

基础知识

货币的价值：交换价值，使用价值
货币的形式：贵金属，纸币，数字货币
经济学：马克思主义：商品的使用价值和jiao换价值；凯恩斯经济学：市场有时候会失灵，需要干预；哈耶克经济学：自由市场、自由经营、自由竞争、自动调节、自动均衡，货币非国有化；现代货币理论 MMT：货币的本质就是欠条，即 IOU (I Owe You)
Hash 算法：Hash 实际上是一种思想，包含很多算法。
HASH 应用：快速定位（数据库中散列表）；错误校验；唯一性验证
HASH 常见算法：MD4、MD5、SHA1、SHA2
hash 算法：趋向困难，抗碰撞
SHA256 输出为 256 位，输出空间是 2^256，是“亿亿亿亿亿亿亿”。
非对称加密两种算法：
RSA 算法：大素数分解
椭圆曲线加密：离散对数难题

区块链技术原理

任何数字货币都会面临虚拟货币和多重支付的问题，比特币解决方法，虚拟货币（数字签名），多重支付（分布式账本）。
比特币核心技术
1. 以链式区块组织账本数据实现账本数据的不可篡改
2. 分布式的可信记账机制
区块链基于哈希值进行链接，特点是区块链中数据无法篡改或删除：区块链越长可信度越高
区块链中每个区块包括区块头和交易数据两个部分，其中
1. 区块头由当前区块的元数据和前一区块的 Hash 值构成
2. Merkle 树用于对交易数据列表进行快速寻址
区块的微观结构

4 字节	区块大小	区块大小
80 字节	区块头	区块头
1-9 字节	交易计数器	交易数量
可变	交易	交易信息

区块头结构

4 字节	版本	版本号
32 字节	父区块头 hash 值	
32 字节	merkel 根 hash 值	
4 字节	时间戳	产生的时间/秒
4 字节	难度	
4 字节	nonce	pow 随机数

注意区块头中的前一个区块的 hash 值为其区块头的 hash 值，不是整个区块。
区块的区分方式
1. 区块主标识符是它的加密哈希值，一个通过 SHA256 算法对区块头进行二次哈希计算而得到的数字指纹。
2. 第二种识别区块的方式是通过该区块在区块链中的位置，即“区块高度（block height）”。例如：高度为 0 的区块就是创世区块，和区块哈希值不同的是，区块高度并不是唯一的标识符，因为有可能出现区块链分叉。
merkle 树
Merkle 树是一种哈希二叉树，它是一种用作快速归纳和校验大规模数据完整性的数据结构。这种二叉树包包含加密哈希值。叶节点是数据的哈希值。非叶节点的哈希值是根据它下面子节点的值哈希计算得到，在比特币网络中，Merkle 树被用来归纳一个区块中的所有交易，同时生成整个交易集合的数字指纹，且提供了

一种校验区块是否存在某交易的高效途径。
Merkle 树中使用两次 SHA256 算法计算结点的哈希值。H-A~ = SHA256(SHA256(交易 A))，两次 SHA256 是为了提高安全强度。当 N 个数据元素经过加密后插入 Merkle 树时，你至多计算 log2(N)次就能检查出任意某数据元素是否在该树中，这使得该数据结构非常高效。
merkel 树的价值：快速比较大量数据、快速定位修改、快速验证其中数据（merkel 路径）
比特币节点
全节点：存储着整个区块链，承担对交易请求进行验证 和执行，可以通过挖矿争取发布区块，还承担着应别的节点之请向其发送区块和相关交易信息的义务，同时也承担转发交易请求和区块的义务。
轻节点：
1. 简单支付验证(SPV)节点：只存储区块头，不存储区块块体，仍可以对到来的交易请求进行验证。
交易验证过程：为交易建立布隆过滤器，只接受指定目标地址的交易，其他全节点探测到某个交易符合 SPV 节点设置的布隆过滤器条件时，以 Merkleblock 消息的形式发送该区块，Merkleblock 消息包含区块头和一条连接目标交易与 Merkle 根的 Merkle 路径。
• **交易的存在性验证**:SPV 节点通过该 Merkle 路径找到跟该交易相关的区 块，并验证对应区块中是否存在目标交易(Merkle Path Proof)。
• **交易是否双化验证**:SPV 节点检查这笔交易所在区块之后的区块个数，区块个数越多说明该区块被全网更多节点共识，一般来说，一笔交易所 属区块之后的区块个数达到 6 个时，说明这笔交易是可信的。
2. 钱包:一个连接区块链的应用软件(app)，记录与所有者有关的信息：区块链地址、私钥、账户余额、UTXO 等，不存储账本。
交易的结构
Class transaction
]const std::vector<CTxIn> vin //输入 utxo
]const std::vector<CTxOut> vout //输出 utxo
]const int32_t nVersion//版本
]const uint32_t nLockTime//锁定时间，在此之前交易不进入区块
]const uint256 hash //交易 hash，不存储不发送
工作量证明
PBFT 要求节点数量有限，点集相对稳定，所以不适用。
pow 机制：划定固定时间段(10 分钟)
• 相同或相似输入数据(组装的区块)
• 算力竞争选出获胜节点，其它节点验证结果后不再发送消息。
最长链原则：从短期共识扩展到长期共识
pow 机制中难度的确定方式：在每个完整节点中独立自动发生的，每 2,016 个区块中的所有节点都会调整难度。难度的调整公式是由最新 2,016 个区块的花费时长与 20,160 分钟（两周，即这些区块以 10 分钟一个速率所期望花费的时长）比较得出的。难度是根据实际时长与期望时长的比值进行相应调整的（或变难或变易）。简单来说，如果网络发现区块产生速率比 10 分钟要快时会增加难度，如果发现比 10 分钟慢时则降低难度。
比特币分叉处理机制：把当前同一个父区块下的若干有效子区块都记录，形成兄弟区块（产生分叉）•后续区块（第 3 代、第 4 代.....）到达后，依次加在前序区块后，若没有其他竞争性区块，这一分支最长，成为主链。
比特币挖矿的奖励机制：挖矿节点必须有钱包

