比特币中有趣的数据结构与特有问题

前言

现如今，“比特币”这个名词已经不再是小众圈子的专有交流话语了，随着媒体新闻的不断报道，随着越来越多的书籍文章的推出，很多技术圈以及非技术圈的朋友，都开始主动或被动的面对这个既熟悉又陌生的概念（老鸟技术玩家不在此列^_^），而随着比特币相关概念铺天盖地的炸出来的，还有各项新名词的诞生，比如区块链，哈希计算，比特币钱包，梅克尔树，分叉，侧链，智能合约，等等等等，让人眼花缭乱，头脑发晕，比起普通互联网应用的一些较为形象的概念比如网站，论坛，聊天室，电子商务等，这些技术概念明显感觉更抽象，甚至有些玄幻。

然而，我们往往都知道，一门技术如果要真的发挥好的作用，为人们的生活带来实际的便利和促进，就不能总是停留在小众的圈子里，更不能搞得神神秘秘，甚至是误解。笔者时常看到身边一些朋友对于比特币及相关技术或是满心疑惑，或是不以为然，或是极端追捧，这对于比特币及其相关技术的推广应用是不利的。

私以为，需要站在一个客观的角度，结合我们日常理解的通俗概念，来捋一捋这里面的关系原理，扫清一些迷雾。笔者研究比特币及相关技术三年有余，虽不才，愿来做这样的一个分享尝试。

认识比特币

说起比特币，就会有一大堆的形容词定语扑面而来，比如“数字加密货币”，“去中心化”，“不可篡改性”，“工作量证明”等，好了好了，让我们回到本质来看吧。

首先，比特币是一个软件系统，就跟进销存系统，QQ聊天软件，网络游戏等一样，它就是一个软件，或者更准确的说，它是属于网络软件，我们现在几乎百分之九十九的软件都是网络软件，淘宝、微信、迅雷、网易云音乐等，这些软件都属于互联网应用软件，都是基于TCP/IP协议族开发的，比特币作为一个互联网应用软件也是不例外的，目前，比特币的软件源代码维护在GitHub上：比特币源码链接。

感兴趣的朋友可以打开这个网址看一看，无论是否是程序员，可以了解一下，另外，GitHub是一个公开的源码版本维护平台，Git是一种版本控制软件，GitHub就是基于Git来交换版本历史信息的，并且可以看到，比特币源码的版权并不属于某个公司或者组织，事实上它是被一个称之为中本聪的人或组织发起的，中本聪这个身份到目前也是一个谜，当然这个已经不重要了，重要的是比特币给我们带来了很多有趣的软件设计思路，咱们下面一一介绍一下。

这里我们先介绍一下比特币软件是怎么来的，作为一个软件，从发行起就算是诞生了，而比特币这个软件的发行标志，是源于2008年11月1日，一位署名为中本聪的人发表了一份白皮书《比特币:一种点对点的电子现金系统》，这份白皮书简明而完整的阐述了一种基于密码技术和工作量证明原理的数字货币体系，在这之后不久比特币软件就诞生了。

在比特币软件之前，对于数字货币的研究和软件实验已经存在了很长时间了，甚至在八九十年代就已经出现过了，比如曾经由美国的一个密码破译者发明的DigiCash，而某种程度上，后来在互联网出现的一些代币比如QQ币，乃至银行支持的网银功能等，某种程度上都是一种货币数字化的形式，只不过，直到比特币出现之前，都没有真正的支持如下的特点：

1. 去中心化，也就是说没有一个央行的中心管理，而是通过一套技术手段，实现可信任的网络。

2. 自动化货币发行共识，在比特币中通过工作量证明算法PoW实现了比特币的发行（使用称之为挖矿的一套算法步骤）。

3. 区块链的数据结构，保证数据的不可篡改。比特币系统从发行开始到现在，近9年的历史中，虽然一路坎坷，多受质疑，但并没有发生过明显的破解、数据破译、货币盗取等严重问题，反而越来越成熟，这也证明了比特币系统的一套技术组合的健壮性和可用性。

