

一种区块链系统的共识方法及装置

本申请要求于 2024 年 06 月 27 日提交中国专利局、申请号为 202410844017.9，申请名称为“一种区块链系统的共识方法及装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

5 技术领域

本申请属于区块链技术领域，尤其涉及一种区块链系统的共识方法及装置。

背景技术

在区块链中，共识机制用于保证网络上的所有节点都同意网络的当前状态和交易的真实性，这对于维护区块链的安全性和完整性至关重要。参与共识的验证节点需要验证区块并对其进行签名投票，根据拜占庭容错的原理，具有法定数量 $2f+1$ 个签名投票的区块才能通过验证，可以被正确地提交并记录到区块链中。而签名法定数量 $2f+1$ 与系统中验证节点的总数量 $3f+1$ 是成正比的，当系统中的验证节点数量剧增时，每个区块需要被签名的数量也会剧增，此时签名消息的传输与验证开销会变得十分巨大且难以控制。因此，如何减少共识过程中签名消息的传输及验证成本，成为亟需解决的技术问题。

15 技术问题

本申请实施例提供了一种区块链系统的共识方法及装置，可以解决共识过程中签名消息的传输及验证成本较高的问题。

技术解决方案

第一方面，本申请实施例提供了一种区块链系统的共识方法，所述区块链系统包括多个验证节点，所述共识方法由一验证节点执行，所述共识方法包括：

在第 $n+x_2$ 次共识轮次中，引用多个提案区块，并结合客户端的当前交易，生成一后置提案区块；其中，多个提案区块是在第 n 次共识轮次中，由多个验证节点引用一前置提案区块所对应产生的，所述前置提案区块在第 $n-x_1$ 次共识轮次中产生；

将所述后置提案区块作为所述前置提案区块的证书，以共识所述前置提案区块。
25 在一种可能的实现方式中， x_1 和 x_2 相同或不同，且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

在一种可能的实现方式中，多个验证节点引用一前置提案区块对应产生提案区块，包括：

本验证节点连同其他验证节点，引用一前置提案区块对应产生提案区块。
30 在一种可能的实现方式中，引用任意一个提案区块的步骤包括：引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

在一种可能的实现方式中，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

35 在一种可能的实现方式中，将所述后置提案区块作为所述前置提案区块的证书，包括：连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

40 在一种可能的实现方式中，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

第二方面，本申请实施例提供了一种区块链系统的共识装置，包括：

后置提案区块生成模块，用于在第 $n+x_2$ 次共识轮次中，引用多个提案区块，并结合客户端的当前交易，生成一后置提案区块；其中，多个提案区块是在第 n 次共识轮次中，由多个验证节点引用一前置提案区块所对应产生的，所述前置提案区块在第 $n-x_1$ 次共识轮次

中产生；

共识模块，用于将所述后置提案区块作为所述前置提案区块的证书，以共识所述前置提案区块。

在一种可能的实现方式中，x1 和 x2 相同或不同，且 x1 和 x2 各自等于 1 或者各自不等于 1。

在一种可能的实现方式中，本验证节点用于连同其他验证节点，引用一前置提案区块对应产生提案区块。

在一种可能的实现方式中，后置提案区块生成模块具体用于引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

在一种可能的实现方式中，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

在一种可能的实现方式中，共识模块具体用于连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

在一种可能的实现方式中，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

第三方面，本申请实施例提供了一种终端设备，包括：存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序，所述处理器执行所述计算机程序时实现如上所述的方法。

第四方面，本申请实施例提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被处理器执行时实现如上所述的方法。

有益效果

本申请提供一种共识方法、装置、终端设备及存储介质，通过引用关系代替传统共识方法中的签名消息，即验证节点引用了某个提案区块，则表示该验证节点验证通过了该提案区块。故利用本申请提供的共识方法，在共识过程中不再需要传递大量的签名消息，而是通过在不同共识轮次产生的区块之间建立引用关系，以对某个区块进行共识。如此一来，减少了共识过程中传输和验证签名消息的开销。