功能，有自己的 160 位密码地址、私钥。
打包生成区块时，区块中额外加一个交易 coinbase 生成一个 UTXO，包含当前奖励数量的比特币（现在是 6.25）。这个 UTXO 的招牌地址是自己的地址，如果记账成功，这个 coinbase 交易就生效，否则不在链上。
挖矿机制的总结：
每个全节点依据综合标准对每个交易进行独立验证，通过完成工作量证明算法的验算，挖矿节点将交易记录独立打包进新区块，每个节点独立地对新区块进行校验并组装进区块链每个节点对区块链进行独立选择，在工作量证明机制下选择累计工作量最大的区块链
UTXO 模型：每个比特币用户有一个 160 位 (20 字节) 长度的地址，产生的过程：用户生产一一对非对称密钥，公钥经 hash 计算（**SHA160**）产生 160 位的地址，私钥自己保存，用于数字签名，这个 160 位地址，是用户在比特币网络中交易的唯一标识，与一般的银行账户模型不同，比特币不维护每个账户的资金余额，而是采用一种称为 UTXO 的模型。
比特币中有两类交易
常规交易，有交易输入（支付者地址和金额）、交易输出（收入者地址和金额）；挖矿交易（Coinbase），产生比特币，只有交易输出（挖矿者地址和金额）。
每个地址的资金余额就是散布在账本中所有 UTXO 的总和，使用时把自己名下的 UTXO 作为交易输入，可能需要拼凑找零。
交易输出的结构
ctxout 结构 = nValue(比特币价值) + scriptPubKey(招领脚本)，不一定是公钥 scriptPubKey 可以分为 P2PK, P2PKH, P2SH, P2WPKH, P2WSH
交易输入的结构
]COutPoint prevout 资金来源
]uint256 hash 存储了交易的 hash 值
]uint32_t n 是上述交易的第几个输出
]CScript scriptSig //认领脚本
]uint32_t nSequence //特殊作用
]CScriptWitness scriptWitness //见证脚本
比特币虚拟机
比特币节点软件的一个模块，堆栈结构
运行的程序：认领脚本 scriptSig、招领脚本 scriptPubKey
指令集：有限能力，主要验证签名是否正确
认领脚本 scriptPubKey
P2PK, “PaytoPublicKey”，付给公钥。付给给定 256 位公钥的主人。
P2PKH, “PaytoPublicKeyHash”，付给公钥的 Hash 值，这里所谓的 Hash，是指针对于 256 位公钥的 160 位 Hash，那就是对方的“地址”。P2SH, “PaytoScriptHash”，付给脚本的 Hash 值，给定一个脚本的 Hash 值，付给能提供这个脚本的对象，所提供脚本的 Hash 值必须与给定的 Hash 值相同，这个脚本是收付双方预先约定的，是双方在“链外”约定的。
需要多方签名并采用 SegWitness 脚本的支付，具体又有两种：
P2WPKH, “PaytoSegWitnessPublicKeyHash”，付给 SegWitness 形式的多个公钥 Hash 值。
P2WSH, “PaytoSegWitnessScriptHash”，付给 SegWitness 形式的脚本 Hash 值。
网络通信
通信方式：p2p
如果存在种子节点，新节点先和和种子节点通信，得到伙伴列表
新节点和上述伙伴通信，对有反应的请求新的