这里不得不提醒一下，通常人们所说的比特币是指通过比特币软件发行出来的数字货币，而所谓的发行，站在软件的角度，就是记录下来的一个数字。当然，由于比特币技术的开源特性以及新颖的设计，后续不少人借鉴了其中的原理，实现出了其他各类数字货币系统，也称之为竞争币，这部分的人形成的圈子也称之为币圈，而另外一部人发现对于比特币来说，支撑这套系统的技术，才是真正有意义的，也就是区块链技术，进而认为区块链技术并不仅仅只能用于产生数字货币，而可以用来在其他很多领域发挥作用，比如公证系统，数字资产交易，智能合约等等。那么，为了让大家对比特币相关的技术有个比较清晰的了解，以下就按照关键的名词术语来进行介绍。

区块链

区块链，英文blockchain，这是比特币系统用于记录账本数据的格式，我们知道，通过客户端软件可以进行挖矿得到新发行的比特币，可以进行转账，而这些数据都是要存下来的，存在哪呢，当然是存在一个文件里了，那就涉及到以什么样的格式来保存了，就跟我们平常记账一样，比如某年某月某日，收入了多少钱，从哪收入的，支出了多少钱，从哪支出的，一般会有个流水账这么记录着，比特币系统也是一样的，它也得将这些数据记录下来，那记就记呗，为啥叫区块链这么个名字呢？因为这是一个很形象的称呼，请看，区块链，区块链，就是讲区块链接起来这么个意思，那也就是说，数据主要是记录在区块中的，所谓区块什么意思，就是一块打包在一起的数据，那多长时间的数据打包在一起呢，在比特币系统中，是每10分钟左右的数据打包在一起记录下来，并且为这一块打包的数据生成一个区块数据头，专门记录区块的识别ID（哈希值）、区块中

所有交易数据计算出来的哈希树（merkle tree)等等，每一次产生的这样的一个区块都会指向上一个区块，这样就好像链条一样，将一个个区块都链接了起来。

大家可能会奇怪，搞这么一个复杂的做法干嘛，而且看起来挺原始的，现在各种数据库存储技术不是很方便么？这里我们先提出一组问题出来：

1. 如之前所述，比特币是一个去中心的系统，也就是说数据并不是统一存储在某个服务器或者服务器集群上的，而是每个客户端都有一份完整的数据在本地，那如何让每一个客户端在彼此进行数据同步的时候，来验证数据的完整性呢？倘若是一个普通的数据表来存储，将很难去验证数据的完整性，尤其是数据越来越多以后，比如现在比特币的完整区块数据达到了120G，以后还会越来越大，这么大的数据，在进行同步下载的时候，很难去校验中间过程中是不是哪个地方下载错了，乱码了，更要命的是，如果不小心断网什么的导致同步中断了，下次还得重新开始下载，因此需要对数据进行一个细粒度的切分。

2. 如果有人故意更改了记录的数据，怎么办？在普通的数据库系统中，假如更改了某个表中的数据，系统是很难马上去发现的，而如何提高更改数据的难度呢，那就让数据之前彼此具备关联关系，你要改可以，你得全部改掉，就像这里说的区块链，每10分钟的发生数据打包一个区块，一块块相连，想要更改某个区块的数据那就得从头改到底，这个代价是很大的，而且由于比特币系统并不是一个单机软件（单机数据要更改再烦也很容易，一段程序执行就全改了），是通过无数个比特币客户端形成的一个网络，你改了本机的数据是必须要通过全网的节点认可才行的。这种通过打散数据，再进行彼此链接的结构就是称之为区块链，其实也可以叫数据链，信息链，数据块链等，就是这个意思，事实上，在区块链中，除了区块是彼此相关联的，每一笔交易的事务也是彼此相关联的，这个下面会讲述到。

