## 以太坊专业术语表

关于加密货币以及任何新的技术的令人不爽的一件事，就是其用于描述所有新概念的词汇的绝对数量。只要不是临时的，最基本的处理的情况，任何与在点对点互联网软件打交道的人都需要与加密的概念打交道，包括散列，签名，公钥，私钥，对称和不对称加密，服务保护拒绝，还有例如分布式哈希表，信任网络之类的神秘的构造。新的比特币用户被迫与学习加密的基本常识而奋斗，还有额外的内部术语，如“区块”，“确认”，“挖矿”，“SPV客户”和“51％的攻击”，以及经济概念，如激励相容和集权与分权的微妙差别。以太坊，作为一个基于加密货币的泛化的分散的应用开发平台，必然既包含这些概念的集合，也增加了许多自身的概念。为了帮助新来以太坊的人，无论他们是加密货币爱好者，企业家，社会或政治愿景家，Web开发人员或只是看看这项技术能如何改善生活的普通人，下面的列表的目的是为了提供一个以太坊用户常常使用的词汇的基本总结：

### ****密码学****

另请参阅：http://en.wikipedia.org/wiki/Public-key\_cryptography

计算上不可行：一个处理被称为是计算上不可行，如果有人想有兴趣完成一个处理但是需要采取一种不切实际的长的时间来做到这一点的（如几十亿年）。通常，2的80次方的计算步骤被认为是计算上不可行的下限。

散列：一个散列函数（或散列算法）是一个处理，依靠这个处理，一个文档（比如一个数据块或文件）被加工成看起来完全是随机的小片数据（通常为32个字节），从中没有意义的数据可以被复原为文档，并且最重要的性能是散列一个特定的文档的结果总是一样的。

此外，极为重要的是，找到具有相同散列的两个文件在计算上是不可能的。一般情况下，即使改变文件的一个字母也将完全打乱散列;例如，“ Saturday”的SHA3散列为c38bbc8e93c09f6ed3fe39b5135da91ad1a99d397ef16948606cdcbd14929f9d，而Caturday的SHA3散列是b4013c0eed56d5a0b448b02ec1d10dd18c1b3832068fbbdc65b98fa9b14b6dbf。散列值经常被用作以下用途：为无法伪造的特定文档而创建的全局商定标识符。

加密：与被称为钥匙（例如c85ef7d79691fe79573b1a7064c19c1a9819ebdbd1faaab1a8ec92344438aaf4）的短字符串的数据相结合，对文档（明文）所进行的处理。加密会产生一个输出（密文），这个密文可以被其他掌握这个钥匙的人“解密”回原来的明文，但是对于没有掌握钥匙的人来说是解密是费解的且计算上不可行。

公钥加密：一种特殊的加密，具有在同一时间生成两个密钥的处理（通常称为私钥和公钥），使得利用一个钥匙对文档进行加密后，可以用另外一个钥匙进行解密。一般地，正如其名字所建议的，个人发布他们的公钥，并给自己保留私钥。

数字签名：数字签名算法是一种用户可以用私钥为文档产生一段叫做签名的短字符串数据的处理，以至于任何拥有相应公钥，签名和文档的人可以验证（1）该文件是由特定的私钥的拥有者“签名”的，（2）该文档在签名后没有被改变过。请注意，这不同于传统的签名，在传统签名上你可以在签名后涂抹多余的文字，而且这样做无法被分辨;在数字签名后任何对文档的改变会使签名无效。

### ****区块链****

也可参考：https://bitcoin.org/zh\_CN/vocabulary

地址：一个地址本质上是属于特定用户的公钥的表现;例如，与上面给出的私钥的相关联的地址是cd2a3d9f938e13cd947ec05abc7fe734df8dd826。注意，在实际中，地址从技术上来说是一个公钥的散列值，但为了简单起见，最好忽略这种区别。

交易：一个交易是一个文档，授权与区块链相关的一些特定的动作。在一种货币里，主要的交易类型是发送的货币单位或代币给别人;在其他系统，如域名注册，作出和完成报价和订立合约的行为也是有效的交易类型。

区块：一个区块是一个数据包，其中包含零个或多个交易，前块（“父块”）的散列值，以及可选的其它数据。除了初始的“创世区块”以外每个区块都包含它父块的散列值，区块的全部集合被称为区块链，并且包含了一个网络里的全部交易历史。注意有些基于区块链的加密货币使用“总账”这个词语来代替区块链。这2者的意思是大致相同的，虽然在使用“总账”这个术语的系统里，每个区块都通常包括每个账户的目前状态（比如货币余额，部分履行的合约，注册）的全部拷贝，并允许用户抛弃过时的历史数据。