附图说明

为了更清楚地说明本申请实施例中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

- 图 1 是本申请一实施例提供的共识方法的流程示意图；
- 图 2 是本申请实施例 1 的示意图；
- 图 3 是本申请实施例 2 的示意图；
- 图 4 是本申请实施例 3 的示意图；
- 图 5 是本申请实施例提供的引用关系示意图；
- 图 6 是本申请实施例 4 的示意图；
- 图 7 是本申请实施例 5 的示意图；
- 图 8 是本申请实施例 6 的示意图；
- 图 9 是本申请实施例 7 的示意图；
- 图 10 是本申请实施例 8 的示意图；
- 图 11 是本申请实施例 9 的示意图；
- 图 12 是本申请实施例 10 的示意图；
- 图 13 是本申请实施例 11 的示意图；
- 图 14 是本申请实施例提供的共识装置的结构示意图；
- 图 15 是本申请实施例提供的终端设备的结构示意图。

本申请的实施方式

以下描述中，为了说明而不是为了限定，提出了诸如特定系统结构、技术之类的具体细节，以便透彻理解本申请实施例。然而，本领域的技术人员应当清楚，在没有这些具体细节的其它实施例中也可以实现本申请。在其它情况中，省略对众所周知的系统、装置、电路以及方法的详细说明，以免不必要的细节妨碍本申请的描述。

应当理解，当在本申请说明书和所附权利要求书中使用时，术语“包括”指示所描述特征、整体、步骤、操作、元素和/或组件的存在，但并不排除一个或多个其它特征、整体、步骤、操作、元素、组件和/或其集合的存在或添加。

还应当理解，在本申请说明书和所附权利要求书中使用的术语“和/或”是指相关联列出的项中的一个或多个的任何组合以及所有可能组合，并且包括这些组合。

如在本申请说明书和所附权利要求书中所使用的那样，术语“如果”可以依据上下文被解释为“当...时”或“一旦”或“响应于确定”或“响应于检测到”。类似地，短语“如果确定”或“如果检测到[所描述条件或事件]”可以依据上下文被解释为意指“一旦确定”或“响应于确定”或“一旦检测到[所描述条件或事件]”或“响应于检测到[所描述条件或事件]”。

另外，在本申请说明书和所附权利要求书的描述中，术语“第一”、“第二”、“第三”等仅用于区分描述，而不能理解为指示或暗示相对重要性。

在本申请说明书中描述的参考“一个实施例”或“一些实施例”等意味着在本申请的一个或多个实施例中包括结合该实施例描述的特定特征、结构或特点。由此，在本说明书中的不同之处出现的语句“在一个实施例中”、“在一些实施例中”、“在其他一些实施例中”、“在另外一些实施例中”等不是必然都参考相同的实施例，而是意味着“一个或多个但不是所有的实施例”，除非是以其他方式另外特别强调。术语“包括”、“包含”、“具有”及它们的变形都意味着“包括但不限于”，除非是以其他方式另外特别强调。

在区块链系统中，必须要通过共识协议来保证系统中分布式节点的一致性，共识协议中的各个验证节点也被称为共识节点，需要验证区块并对其进行签名投票。根据拜占庭容错的原理，具有法定数量 $2f+1$ 个签名投票的区块才能通过验证，可以被正确地提交并记录到区块链中。而签名法定数量 $2f+1$ 与系统中验证节点的总数量 $3f+1$ 是成正比的，当系统中的验证节点数量剧增时，每个区块需要被签名的数量也会剧增，此时签名消息的传输与验证开销会变得十分巨大且难以控制。因此，如何减少共识过程中签名消息的传输及验证成本，成为亟需解决的技术问题。

本申请提供一种共识方法、装置、终端设备及存储介质，通过引用关系代替传统共识方法中的签名消息，即验证节点引用了某个提案区块，则表示该验证节点验证通过了该提案区块。故利用本申请提供的共识方法，在共识过程中不再需要传递大量的签名消息，而是通过检查引用关系，确定某个提案区块是否被足够数量的验证节点验证通过。如此一来，减少了共识过程中传输和验证签名消息的开销。

为了说明本申请的技术方案，下面通过具体实施例来说明。

参阅图 1 示出的区块链系统的共识方法的一个实施例的流程，作为示例而非限定，包括以下步骤：

步骤 S100: 在第 $n+x2$ 次共识轮次中, 引用多个提案区块, 并结合客户端的当前交易, 生成一后置提案区块; 其中, 多个提案区块是在第 n 次共识轮次中, 由多个验证节点引用一前置提案区块所对应产生的, 前置提案区块在第 $n-x1$ 次共识轮次中产生;

