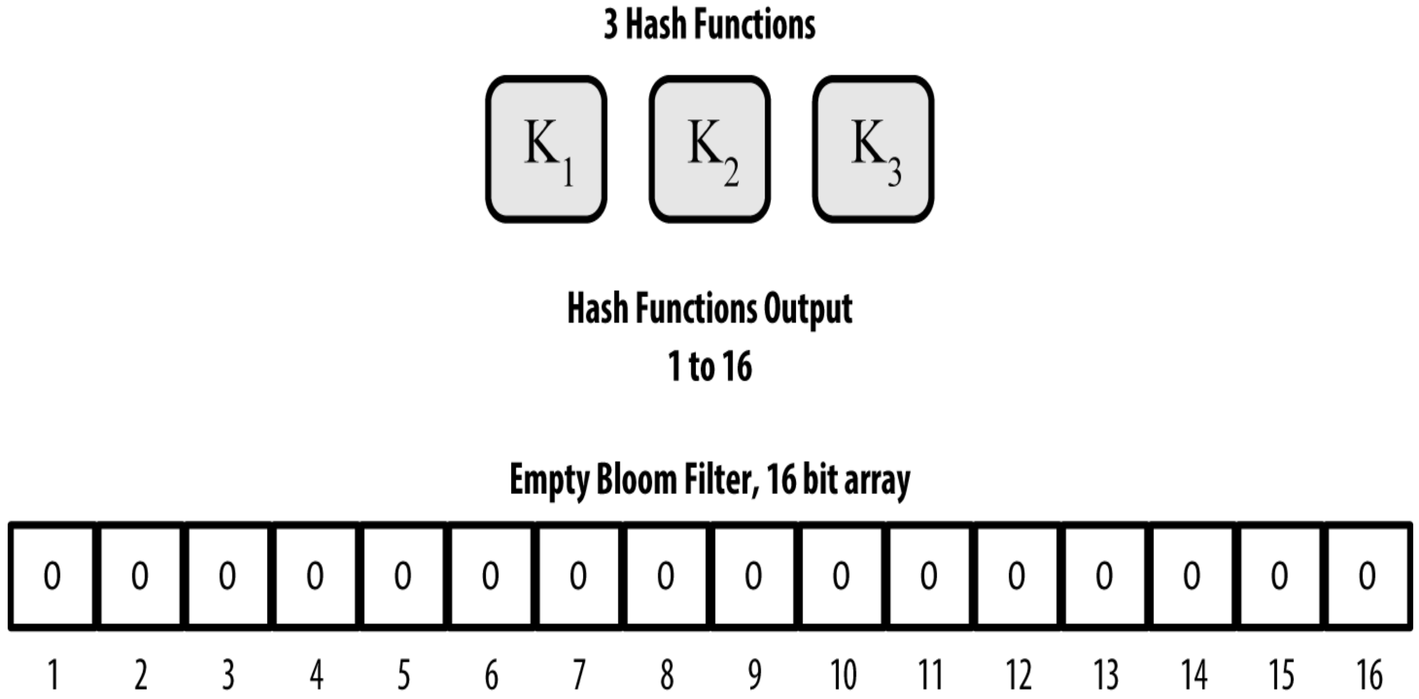
- 在每一页上我们都先盖上一个数字签章：H_A，H_B... H_P.
- $H_{AB}=Hash(H_A，H_B)$，$H_{ABCD}=Hash(H_{AB}，H_{CD})$，以此类推。

原理

任何一个叶子节点的内容被篡改了，那么从那个叶子节点开始到根的那一条线的哈希值都会不符合原先的数字签章。比如H_B值变了（也就是B页内容被篡改了），那么H_{AB}，H_{ABCD}，$H_{ABCDEFGH}$，$H_{ABCDEFGHIJKLMNOP}$都会不符合原先的数字签章