帐户：帐户是在总账中的记录，由它的地址来索引，总账包含有关该帐户的状态的完整的数据。在一个货币系统里，这包含了货币余额，或许未完成的的交易订单;在其它情况下更复杂的关系可以被存储到账户内。

工作证明：在比特币，以太坊和许多其他加密总账里的一个重要特性，意思是在区块中的散列值必须比某个目标值小。这个必要的原因是，在分散式系统中任何人可以产生区块，因此为了防止网络中区块泛滥，并提供一种方法来衡量在区块链的一个特定版本后有多少共识，使得产生一个区块非常艰难。由于散列值是伪随机的，找一个散列值比0000000100000000000000000000000000000000000000000000000000000000还小的区块，平均需要43亿次尝试。在所有这些系统中，目标值进行自我调整以便在网络上的一个节点平均每N分钟（例如，比特币N =10，以太坊N＝1）发现一个区块，

随机数：在一个区块里的一个无意义的值，为了努力满足工作证明的条件来进行调整。

挖矿：挖矿是反复总计交易，构建区块，并尝试不同的随机数，直到找到一个随机数可以符合工作证明的条件的过程。如果一个矿工走运并产生一个有效的区块的话，会被授予的一定数量的币（区块中的交易全部费用）作为奖励。而且所有的矿工开始尝试创建新的区块，这个新区块包含作为父块的最新的区块的散列。

陈腐区块：对于同一个父块，已经有另外一个区块被创建出来之后，又被创建的区块;陈旧区块通常被丢弃，是精力的浪费。

分叉：指向同一个父块的2个区块被同时生成的情况，某些部分的矿工看到其中一个区块，其他的矿工则看到另外一个区块。这导致2种区块链同时增长。通常来说，随着在一个链上的矿工得到幸运并且那条链增长的话，所有的矿工都会转到那条链上，数学上分几乎会在4个区块内完结自己。

双重花费：是一个故意的分叉，当一个有着大量挖矿能力的用户发送一个交易来购买产品，在收到产品后又做出另外一个交易把相同量的币发给自己。攻击者创造一个区块，这个区块和包含原始交易的区块在同一个层次上，但是包含并非原始交易而是第二个交易，并且开始在这个分叉上开始挖矿。如果攻击者有超过50％的挖矿能力的话，双重花费最终可以在保证在任何区块深度上成功。低于50％的话，有部分可能性成功。但是它经常在深度2-5上有唯一显著的可能。因此在大多数的加密货币交易所，博彩站点还有金融服务在接受支付之前需要等待6个区块被生产出来（也叫“6次确认”）。

SPV客户端（或轻客户端）：一个只下载一小部分区块链的客户端，使拥有像智能手机和笔记本电脑之类的低功率或低存储硬件的用户能够保持几乎相同的安全保证，这是通过有时选择性的下载的小部分的状态，而在区块链验证和维护时，不需要花费兆字节的带宽或者千兆字节的存储空间。

### ****以太坊区块链****

也可参考： http://1tf.org/index.php?p=/discussion/71/

序列化：将一个数据结构转换成一个字节序列的过程。以太坊在内部使用的编码格式称为递归长度前缀编码（RLP），在这里有描述

帕特里夏树：一种数据结构，它会存储每个帐户的状态。这个树的建立是通过从每个节点开始，然后将节点分成多达16个组，然后散列每个组，然后对散列结果继续散列，直到整个树有一个最后的“根散列”。该树具有重要的特性：（1）只有正好一个可能的树，因此，每个数据集对应一个可能的根散列（2）很容易的更新，添加，或者删除树节点，以及生成新的根散列，（3）不改变根散列的话没有办法修改树的任何部分，所以如果根散列被包括在签名的文档或有效区块中话，签名或工作证明可以担保整个树（ 4）任何人只可以提供一个下到特定节点的分支，可以加密得证明拥有确切内容的节点的确是在树里。帕特里夏树也被用来存储账户，交易已经叔块的内部存储。在这里能看到更详细的说明。