步骤 S200: 将后置提案区块作为前置提案区块的证书, 以共识前置提案区块。

5 需要说明的是, 本申请的区块链系统包括多个验证节点, 本申请实施例提供的共识方法由验证节点执行。验证节点在区块链网络中是经过特定认证和授权的节点, 它们具有参与共识过程、验证交易和打包区块的权限。具体的, 验证节点负责的任务包括执行共识算法, 如上述方案中所描述的, 它们会在特定的共识轮次中引用提案区块、生成后置提案区块, 并参与对前置提案区块的共识。验证节点还会对交易进行验证, 确保交易的合法性、
10 有效性以及符合区块链的协议规则。此外, 验证节点还会根据共识算法的结果, 打包合法的交易生成新的区块, 并尝试将这些区块添加到区块链中。

可以理解的是, 区块链系统通常会设置一个阈值, 要求一定数量的验证节点验证通过相同的提案区块, 才能将该提案区块添加到区块链中。该过程确保了网络中的大多数验证节点都对新的区块内容达成了共识。在传统的共识过程中, 某个验证节点若验证通过了一个提案区块需要对其进行签名并广播给系统中的其他验证节点, 其他验证节点需要对签名进行验证, 因此在共识过程中, 需要传递和验证大量的签名信息。
15

而在本申请提供的共识方法中, 若验证节点引用了之前共识轮次中生成的提案区块, 则表示该验证节点已经验证通过了该提案区块。即在本申请中, 用引用关系代替了传统共识方法中的签名消息, 故共识过程不在需要传递大量的签名消息, 而是通过检查引用关系, 确定某个提案区块是否被足够数量的验证节点验证通过。
20

需要说明的是, 在本申请中, 引用关系的确定建立在验证通过的基础上, 即需要对某一区块进行验证, 以确保提案区块的内容符合区块链的规则和网络的安全要求, 在验证通过后, 才能与其建立引用关系。

示例性的, 验证节点可以通过检查提案区块签名、区块数据内容、区块哈希、区块时间戳来验证提案区块的正确性。例如, 验证节点可以检查区块内的交易数据是否有效和合法, 包括检查交易的字段是否完整、格式是否符合区块链的协议规则等, 对于每一笔交易, 验证节点可能还需要检查交易的输入和输出是否平衡。又如, 验证节点还可以计算提案区块的哈希值, 并与区块头中存储的哈希值进行比较, 确保区块在传输过程中没有被篡改。又或者, 验证节点可以检查提案区块中的时间戳是否合理, 时间戳用于记录区块创建的时间, 可以帮助验证节点检测潜在的双重支付或重复交易。
25
30

因此, 也可以理解为引用某个提案区块即表示对该提案区块的认可。在后续的共识过程中, 可以通过检查提案区块间的引用关系, 来判断某个提案区块是否被足够数量的验证节点引用, 如果引用该提案区块的验证节点数量满足一定要求, 即可认为该提案区块得到了足够多的认可, 即该提案区块是可信的。

35 相比于传统的共识方法, 每个提案区块至少需要携带 $2f+1$ 个签名, 当系统中的验证节点数量剧增时, 每个区块需要被签名的数量也会剧增, 而本申请提供的共识方法是通过引用关系来表示对某个提案区块的认可, 不是通过签名来共识某个提案区块, 因此在共识过程中不再需要传递大量的签名消息, 如此一来, 减少了共识过程中传输和验证签名消息的开销。

需要说明的是，在步骤 S100 中，验证节点在生成后置提案区块时，还需要结合客户端的当前交易。其中，交易指的是在区块链网络上发生的、由用户（即客户端）发起的一系列操作或请求，这些操作或请求会改变区块链上的状态或数据。这些交易可能涉及智能合约的调用、数据的写入或更新等。

需要说明的是，引用提案区块的实现过程基于一有向无环图（Directed Acyclic Graph, DAG），即区块链系统不仅仅是一个简单的线性结构，而是采用了一种更为复杂的图结构来组织和引用提案区块。