区块链关键数据结构

布隆过滤器（Bloom Filter）



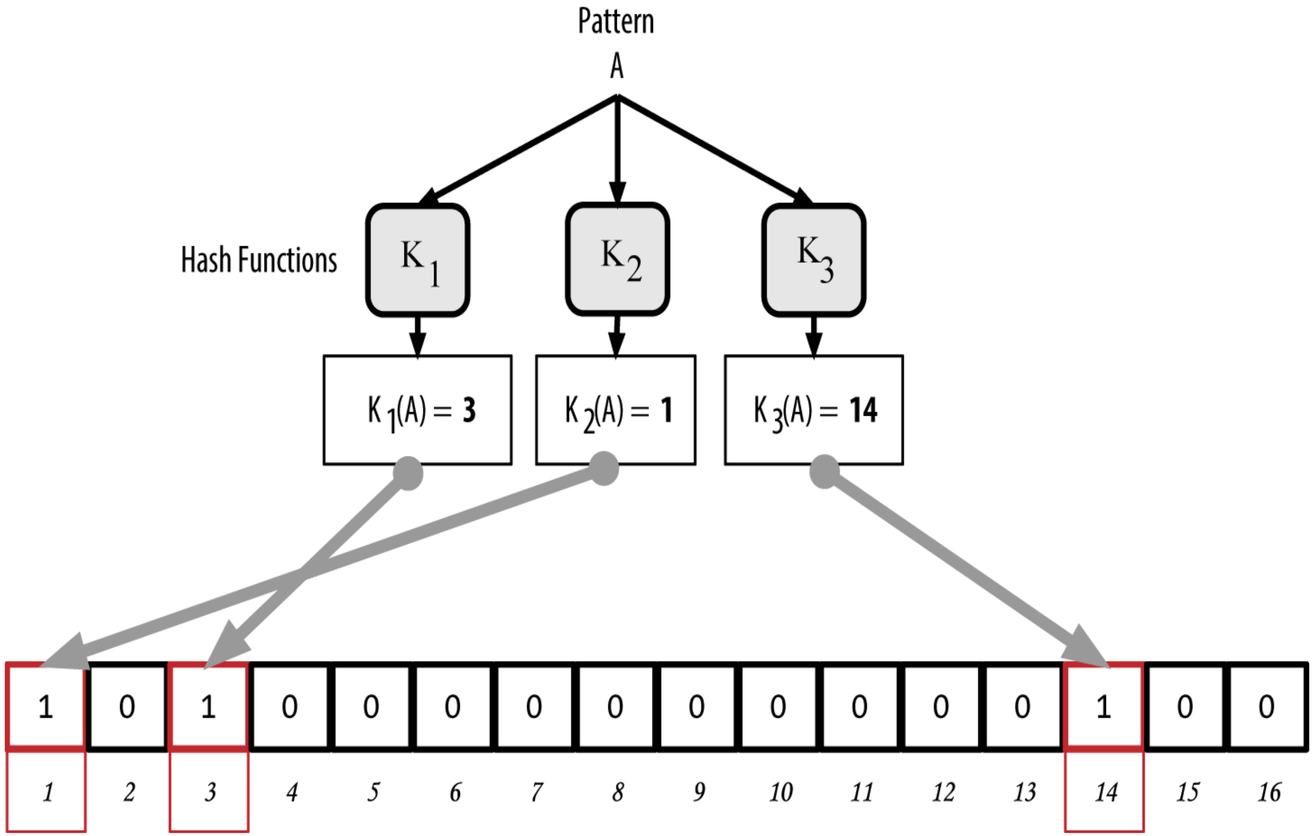
空的布隆过滤器，本例中M=3，N=16

布隆过滤器是由巴顿.布隆于一九七零年提出的。它实际上是一个很长的二进制向量和一系列随机映射函数

- M个哈希函数K，而且哈希函数K的输出范围是1到N
- N个二进制位的可变大小的数组ARR
- M和N都是可以调的

数据结构

布隆过滤器



- 每当想向集合加入一个元素A的时候，需要计算K1（A），K2（A），K3（A）。所有哈希函数K的输出是1-16之间。
- 找到ARR数组里的K1（A），K2（A），K3（A）位置，将其设为1。即