伙伴列表，最终获得最新的伙伴列表
区块链技术的优点
去中心化：避免垄断，点对点交易，去代理数据公开：无暗箱操作，平等，开放生态体系可信：数据永久存储，记录可信
比特币的问题
隐私问题
匿名地址如何监管
性能问题
交易确认时间长
区块容量有限
系统性风险
区块链分叉和 51%攻击
什么是比特币分叉
比特币协议分叉又不同于区块链共识中常规分叉，而是因为协议升级或分歧，导致遵循不同协议的软件所产生区块链不兼容，形成各自生长的链。
支付通道
用于链外支付或其他交易，向性能要求高、交易数量大等场景
涉及的链上交易：
注资交易（Funding Transaction)
A、B 双方建立一个 2-of-2 多签名的联合地址，由拟议中的付方发布一个交易将一笔钱打到这个地址中；但这是 P2SH 支付，即支付给能够正确提供清算脚本 Hash 值的收款方，因为是 2-of-2 就必须有双方的签名才能花，。这是后面链外支付的资金来源。
1 个付款方：单向通道，2 个付款方：双向通道
应承交易（Commitment Transaction)
A、B 双方的链下交易，可以有很多个，不上链。每次支付的资金都来自同一个 UTXO，每次交易的输出分成两部分，一是给收方的 UTXO，其数值是付方至此为止承诺支付的总和，二是给付方自己的找零。
决算交易（Settlement Transaction)
收款方把手中由付款方开具并签名的最后一个应承交易上签上自己的名并把它发送到比特币网上，从当初注资阶段生成的那个 UTXO 中把钱划给自己，同时把剩余的钱（如果还有的话）找还给付款方。这个操作也可以由付款方发起，因为收款方每次收到应承交易都会签上自己的名并发还给付款方。
退款交易（Refund Transaction)
闪电网络，大致有以下几种支付方式
转存交易（DeliveryTransaction），任何一方都可以从联合地址认领属于自己的资金，把它转到一个自己更方便花费的地址中。
可撤销转存交易（RevocableDeliveryTransaction），可以撤销。
救交易（BreachRemedy），这是在对方违约情况下加以补救并使对方受惩罚的交易。

数字货币和区块链生态

区块链生态：
联盟链（ConsortiumChain）对产业或者国家的特定清算和结算用途，容易进行控制权限设定，更高的可扩展性。
公链（任何人）任何人都可以参与，容易部署应用程序，全球范围可以访问，不依赖于单个公司
私链(内部链)由单独的个人或者组织拥有，对组织内部的审计和测试有用。
无许可区块链不一定是公链
Script hash 算法介绍
Script 算法由 FreeBSD 黑客 Colin Percival 开发

