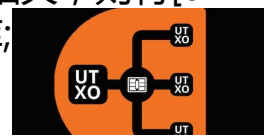
SPV是什么？SPV是“Simplified Payment Verification”(简单支付验证)的缩写。中本聪论文简要地提及了这一概念，指出：不运行完全节点也可验证支付，用户只需要保存所有的block header就可以了。用户虽然不能自己验证交易，但如果能够从区块链的某处找到相符的交易，他就可以知道网络已经认可了这笔交易，而且得到了网络的多少个确认。

按照中本聪的原文，这里有个细节需要注意，SPV指的是“支付验证“，而不是“交易验证”。这两种验证有很大区别。“交易验证”非常复杂，涉及到验证是否有足够余额可供支出、是否存在双花、脚本能否通过等等，通常由运行完全节点的矿工来完成。“支付验证”则比较简单，只判断用于“支付”的那笔交易是否已经被验证过，并得到了多少的算力保护(多少确认数)。考虑这样一种情况，A收到来自B的一个通知，B声称他已经从某某账户中汇款一定数额的钱给了A。去中心方式下，没有任何人能证明B的可靠。接到这一通知，A如何能判断B所说的是真的呢?在比特币系统中，这一通知是以一个固定格式的“交易”来实现的，该交易中包含B的汇款账户支票、B的签名、汇给A的金额以及A的地址。如果A想本人亲自验证这笔交易，首先，A要遍历区块链账本，定位到B的账户上，这样才能查看B所给的账户支票上是否曾经有足够的金额;接下来，A要遍历后续的所有账本，看B是否已经支出了这个账户支票上的钱给别人(是否存在双花欺骗);然后还要验证脚本来判断B是否拥有该账户的支配权。这一过程要求A必须得到完整的区块链才行。但是，如果A只想知道这笔支付是否已经得到了验证(如果验证了就发货)，他可以依赖比特币系统来快速验证。即，检查发生此项支付的那笔交易是否已经收录于区块链中，并得到了多少个确认。原理：block header中有三个关键字段，一是prev_block_hash(前一区块的hash值，确保了区块链所记录的交易次序);二是bits(当前区块的计算难度)，三是merkle_root_hash(借助merkle tree算法，确保收录与区块中所有交易的真实性)。验证某个交易是否真实存在时，理论上，用户可以通过以下方式进行验证：(为了简化模型，我们假设用tx_hash来定位block。这种方法有被“交易可锻性”攻击的风险，实际应用中可以根据output_point来定位。)0.
从网络上获取并保存最长链的所有block header至本地;1.
计算该交易的hash值tx_hash;2. 定位到包含该tx_hash所在的区块，验证block

header是否包含在已知的最长链中;3. 从区块中获取构建merkle tree所需的hash值;4. 根据这些hash值计算merkle_root_hash;5. 若计算结果与block header中的merkle_root_hash相等，则交易真实存在。6. 根据该block header所处的位置，确定该交易已经得到多少个确认。优点：极大地节省存储空间。减轻终端用户的负担。无论未来的交易量有多大，block header的大小始终不变，只有80字节。按照每小时6个的出块速度，每年产出52560个区块。当只保存block header时，每年新增的存储需求约为4兆字节，100年后累计的存储需求仅为400兆，即使用户使用的是最低端的设备，正常情况下也完全能够负载。问题：如何才能通过tx_hash定位到该交易所在的区块? 以往的比特币协议中缺少对此的支持。比特币钱包在进一步讨论SPV的实现之前，先要说明一下比特币钱包存放的是什么，钱包和私钥之间是什么关系?既然用到“钱包”一词，那么应该与我们日常生活中使用的钱包有一定的相似之处。为了更直观说明，我们与日常生活中所使用的钱包做一下对比。

日常生活中里面存放的可能是纸币、支票、印鉴等等(为了简化说明，我们把银行卡排除在外，使用银行卡涉及到很多中间环节，增加表述上的复杂度)。用纸币购物时，1. 从钱包中凑足若干张不同面值的纸币，计算总面值是否大于所需金额以及应找回多少零钱;2. 将这些纸币直接交给卖方;3. 卖方验证这些纸币的真伪;4. 卖方计算这些纸币的面值是否大于或等于商品价格，并找回相应的零钱。5. 将收到的零钱放回钱包。比特币的钱包里存放的相当于是一张张标有面值的“一次性支票”和对应的“印鉴”。支付时，1. 用户从钱包中取出若干张“一次性支票”，自己计算总面值是否大于所需金额以及应找回多少零钱，注意要扣除比特币系统所收取的手续费;2. 给卖方开一张支票，注明卖方地址和支付金额;如果需要找零，给自己开一张找零支票(写上自己的地址和找零金额);3. 在每张从钱包中取出的支票上加盖对应的印鉴，确认支付权;4. 将这些票据提交给比特币系统，比特币系统验证支票的真伪和支付是否有效。5. 若比特币系统验证通过，收款方将收到的支票放入钱包。用户则将自己钱包中的已支付的支票丢弃(这些支票已经被比特币系统视为无效了，无法继续使用)，即使是刚接触比特币的人，估计也能猜出“印鉴”指的是“私钥”。但“一次性支票”是什么?比特币系统中，这种“一次性支票”的术语是UTXO，全称是Unspent