  ARR[K1（A）] = 1

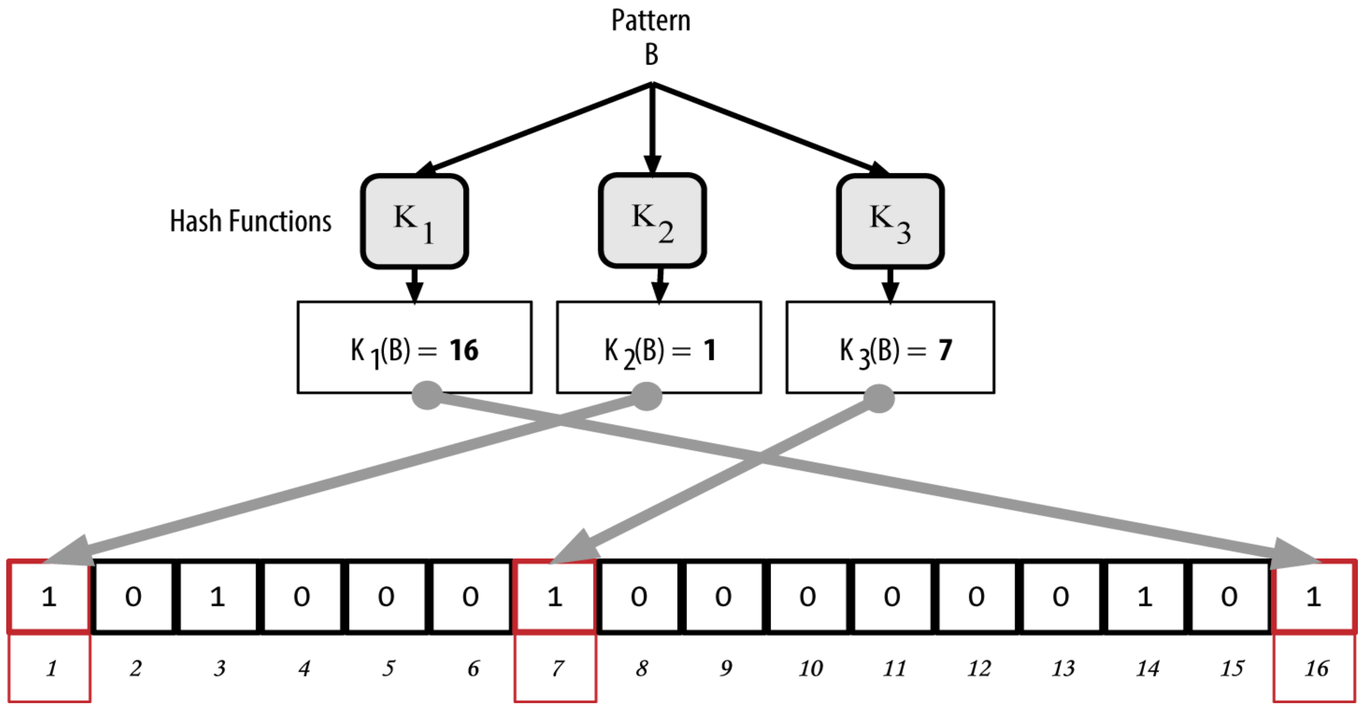
  ARR[K2（A）] = 1

  ARR[K3（A）] = 1

本例中假设K1（A）=3，K2（A）=1，K3（A）=14

数据结构

布隆过滤器



- 向集合加入一个元素B的时候，需要计算K1（B），K2（B），K3（B）。所有哈希函数K的输出是1-16之间。
- 找到ARR数组里的K1（B），K2（B），K3（B）位置，将其设为1。即