为了更好的理解本申请，首先对有向无环图（DAG）进行简单介绍，有向无环图是一种包含方向性边且没有环路的图。在区块链系统中，有向无环图可以被用来表示提案区块之间的引用关系，其中每个提案区块都可以指向一个或多个先前的提案区块，形成一个复杂的网络结构。

具体的，区块链系统维护了一个预存储的有向无环图，这个图记录了所有提案区块以及它们之间的引用关系。当一个验证节点想要引用先的提案区块时，会查询这个预存储的 DAG，找到合适的区块进行引用。引用的依据可能是区块的年龄（即生成时间）、包含的交易数量、区块的质量或其他自定义的策略。

验证节点在构建新的提案区块时，会根据当前的交易数据和 DAG 中的引用关系，选择一个或多个先前的提案区块进行引用。当新的提案区块被成功添加到 DAG 中，并且被网络中的其他验证节点所确认时，DAG 的结构就会相应地更新，以反映新的引用关系和区块的添加。

通过有向无环图引用提案区块的共识方法，可以使得区块链系统在处理大量交易和快速生成区块时更加灵活和高效。验证节点可以通过并行处理多个提案区块，并在 DAG 中找到合适的路径来达成共识，从而提高了整个系统的吞吐量和性能。

在一种可能的实现方式中， x_1 和 x_2 相同或不同，且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

需要说明的是， x_1 可以为 1 或者大于 1 的正整数，若 x_1 为 1，则表示产生提案区块的轮次与产生前置提案区块的轮次是相邻的，若 x_1 不为 1，则表示产生提案区块的轮次与产生前置提案区块的轮次是不相邻的。同理， x_2 也可以为 1 或者大于 1 的正整数，若 x_2 为 1，则表示产生提案区块的轮次与产生后置提案区块的轮次是相邻的，若 x_2 不为 1，则表示产生提案区块的轮次与产生后置提案区块的轮次是不相邻的。

下面以具体的实施例解释上述内容。

实施例 1：参阅图 2，在本实施例中， $x_1=1$ ， $x_2=1$ ，即在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 B_{n-1} ，故将后置提案区块 B_{n+1} 作为前置提案区块 B_{n-1} 的证书，用以共识前置提案区块 B_{n-1} 。

实施例 2：参阅图 3，在本实施例中， $x_1=1$ ， $x_2 \neq 1$ ，具体的，以 x_2 等于 2 为例，即在第 $n+2$ 次共识轮次中，通过间接引用的方式，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+2} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共

识轮次中产生的前置提案区块 B_{n-1} ，故将后置提案区块 B_{n+2} 作为前置提案区块 B_{n-1} 的证书，用以共识前置提案区块 B_{n-1} 。

需要说明的是，由于第 $n+1$ 次共识轮次中产生的区块与第 n 次共识轮次中产生的提案区块存在引用关系，因此后置提案区块 B_{n+2} 可以通过引用第 $n+1$ 次共识轮次中产生的区块而间接引用到第 n 次共识轮次中产生的提案区块。

示例性的，由于区块 A_{n+1} 与提案区块 A_n 存在引用关系，因此后置提案区块 B_{n+2} 可以通过引用区块 A_{n+1} 来实现对提案区块 A_n 的引用。同理，由于区块 C_{n+1} 与提案区块 B_n 存在引用关系，因此后置提案区块 B_{n+2} 可以通过引用区块 C_{n+1} 来实现对提案区块 B_n 的引用。同理，由于区块 D_{n+1} 与提案区块 C_n 存在引用关系，因此后置提案区块 B_{n+2} 可以通过引用区块 D_{n+1} 来实现对提案区块 C_n 的引用。

实施例 3：参阅图 4，在本实施例中， $x_1 \neq 1$ ， $x_2 = 1$ ，具体的，以 x_1 等于 2 为例，即在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均通过间接引用的方式引用了第 $n-2$ 次共识轮次中产生的前置提案区块 B_{n-2} ，故将后置提案区块 B_{n+1} 作为前置提案区块 B_{n-2} 的证书，用以共识前置提案区块 B_{n-2} 。

需要说明的是，由于第 $n-1$ 次共识轮次中产生的区块与第 $n-2$ 次共识轮次中产生的前置提案区块存在引用关系，因此提案区块可以通过引用第 $n-1$ 次共识轮次中产生的区块而间接引用到第 $n-2$ 次共识轮次中产生的前置提案区块。