Transaction Outputs(未花费的交易输出)。区块链是一个收录所有历史交易(Transaction)的总帐，每个区块(block)中包含若干笔交易记录。每个交易记录由两部分构成：资金来源(可以有多个来源)和资金去向(可以有多个去向)，术语为Tx_in(交易输入)和Tx_out(交易输出)。也就是说，每笔交易TX包含有若干个Tx_in和若干个Tx_out。除创世区块中的交易(genesis block)外，每笔交易必须要有资金来源。资金来源有两种，一种是挖矿奖励(依照固定算法实现的货币发行)，出现在每个block的第一笔交易中;另一种是先前的交易中未曾使用的某个Tx_out(交易输出)，即UTXO。支出方要出示证据来证明自己对该Tx_out拥有所有权，而比特币系统则要验证该Tx_out是否真的未被花费(是否是UTXO)以及支出方是否有权将其花费。资金去向(TX_out)包含两个部分，一是传递的金额，二是支配权(谁可以动用)。取款权通过比特币的脚本系统来实现。若收款方地址是以1开头的普通地址，则脚本中会包含地址所对应公钥的hash值(hash160)，动用款项时一般需要用对应的私钥进行签名;若收款方地址是以3开头的多重签名地址，则脚本中会包含某个特定脚本的hash值(hash160)，动用款项时，一般需要依照特定的脚本，用多个私钥来签名。用户钱包中的比特币实际上是用户拥有支配权的、且尚未花费的Tx_out中记录的金额总和，即用户可支配的所有UTXO金额之和。完整的钱包中应存有若干个UTXO和支配每个UTXO时所对应的私钥。当然，有时从安全角度出发，可能会把钱包划分为两个部分，在线钱包中只有UTXO，而离线钱包只存私钥。但是，用户怎么才能把自己的所有UTXO都放到钱包中呢?用户如何收录自己的UTXO(一)去中心化方式：实现方法：1. 在本地建立一个用于存储UTXO的数据库;2. 设置区块扫描起始点(区块链上的扫描起始高度)，从该点开始，依次下载该点之后所有区块(block)的完整数据。3. 解析每个block的所有TX数据，依次读取每个Tx_in的prev_Tx_out([tx hash] + [tx_out的序号])，检索UTXO数据库中是否存在这个Tx_out，如果有，则从UTXO数据库中删除(或标记删除)。4. 依次解析每个Tx_out的脚本，若与用户相关，则将[tx hash] + [Tx_out的序号]以及整个tx_out的内容记录到UTXO数据库;



备注：如果钱包中只有新创建的私钥，可以从最新的区块开始扫描(由于私钥发生碰撞的可能性可以视为0.在你告知他人比特币地址之前，该私钥对应的地址上不会有任何收入)优点：不依赖于信任;数据准确。缺点：速度慢，需要从比特币网络下载大量数据，对网络造成的压力大。(二)中心化方式：1. 某个中心化机构(或个人)运行完整的比特币节点，建立一个收录所有UTXO的数据库。2. 用户用中心化机构提供的api来请求与自己有关的UTXO数据。优点：速度快，不拖累比特币网络;缺点：依赖于信任;数据不一定准确(有可能中心化服务器出现故障，或是与中心服务器的会话被劫持，数据遭篡改)四、瘦客户端、SPV轻钱包和SPV节点是什么?瘦客户端：参考了SPV的机制，在监听收款地址时，客户端在本地只需保存与用户可支配交易相关的数据。因为本地没有完整的区块链，缺少发送方的相关数据，客户端无

法亲自验证交易是否合法，只能判断交易是否是被收录，并且得到了几个确认。这与SPV有很多相似之处，因而很多场合下这种瘦客户端也常被成为是“SPV客户端”，不过，与SPV的区别是，在去中心化方式下，这些客户端仍需下载每个新区块的全部数据并进行解析，只是无需在本地保存全部数据而已。“轻钱包”是用瘦客户端模式实现的钱包，因为不存储完整区块链，就涉及到如何获取UTXO的问题。不同的开发者可能有各自的实现方法，但从效率上考虑，往往多用中心化的方式来实现。SPV节点：支持使用布隆过滤器(Bloom filter)在快速检索并返回相关数据的节点。SPV在实现上涉及到一个问题，如何才能通过交易特征值(比如tx_hash)来定位到该支付交易所在的区块?用中心化方式来实现很好解决，但用去中心化就不那么简单了，因为以往的比特币系统协议中缺少对SPV的支持。去中心方式下获取数据必须做到以下两点：1. 客户端和节点间采用公认的协议通信;2. 数据真实性的自验证——客户端无需信任节点是否是诚实节点，返回的数据本身可以证明该数据的真实性。原有协议中，可以通过getheaders命令来获取block headers，可以通过getdata命令支持获取指定的block, 但不支持通过tx_hash反向查找所在的block。为了定位block，客户端往往不得不下载整个区块链。新的比特币协议中增加了Bloom filter的功能，Bloom filter解决了客户端检索的问题，原理是Bloom filter可以快速判断出某检索值一定不存在于某个指定的集合，从而可以过滤掉大量无关数据，减少客户端不必要的下载量。这样的节点可以为去中心化方式SPV查询提供必要的支持。前文提到，SPV的用途是验证某个支付是否确实存在，并得到多少个确认。而钱包的用途则是用于管理自己的资产以及进行支付。简言之，SPV的应用场合往往是为发货做准备(知道钱到帐了)，“轻钱包”的应用场合往往是数钱或花钱。虽然“轻钱包”中部分借鉴了SPV的机制，但和SPV是完全不用的应用方向，直接把这两个词连起略显牵强。这种钱包要么采用中心化的方式——提高了效率，但引入了信任的风险;要么采用去中心化方式——无需信任，但效率低，且增加网络的负担。SPV节点的出现使以去中心化方式来实现高效、低负荷的“轻钱包”成为了可能。笔者认为将基于SPV节点来实现的”轻钱包”简称为“SPV轻钱包”可能会更为合适些。