  ARR[K1（B）] = 1

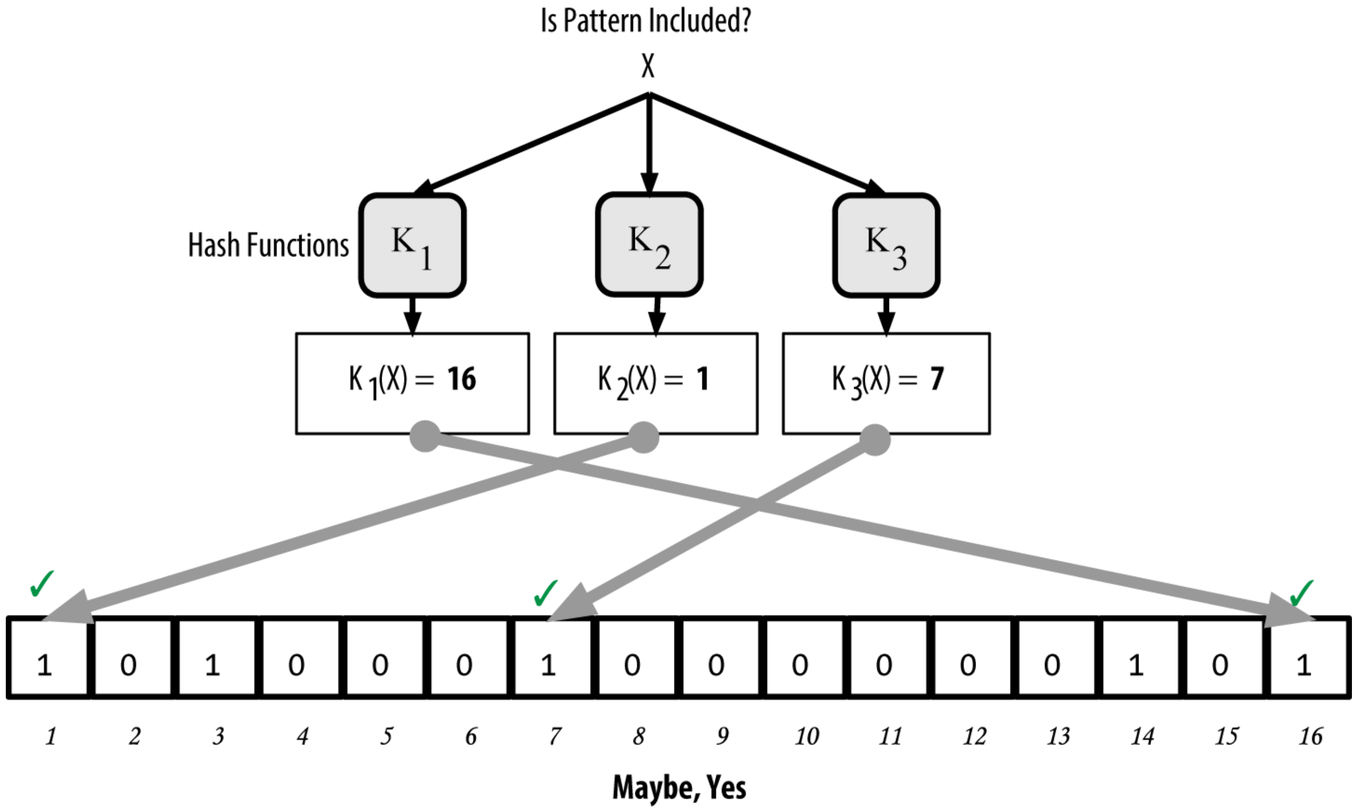
  ARR[K2（B）] = 1

  ARR[K3（B）] = 1

本例中假设K1（B）=16，K2（B）=1，K3（B）=7

数据结构

布隆过滤器



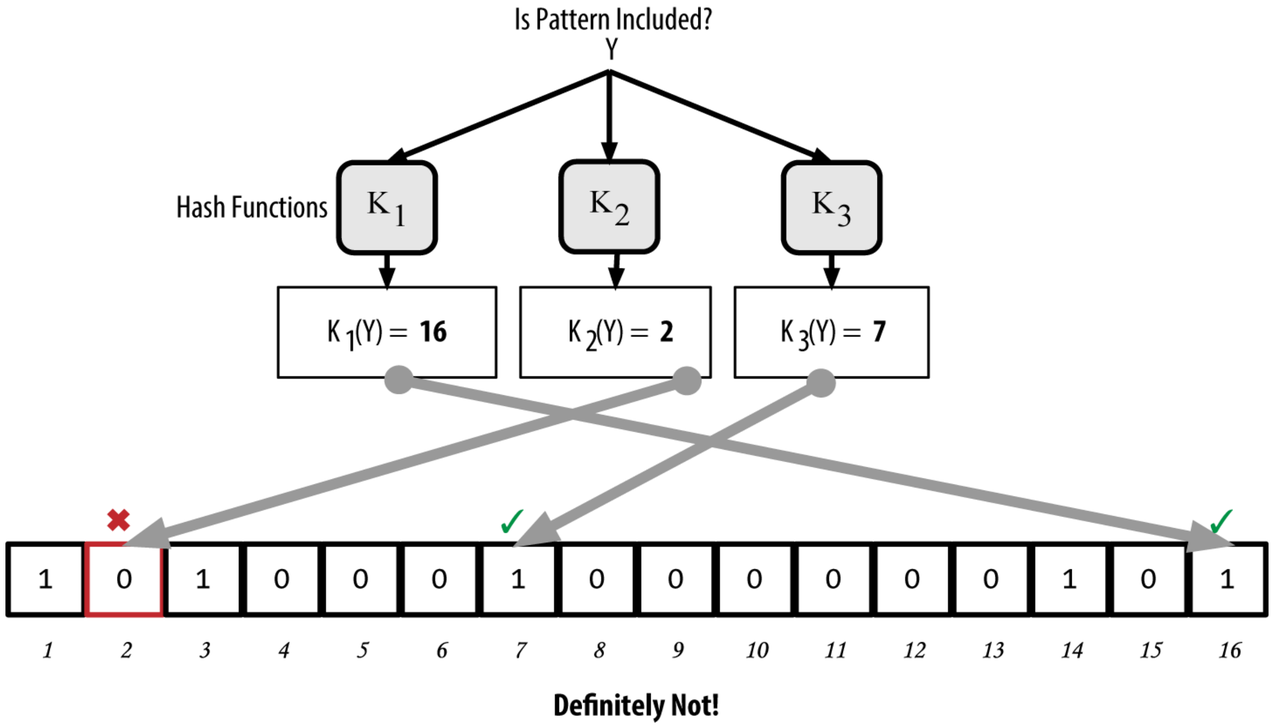
查询集合是否包含一个元素X:

- 计算K1（X），K2（X），K3（X）
- 找到ARR里相应的位置，如果都为1，则表明该集合包含该元素

本例演示集合包含X元素

数据结构

布隆过滤器



查询集合是否包含一个元素Y:

- 计算K1（Y），K2（Y），K3（Y）
- 找到ARR里相应的位置，只要有一个为0，则表明该集合不包含该元素

本例演示集合不包含Y元素

既见君子，云胡不喜

电话 **13240946967**
邮箱 zy731@hotmail.com
微信 **gavinzheng731**
博客**https://my.oschina.net/gavinzheng731/**

谢谢聆听

Q&A

郑嘉文