示例性的，由于区块 A_{n-1} 与提案区块 B_{n-2} 存在引用关系，因此提案区块 A_n 可以通过引用区块 A_{n-1} 来实现对前置提案区块 B_{n-2} 的引用。同理，由于区块 B_{n-1} 与前置提案区块 B_{n-2} 存在引用关系，因此提案区块 C_n 可以通过引用区块 B_{n-1} 来实现对前置提案区块 B_{n-2} 的引用。同理，由于区块 C_{n-1} 与前置提案区块 B_{n-2} 存在引用关系，因此提案区块 D_n 可以通过引用区块 C_{n-1} 来实现对前置提案区块 B_{n-2} 的引用。

需要说明的是，通过上述实施例可以看出，引用包括直接引用和间接引用，直接引用即引用上一轮次产生的某个区块，示例性的，如图 3 中区块 A_n 引用区块 B_{n-1} 、区块 B_{n+2} 引用区块 C_{n+1} 或者区块 A_{n+1} 引用区块 A_n ，间接引用即引用不相邻的轮次中产生的提案区块，示例性的，如图 3 中区块 B_{n+2} 引用区块 A_n ，即是区块 B_{n+2} 通过区块 A_{n+1} 间接引用区块 A_n 。

在一种可能的实现方式中，多个验证节点引用一前置提案区块对应产生提案区块，包括：

本验证节点连同其他验证节点，引用一前置提案区块对应产生提案区块。

需要说明的是，在第 n 次共识轮次中，包括引用一前置提案区块以产生提案区块的过程，该过程的执行主体可以包含本验证节点，也可以不包含。

以上述实施例 1 和实施例 3 对此进行说明。参阅图 2 所示的实施例 1 的示意图可知，在第 n 次共识轮次中，本验证节点 (Validator B) 就参与了提案区块的生成过程，即 Validator B 在第 n 次共识轮次中引用了前置提案区块 B_{n-1} 生成提案区块 B_n 。而在图 4 所示的实施例 3 中，在第 n 次共识轮次中，本验证节点 (Validator B) 没有参与提案区块的生成过程，即 Validator B 在第 n 次共识轮次中没有引用前置提案区块生成提案区块 B_n 。

故在本申请实施例中，引用一前置提案区块以产生提案区块的过程的执行主体可以包含本验证节点，也可以不包含。

在一种可能的实现方式中，引用任意一个提案区块的步骤包括：引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

需要说明的是，提案区块包含：交易列表、当前共识轮次、提案签署人、提案签名以及与之前轮次中生成的提案区块之间的引用关系。参阅图 5，每个提案区块包含交易列表 transactions、当前共识轮次 round、提案签署人 author、提案签名 signature 以及对之前轮次提案区块的引用关系 parents。

在一种可能的实现方式中，哈希索引可以根据当前共识轮次、提案签署人、交易列表以及引用关系生成。参阅图 5，在图示情况中，哈希索引包括 h1、h2 和 h3，其中 h1 根据当前共识轮次 r、提案签署人 v1、交易列表 txs1 以及引用关系 p1 通过哈希函数计算得到；h2 根据当前共识轮次 r、提案签署人 v2、交易列表 txs2 以及引用关系 p2 通过哈希函数计算得到；h3 根据当前共识轮次 r、提案签署人 v3、交易列表 txs3 以及引用关系 p3 通过哈希函数计算得到。

需要说明的是，引用的基础是提案区块已被验证通过，包括对提案区块签署人与签名的对应关系的验证，以及对引用关系正确性的验证。

具体的，需要验证提案区块签署人与签名的对应关系。本申请通过多个共识轮次进行共识，每个诚实的验证节点都会在每一共识轮次将自己在该轮中的唯一一个提案区块签名后广播给其他验证节点，以证明该区块确实是由其生成的。验证节点会检查签名与签署人是否匹配，以及签名是否有效（没有被篡改或伪造），如此一来，可以防止恶意节点伪造区块或篡改交易。

另外，还需要验证引用关系的正确性。由于每个提案区块会引用若干个（不小于法定数量）提案区块，因此验证节点需要检查引用的提案区块的哈希值是否正确，以确保区块链的完整性和连续性。