的，原来是用于密码抗 rainbow table 攻击设计的。
Script 计算所需时间长，而且占用的内存也多，使得并行计算多个摘要异常困难。
Script 也是一种符合区块链 PoW 共识机制的算法。Script 算法过程中也需要计算哈希值，但是 Script 计算过程中需要使用较多的内存资源，可以抗 ASIC 挖矿（ASIC resistance）ASIC（Application Specific Integrated Circuit）是专用集成电路芯片
以太坊
以太坊的组成
P2P 网络 以太坊以 P2P 方式进行网络通信，通过 TCP 端口 30303 访问。
交易 Transaction 以太坊交易是网络消息，包括转账交易、合约交易等。
状态机 State Machine 以太坊的状态转移由 以太坊虚拟机(EVM) 处理，这是一个执行 bytecode(机器语言指令)基于栈的虚拟机，称为“智能合约”的 EVM 程序以高级语言(如 Solidity)编写，并编译为字节码以便在 EVM 上执行。
区块链账本 以太坊的区块链账本存储在每个节点上，该区块链在 Merkle Patricia Tree 的序列化哈希数据结构中包含交易和系统状态。
共识算法 以太坊 1.0 使用名为 Ethash 的工作量证明算法，以太坊 2.0 过渡到称为 Casper 的权益证明机制(Proof-of-Stake)。
客户端 以太坊有几个可互操作的客户端软件实现，最著名的是 Go-Ethereum(Geth)和 Parity
以太坊技术架构
应用层：数字钱包，Dapp，以太坊应用
合约层：智能合约 evm
激励层：发行机制、分配机制
共识层：pow, pos, dpos, pow+pbft, Pbft, poa
协议层：http, rpc, les, eth, whisper
网络层：p2p 网络、数据传输机制、数据校验机制
数据层：数据区块、链式交易、交易池、merkel 树、非对称加密、event 事件
存储层：leveldb, log
以太坊中的两种账户以及存储结构
外部账户:由区块链外部主体创建，拥有一对公私钥，账户地址没有合约代码。
合约账户:由合约交易创建，账户拥有合约代码，地址是代码的 hash 值。
•nonce: 对外部账户，代表该账户发出的交易数, 对合约账户，表示该账户创建的合约数量。
•balance: 该账户地址的 Wei，1018 Wei = 1 Ether
•storageRoot: Merkle Patricia 树的根节点 hash 值，该树是该账户下存储信息的 hash 值，缺省为 0。
•codeHash: 该账户的 EVM 代码 hash 值，对于合约账户，是合约代码存储的 hash 值，对外部账户，是空字符串的 hash 值。
合约账户和外部账户之间的区别
外部账户
•创建外部账户没有成本
•可以启动交易 •两个外部账户间的交易只能是以比特币转账交易 •由公私钥对控制
合约账户

•创建合约账户需要成本，因为使用了网络存储 •只能在收到一个交易作出响应而发出一个交易 •从一个外部账户到一个合约账户的交易可以触发合约账户上的代码，执行代码 中的各种动作，例如代币转账、创建新合约等 •合约账户没有私钥，相反，它们由智能合约代码的逻辑控制
以太坊世界状态
以太坊本质上是一个状态机，有一个 genesis 作为初始状态，交易是其状态转换的最小单元(原子性，一致性)，每次执行一条**或者多条交易**之后发生状态转换。
世界状态中外部账户的地址就是公钥计算得到，合约账户的地址以以太坊合约的地址是根据创建者 (sender) 的地址以及创建者发送过的交易数量 (nonce) 来计算确定的，sender 和 nonce 进行 RLP 编码，然后用 Keccak-256 进行 hash 计算。
以太坊中交易的类型
简单支付交易 - 以比特币的账户间转账，从知己账户转到对方账户，不涉及智能合约，无需 动用以太坊虚拟机，“耗油”也最少。
存证交易 - 在简单支付交易中把支付额设置成 0，需要存证的内容写在 data 字段中，就构成了存证交易。当然，也可以为存证机制专门部署一个智能合约，以后要存证时就调用这个合约，这样可以为存证增添一些附加的操作。因为是 0 支付，对方账户就无关紧要，但 不能是 0。
合约部署交易 - 将对方地址设置成 0，就可以将智能合约的程序部署到以太坊网络中，实际上是所有的验证节点上，一旦交易记录进入区块链即部署生效，以太坊网络会在交易“收据”中返回新建合约账户的地址。
合约调用交易 - 对智能合约的调用，依具体智能合约的不同，又可以分为以下几种：
复杂支付交易，更确切地说是数字资产转移交易，这是对智能合约中代表着数字资产的 Token 的转移。如果用 Token 实现各种虚拟货币，这样的转移就成了采用某种自定义虚拟货币的支付。
查询交易，查询是区块链网络中常用的操作。以太坊网络中提供了若干“系统合约”，即由系统提供、无需用户部署的合约，其中之一就是用于查询。
其它(智能合约调用)交易，如在电子商务，电子政务等领域的应用。
交易的数据结构
transaction
]byte[] hash; //对 RLP 编码之后的交易请求 Tx 的 Hash 值。
]byte[] nonce; //实质上是 Tx 的序号，用以防止对同一交易请求的重复处理。
]byte[] value; //支付币值
]byte[] receiveAddress; // 接受者地址
]byte[] gasPrice; //油价
]byte[] gasLimit; // 最大油量
]byte[] data;
]Integer chainId; // 链 id
]ECDSASignature signature; // 签名
]byte[] sendAddress // 发送者地址
其中 data 在支付交易中可以作为付款说明，如果 value=0，那么它可以作为存证交易（但是对方账户不能是 0），在部署交易中（对方账户地址为 0）表示合约本身，在合约调用交易中，含有函数名和参数。

出结果是一样的