梅克尔树

我们知道，在比特币网络中，区块数据是要在每一个客户端中进行同步的，所谓的同步其实就是下载，比如当前的区块链高度到了10000了，我才是9000，那我得下载这缺少的1000个区块，那么，问题就来了，这些数据可是不少的，而且比特币系统并不是统一从一个服务器去下载，它是一个对等网，也就是俗称的P2P网络，在这种环境下，下载的数据来源是会有多个的，怎么保证在下载过程中不会出错呢？或者说如果出了错，可以不用全部重新下载，而只要重新下载缺失的即可，再一个，如何去校验下载下来的数据到底对不对呢（我们希望能够快速的去校验而不是一条条区块数据去校验），这个时候，梅克尔树就能发挥作用了。

梅克尔树，英文merkle tree，是一种数据结构，它的基本原理其实是很简单的，对于比特币系统中的数据，或者说需要同步的数据，主要就是一个个区块，这个merkle tree就是为这些个众多的区块而生成的，具体怎么办的呢？首先为每一个区块去计算一个hash值，举个例子吧，比如现在一共10个区块，那就计算出10个区块的哈希值，然后每两个相邻的区块的哈希值再次进行计算得出新的哈希值，按这个方法，比如1号和2号区块的哈希再次生成一个哈希，3号和4号区块的哈希再次生成一个哈希，5号与6号，7号与8号，9号与10号，以此类推，直到生成一个根哈希，形成了一棵树的结构。

那么，这么做有什么好处呢，我们看到，针对区块数据计算出一棵哈希树，哈希的用途不必多说，就是一个身份证，那么，所谓的merkle tree，其实就是一棵身份证树，好，回到我们之前的问题设定，在进行同步下载的时候，假设3号区块数据有问题了，此时，我们在进行校验的时候，会非常方便，可以直接定位到是哪个块有问题，因为哈希值不一样，而如果要校验整个区块数据是否完整正确，直接看最顶端的树根那个哈希值就可以了，根本就不用一条一条数据去核对。

这个就是梅克尔树的作用。

UTXO

这是一个缩略语，全称是unspend transaction output，翻译过来是未花费事务输出，还是举个例子吧，比如一个小店，卖衣服的，分批分次入库了10件，20件，30件衣服，那么如果是一般性的做法，会怎么记账，除了记录下这些入库流水账外，还会设立一个库位或者说账户，比如某西装共60件，然后销售出货的时候，每卖掉一批就从这个60的数量里面减掉相应的数量。这个思路可以说简直就是常识了，那么这个跟UTXO有什么关系呢，好，解释UTXO之前，先来看看比特币中的那些个转账事务怎么记录的，有没有像这样也搞一个账户呢？事实是：没有！那么比特币中是怎么存储的，很简单，就是存储一次次的转入流水账和一次次的转出流水账，这样好像也没什么问题啊，大家注意没有，在这种情况下，如果说我要发出5件衣服，就不是简单的到一个汇总账户中，把60减成55了，因为这个时候没有一个60的账户，那么应该怎么出呢，此时需要找到一个之前的入库事务，比如之前入过10件，可以以这个入库的10件的流水事务作为来源，进行发货，入库10，则表明可以出掉10，现在只需要出掉5，出掉后，之前的入库10的事务就不存在了，而只有一个入库5的事务了，这个入库5的事务，就是属于未花费输出，顾名思义，因为比特币系统中，输入输出的是货币单位，因此用花费表示，如果是上述的衣服，也可以叫“未使用的，可出货的入库来源”。

UTXO基本就是这么一个意思。

SPV

SPV也是一个术语，是属于比特币钱包应用方面的术语，这里首先得提一下比特币钱包的概念，在中本聪客户端也即比特币的核心程序中，提供有比特币钱包的功能，用于保存比特币地址，私钥这些数据，类似于生活中使用的钱包，而这个中本聪钱包，是需要与完整的区块数据在一起使用的，这就有一个问题了，区块数据是很大的，尤其是时间长了以后，那是几十个G，上百个G这样的数据量，而且还会继续增长，为了进行转账操作，得附带这么大一坨数据，如何是好？于是就有了这么一个概念，SPV，全称是Simplified Payment Verification，翻译过来是简单支付验证，什么意思呢？我们知道，在区块中，占据数据量的主要是那些交易数据或者说是转账事务，而对应这些数据在区块头中有一个上述所说的梅克尔树，那么，既然如此，能不能这样呢，钱包中只保留区块头，在进行支付的时候，通过区块头可以进行支付验证，注意了，是支付验证，不是交