此外，在生成后置提案区块时，还需要结合客户端的当前交易，对于该交易，也需要进行验证。具体的，可以验证当前交易的交易列表，包括验证每笔交易的有效性、发送者和接收者的身份以及交易金额是否正确等，确保只有合法和有效的交易才能被添加到区块链中。

在一种可能的实现方式中，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

需要说明的是，在上述实施例 1、实施例 2 和实施例 3 中，前置提案区块与后置提案区块均由相同的验证者产生。而在一些可能的实现方式中，对应上述三种情况，前置提案区块与后置提案区块也可由不同的验证者产生。下面以具体的实施例对此进行解释。

实施例 4：参阅图 6，在本实施例中，x1=1，x2=1，且前置提案区块与后置提案区块由不同验证节点产生。

需要说明的是，本实施例与实施例 1 的区别在于，前置提案区块与后置提案区块由不同验证节点产生。在实施例 1 中，前置提案区块 B_{n-1} 与后置提案区块 B_{n+1} 均由验证节点 Validator B 产生，而在实施例 4 中，前置提案区块 D_{n-1} 由验证节点 Validator D 产生，后置提案区块 B_{n+1} 由验证节点 Validator B 产生。

5 具体的，在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 D_{n-1} ，故将后置提案区块 B_{n+1} 作为前置提案区块 D_{n-1} 的证书，用以共识前置提案区块 D_{n-1} 。

10 实施例 5：参阅图 7，在本实施例中， $x_1=1$ ， $x_2 \neq 1$ ，且前置提案区块与后置提案区块由不同验证节点产生。

需要说明的是，本实施例与实施例 2 的区别在于，前置提案区块与后置提案区块由不同验证节点产生。在实施例 2 中，前置提案区块 B_{n-1} 与后置提案区块 B_{n+1} 均由验证节点 Validator B 产生，而在实施例 5 中，前置提案区块 C_{n-1} 由验证节点 Validator C 产生，后置提案区块 B_{n+2} 由验证节点 Validator B 产生。

15 具体的，在第 $n+2$ 次共识轮次中，通过间接引用的方式，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+2} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 C_{n-1} ，故将后置提案区块 B_{n+2} 作为前置提案区块 C_{n-1} 的证书，用以共识前置提案区块 C_{n-1} 。

20 实施例 6：参阅图 8，在本实施例中， $x_1 \neq 1$ ， $x_2=1$ ，且前置提案区块与后置提案区块由不同验证节点产生。

需要说明的是，本实施例与实施例 3 的区别在于，前置提案区块与后置提案区块由不同验证节点产生。在实施例 3 中，前置提案区块 B_{n-2} 与后置提案区块 B_{n+1} 均由验证节点 Validator B 产生，而在实施例 6 中，前置提案区块 D_{n-2} 由验证节点 Validator D 产生，后置提案区块 B_{n+1} 由验证节点 Validator B 产生。

25 具体的，在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均通过间接引用的方式引用了第 $n-2$ 次共识轮次中产生的前置提案区块 D_{n-2} ，故将后置提案区块 B_{n+1} 作为前置提案区块 D_{n-2} 的证书，用以共识前置提案区块 D_{n-2} 。

30 在一种可能的实现方式中，将所述后置提案区块作为所述前置提案区块的证书，包括：连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

需要说明的是，证书代表了验证节点对待提交提案区块的确认和接受。每个验证节点都可以为一个或多个待提交提案区块生成证书，这些证书累积起来，形成了一个提案区块被接受程度的量化指标。为了确保区块链的稳定性和安全性，只有当待提交提案区块的证书数量达到或超过预设值时，该提案区块才会被认定为得到了足够的支持，从而被提交到目标区块链中。

35

还需要说明的是，证书数量的预设置可以根据区块链网络的具体情况进行调整。例如，在区块链网络规模较小或验证节点数量较少的情况下，可以适当降低证书数量的预设置；而在区块链网络规模较大或验证节点数量较多的情况下，则可以适当提高证书数量的预设置。

5 在本申请实施例中，对于同一个前置提案区块，可以有多个后置提案区块作为它的证书，且多个证书可以由不同的验证节点在同一个共识轮次中产生，也可以由相同的验证节点在不同轮次中产生。下面通过具体的实施例对此进行说明。