幽灵（Ghost）：幽灵是一个协议，通过这个协议，区块可以包含不只是他们父块的散列值，也散列父块的父块的其他子块（被称为叔块）的陈腐区块。这确保了陈腐区块仍然有助于区块链的安全性，并减轻了大型矿工在快速区块链上的有优势的问题，因为他们能够立即得知自己的区块，因此不太可能产生陈腐区块。

叔块：是父区块的父区块的子区块，但不是自个的父区块，或更一般的说是祖先的子区块，但不是自己的祖先。如果A是B的一个叔区块，那B是A的侄区块。

帐户随机数：每个账号的交易计数。这样可以防止重放攻击，其中一个交易发送比如 20个币从A到B，并可以被B重放一遍又一遍，直到不断抽干A的账户余额。

EVM代码：以太坊虚拟机代码，以太坊的区块链可以包含的编程语言的代码。与帐户相关联的EVM代码在每次消息被发到这个账户的时候被执行，并且具有读/写存储和自身发送消息的能力。

消息：一种由EVM代码从一个账户发送到另一个账户的“虚拟交易”。需要注意的是“交易”和“消息”在以太坊种是不同的;在以太坊用语的“交易”具体指的是物理的数字签名的一串数据，并且每个交易触发相关联的消息，但消息也可以通过EVM代码发送，在这种情况下，它们从不表示成任何数据。

储存：包含在每个帐户里的键／值数据库，其中键和值都是32个字节的字符串，但可以以其他方式包含任何东西。

外部拥有账户：通过私钥控制的账户。外部拥有账户不能包含EVM代码。

合约：一个包含并且受EVM的代码控制的账户。合约不能通过私钥直接进行控制，除非被编译成EVM代码，一旦合约被发行就没有所有者。

以太（Ether）：以太坊网络的内部基础的加密代币。以太是用来支付交易和以太坊交易的计算费用。

瓦斯：大致相当于计算步骤的计量。每一笔交易需要包括瓦斯的限制，还有愿意为每瓦斯支付的费用;矿工可以选择是否收录交易和收集费用。由包括原始消息以及任何可能被触发的子消息的交易产生的计算所使用的瓦斯总量，如果大于或者等于瓦斯的限制，则交易被处理。除非交易仍然有效并且费用仍然被矿工收集，否则瓦斯的总量小于限制则所有变更被还原。每一个操作都有瓦斯支出;对于大多数操作，花费是1瓦斯，尽管一些昂贵的操作会支出高达100瓦斯，交易本身会有500瓦斯的支出。

### ****非区块链****

以太浏览器（Mist）：即将到来的以太坊基础客户端，会以Web浏览器的形式存在，可被用来访问正常的网站以及建立在以太坊平台上的应用程序。

耳语（Whisper）：即将到来的点到点信息协议，将会被整合到以太浏览器。

蜂群（Swarm）：即将到来的，为静态web托管而优化的点到点数据存储协议，将被整合到以太浏览器。

LLL，Serpent和Mutan：为编写合约代码的编程语言，可被编译成EVM代码。serpent可以被编译为LLL。

PoC：概念证明(proof-of-concept)的英文缩写，预发布版的另一个称呼。

### ****周边概念：应用程序和治理****

分散化应用程序：为了某些特定目的（如：在某些市场上连接买家和卖家，共享文件，网络文件存储，维持货币），无论是使用还是创建一个分散的网络，由许多人来运行的程序。基于以太坊的分散式的应用程序（也称为Đapps，其中Đ为北欧字母“eth”）通常包括一个HTML/ JavaScript的网页，并且如果在以太浏览器内部查看的话，可识别特殊的Javascript的API，用于发送交易数据到区块链，从区块链读取数据，和耳语，蜂群交互数据。一个Đapp通常在区块链上有特定的相关合约，但有利于创造许多合约的Đapps是完全可能的。

分散化组织（GDO）：一个没有中央领导，而是使用正式民主投票进程和共识主动性自我组织的结合来作为其基本操作原则。一个不太令人印象深刻，但有时混淆的概念是“地理上的分散化组织”（GDO），组织里人在相距甚远的地方工作，甚至可能都没有办公室; GDOs可能会有正式的中央领导。

忒修斯标准：用于查明一个组织的分散化程度的测试。测试如下：假设组织有N个人，然后外星人一次从组织中（比如每周一次）挑选K个人出来，摧毁他们存在，在每个群里以K个对组织不了解的新人来代替。现在为了让组织起作用，K可以高达多少人呢？在独裁政权里，当K＝1即独裁者被摧毁后就会失败。美国政府稍微好一点，但如果参议院和国会的所有638成员突然消失了的话，仍然会有很大的问题。但像比特币或BitTorrent即便对极高的K值也具有复原性，因为新的代理人可以简单地根据自己的经济动机来填补缺失的角色。还有一个更严格的测试，拜占庭忒修斯标准，它包含同一时间内随机的用恶意行为者取代K个用户一段时间，之后再替换成新用户。