易验证，支付验证，主要是为了验证这笔支付是否已经得到了网络的确认，而真正的交易确认，仍然是通过完整客户端进行验证的。

这个思路，是把钱包功能从核心客户端中分离了出来。

分层确定性钱包

这个概念仍然是与钱包相关的，在比特币系统中，一个人可以拥有的钱包地址、私钥这些是几乎无限的，那么，不管怎么说，一个人往往也不会只用一个地址，而对于团体用户来讲，更加不可能只是一个地址了，如果是按照比特币核心客户端的做法，1个地址，1个独立的私钥，这样固然很安全，但是却很不方便使用，谁也没法去记住那么长，那么多的地址，私钥，于是就有人提出了这么一个概念。

大体是这么一个思路：

1. 用一个随机数来生成根私钥。

2. 用一个确定不可逆的算法，基于根钥生成任意数量子钥。

可以看到，在这样一个钱包中，其实还是产生了一棵树，是一棵密钥树，为什么说是确定性钱包呢，因为只要根据根钥就能产生其他所有的密钥，为什么说是分层呢，因为结构是一棵树，是有层次的。

双花与51%攻击

从这里开始，介绍一些比特币系统中涉及到的问题，双花是个什么概念呢，就是一笔钱被花了两次，在纸币的使用过程中，钱不可能被重复的花，因为纸币是一个实体，花出去就没了，而我们日常使用的银行卡，也不会有双花的问题，因为银行为每张卡维护了一个余额账户（极少数银行系统出问题的情况就不算了），那么，对于比特币系统，既没有一个银行这样的账户中心，也不是纸币这样的实物货币，怎么保证比特币不被重复花呢？

我么知道，在比特币网络中，所有的交易事务，每隔10分钟会打包成一个数据包，也就是所谓的区块，存储下来，并且广播到全网，每一个收到广播的软件客户端都会去进行一个同步。也就是说，钱能不能再次被花出去，涉及以下几个关键点：

1. 自己重复花的钱能够被自己打包篡改。

2. 打包篡改的金额能被网络确认。

那么，这里就涉及一个概念了，也就是比特币网络中，既然没有一个中心服务器做统一可靠的处理，那是靠谁来打包的呢？这个每10分钟打包一个区块的事情谁来做呢？这是通过一种叫做工作量证明的机制来进行的，简单的说，就是每个客户端，都会去计算一

个密码题，谁先计算出来谁就拥有打包权，那谁愿意来做这么个事情呢？比特币网络是有激励机制的，获得打包权的会奖励一定数量的比特币。那么好，这就保证了，并不是自己想要打包就打包的，自己就算修改了本机的区块数据，可是广播出去被验证为无效还是白搭。

也就是说，想要双花，起码要能显著优势的抢夺打包权。目前比特币网络的算力非常庞大，很难发送这样的攻击。

这个攻击就是51%攻击，也就是说，攻击者拥有全网算力的一半以上，数据总是自己来打包的，自然就能不断的去修改数据，广播出去，再自己打包。

分叉

这个问题很容易解释，区块链，数据结构就是一个链表的样子，那么，有无可能，在有一天，这个链条突然变成了两个链了呢？举一个例子，比特币的数据结构或者协议进行了修改，升级到新版了，但是由于比特币软件到处都是，你没法强迫每个人都马上升级到新版，那就有可能出现，旧版本与新版本共存的情况，而如果新版本启用了某些新的数据协议或者数据结构，旧版本无法去完全理解新版本打包的数据，这就会导致分叉了。

对于分叉，也有区分，有软分叉，硬分叉。

简单的说，软分叉是指旧版本的客户端能读取自己能理解的那部分区块数据，但是对于新增的字段不能理解，大家还是能共存；硬分叉是指旧版本不能理解新版本的数据，从而导致完全分叉。

总结

比特币的相关技术是非常多的，本文只能是截取其中一部分的关键点进行一些阐述，行文过程难免会有纰漏，请大家见谅，欢迎大家指正。