10 实施例 7：参阅图 9，在本实施例中，对于 Validator B 在第 $n-1$ 次共识轮次中产生的前置提案区块 B_{n-1} ，存在两个证书，分别由 Validator B 和 Validator D 在第 $n+1$ 次共识轮次中产生。

15 具体的，验证节点 Validator B 在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 B_{n-1} ，故将后置提案区块 B_{n+1} 作为前置提案区块 B_{n-1} 的证书，用以共识前置提案区块 B_{n-1} 。同时，验证节点 Validator D 在第 $n+1$ 次共识轮次中，也引用了第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 D_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 B_{n-1} ，故将后置提案区块 D_{n+1} 作为前置提案区块 B_{n-1} 的证书，用以共识前置提案区块 B_{n-1} 。

20 需要说明的是，在实施例 7 中，Validator D 产生的后置提案区块 D_{n+1} 和 Validator B 产生的后置提案区块 B_{n+1} ，均可以作为前置提案区块 B_{n-1} 的证书，以共识前置提案区块 B_{n-1} 。

25 在上述实施例中，一个前置提案区块具有多个后置提案区块作为它的证书，且作为证书的多个后置提案区块均在同一共识轮次中产生。需要说明的是，在一些可能的实施例中，作为证书的多个后置提案区块可能在不同的共识轮次中产生，下面通过实施例 8 对该情况进行说明。

 实施例 8：参阅图 10，在本实施例中，验证节点 Validator B 在第 $n+1$ 次共识轮次中，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 C_{n-1} ，故将后置提案区块 B_{n+1} 作为前置提案区块 C_{n-1} 的证书，用以共识前置提案区块 C_{n-1} 。

30 同时，验证节点 Validator B 在第 $n+2$ 次共识轮次中，通过间接引用的方式，引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+2} ，其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 C_{n-1} ，故也可将后置提案区块 B_{n+2} 作为前置提案区块 C_{n-1} 的证书，用以共识前置提案区块 C_{n-1} 。

35 因此，在本实施例中，后置提案区块 B_{n+1} 和后置提案区块 B_{n+2} 均作为前置提案区块 C_{n-1} 的证书，且后置提案区块 B_{n+1} 和后置提案区块 B_{n+2} 在不同的共识轮次中产生。

 在上述实施例 8 中，两个作为证书的后置提案区块在不同轮次中产生，但是均由相同的验证节点 Validator B 产生。需要说明的是，在一些可能的实施例中，多个作为证书的后置提案区块也可由不同的验证节点在不同的共识轮次中产生。下面通过实施例 9 对此进行说明。

参阅图 11, 在实施例 9 中, 验证节点 Validator D 在第 $n+1$ 次共识轮次中, 引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 D_{n+1} , 其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 C_{n-1} , 故将后置提案区块 D_{n+1} 作为前置提案区块 C_{n-1} 的证书, 用以共识前置提案区块 C_{n-1} 。

- 5 同时, 验证节点 Validator B 在第 $n+2$ 次共识轮次中, 通过间接引用的方式, 引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+2} , 其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 C_{n-1} , 故也可将后置提案区块 B_{n+2} 作为前置提案区块 C_{n-1} 的证书, 用以共识前置提案区块 C_{n-1} 。

10 因此, 在本实施例中, 后置提案区块 D_{n+1} 和后置提案区块 B_{n+2} 均作为前置提案区块 C_{n-1} 的证书, 且后置提案区块 D_{n+1} 和后置提案区块 B_{n+2} 在不同的共识轮次中由不同的验证节点产生。

还需要说明的是, 在一些可能的实施例中, 同一个后置提案区块还可能同时作为多个前置提案区块的证书。下面通过实施例 10 对该情况进行说明。

15 参阅图 12, 在实施例 10 中, 前置提案区块包括 B_{n-1} 和 D_{n-1} , 根据图示的引用关系可知, 后置提案区块 B_{n+1} 既可以作为前置提案区块包括 B_{n-1} 的证书, 也可以作为前置提案区块包括 D_{n-1} 的证书。