委任式民主（或流动式民主）：一个对于DOs(分散式组织)和DAO(分散式自治组织)的治理机制，在默认情况下每个人对每件事情都投票，但在某些特定的问题上个人可以选择特定的他人为他们投票。这个想法概括了以下2种民主的权衡，完全直接民主(每个人都有相同的权力)和专家意见/有某些特定人提供的快速决策能力（允许人们自己顺从朋友，政治家，领域专家或者自己选择的任何人）。

部分被投机市场控制的理论上的政府：最初是由Robin Hanson提出的，为了管理政治组织治理机制。但它实际上是非常适用DOs和DAO的：通过预测市场来管理。从根本上，一些易于衡量成功的标准被选择，还有发行由成功标准的值来决定的代币，这些代币将在未来的某个时间（例如，1年后）被支付，对于每个可能要采取的行动都用一个这样的代币。这些代币都被兑换为相应的美元代币，如果相应的措施被执行,正好1美元会被支付 （如果相应的措施没有被执行，这两种类型的代币支付0美元，所以正在被执行的行动的概率不会影响价格）。市场预计的行动将有最好的结果，当其代币在市场上有高价格时会被执行。这提供了另一种自治的，选择机制，同时奖励专家的意见。

### ****经济学****

代币制度：本质上是可以交易的虚拟代替物。更正式地说，代币制度是一个数据库，它映射地址到数字，并具有以下属性，基本允许的操作是把N个代币从A转给B，条件是N是非负，且N不小于A的当前余额，授权该转账的证件由A进行数字签名。二次“发行”和“消费”的操作也可以存在，交易费用也可以被收集，许多当事人同时进行转账也是可能的。典型应用案例，包括货币，网络加密代币，公司的股份和数字礼品卡。

命名空间：一个映射名字到值的数据库。举个最简单的例子，如果名称尚未被占有的话（也许可以在支付一些费用之后）任何人可以注册一个条目。如果名称已经被占有了，那么就只能被改变（如果有的话），由作出原始注册的账户（在许多系统中，所有权也转移）。命名空间可以用来存储用户名，公钥，互联网域名，代币制度或其他命名空间，和许多其他应用。

身份：一组可以加密验证的互动，具有同一个人创建的的属性。

唯一的身份：一组可以加密验证的互动，具有以下属性：同一个人创建的。再加上一个人不能有多个唯一身份的约束。

激励相容：如果每个人都更好的“遵守规则”而不是试图欺骗，除非至少要大量的人都同意同时一起欺骗，那么协议是激励相容的。

基本收入：每隔一段时间（比如几个月）就给每一个唯一的身份发送一定量的代币的想法。其最终目的是为了让不愿意工作或者不能工作的人能够依靠这份津贴活下来。这些代币可以简单的凭空制作出来，或者来自收益流（比如来自创收实体或政府）。为了单靠基本收入使人能够生活，可能会用到多个收益流的组合。

公益：一个为非常多的人提供了一个非常小的好处的服务。这样就没有任何个体对是否进行生产有影响力，因此也没有人有动力来支付。

声誉：身份的一个属性，其他实体认为这个身份可以（1）胜任一些特定的任务，或（2）在一些情况下是值得信赖。比如说不太可能因为短期的获利而出卖别人。

信任网络：如下的想法，如果A高度信任B，B高度信任C，则A可能是信任C的。为决定特定个体在特定概念下的可靠性的复杂而有力的机制，理论上可以由这个原则推断出来。

第三方托管：如果两个低信誉的实体所从事的贸易时，付款人可能希望把钱留在具有高信誉的第三方，并指示只有在产品交付后，才让第三方把钱发给收款人。这减少了付款人或收款人欺诈的风险。

保证金：放入合约里的涉及另外一方的数字资产，如果某些条件不满足时，该资产会自动被对方没收。

抵押：放入合约里的涉及另外一方的数字资产，如果某些条件不满足时，该资产会自动被销毁或据献给慈善或者基本收入基金。也许可以让利益广泛分配，但必须让特定的个人不能显著的受益。