具体的, 在第 $n+1$ 次共识轮次中, 验证节点 Validator B 引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} , 其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 D_{n-1} , 故将后置提案区块 B_{n+1} 作为前置提案区块 D_{n-1} 的证书, 用以共识前置提案区块 D_{n-1} 。并且, 提案区块 A_n 、 B_n 、 C_n 均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 B_{n-1} , 故后置提案区块 B_{n+1} 也可以作为前置提案区块 B_{n-1} 的证书, 用以共识前置提案区块 B_{n-1} 。故在本实施例中, 后置提案区块 B_{n+1} 同时作为前置提案区块 B_{n-1} 和前置提案区块 D_{n-1} 的证书。

25 需要说明的是, 对于同一个后置提案区块作为多个前置提案区块的情况, 仍然不限制前置提案区块与后置提案区块必须由同一个验证节点产生。下面以实施例 11 对此情况进行说明。

30 参阅图 13, 在第 $n+1$ 次共识轮次中, 验证节点 Validator B 引用第 n 次共识轮次中产生的提案区块 A_n 、 B_n 、 C_n 以生成后置提案区块 B_{n+1} , 其中 A_n 、 B_n 、 C_n 三个提案区块均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 D_{n-1} , 故将后置提案区块 B_{n+1} 作为前置提案区块 D_{n-1} 的证书, 用以共识前置提案区块 D_{n-1} 。并且, 提案区块 A_n 、 B_n 、 C_n 均引用了第 $n-1$ 次共识轮次中产生的前置提案区块 A_{n-1} , 故后置提案区块 B_{n+1} 也可以作为前置提案区块 A_{n-1} 的证书, 用以共识前置提案区块 A_{n-1} 。故在本实施例中, 后置提案区块 B_{n+1} 同时作为前置提案区块 A_{n-1} 和前置提案区块 D_{n-1} 的证书。

35 在一种可能的实现方式中, 所述提案区块的数量不小于 $2f+1$, 其中 f 为在第 n 次共识轮次中, 不诚实的验证节点的数量, 且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

需要说明的是, 将后置提案区块作为前置提案区块的证书需要满足一定的条件, 即后置提案区块所引用的提案区块的数量需要满足 $2f+1$ 的条件, 该条件基于拜占庭容错算法 (Byzantine Fault Tolerant), 拜占庭容错算法是面向拜占庭问题的容错算法, 解决的是在网络通信可靠但节点可能故障和作恶情况下如何达成共识的问题。其核心理论是系统中的

总节点数大于等于 $3f+1$, f 是允许出现故障的节点数, 换句话说, 对于某个提案区块, 只有法定数量 (不小于 $2f+1$) 个验证节点验证通过该区块, 才能保证该提案区块的正确性。故本申请实施例要求对于前置提案区块来说, 引用其生成的提案区块数量不得小于 $2f+1$ 。

- 5 此外, 对于引用关系的检查, 可以采用深度优先遍历的方式, 以第 $n+x_2$ 次共识轮次的某个后置提案区块作为起始点, 优先遍历与其存在引用关系的提案区块, 以通过引用关系确定某个前置提案区块是否得到足够多的验证节点的支持。

10 深度优先遍历是一种用于遍历或搜索树/图的算法, 在本申请实施例中, 可以基于有向无环图进行遍历, 即如果某个后置提案区块引用了之前轮次的提案区块, 则以该后置提案区块作为起点, 采用深度优先遍历 (Depth-First Search, DFS) 的方式来追踪并验证这些被引用的区块。

下面结合深度优先遍历的原理对遍历过程进行说明。验证节点以第 $n+x_2$ 次共识轮次中生成的后置提案区块作为起始点, 接着需要选择第一个邻居节点, 由于当前区块引用了之前的某个或某些区块, 验证节点会首先选择这些被引用的区块作为第一个邻居节点进行遍历; 然后, 对于每个被引用的区块 (邻居节点), 验证节点会检查该区块是否又引用了其他的区块, 如果是, 验证节点则会递归地继续遍历这些被引用的区块, 直到没有更多的引用区块为止; 再接着, 当验证节点遍历完一个分支的所有区块后, 会回溯到上一个节点, 并继续遍历该节点的其他未被遍历的邻居节点; 当验证节点遍历完所有可达的区块, 或者达到某个预设的终止条件 (如遍历深度限制、时间限制等) 时, 遍历过程结束。如此一来, 即可确定某个前置提案区块是否得到足够多的验证节点的支持。

20 本申请实施例提供的一种区块链系统的共识装置 400, 参阅图 14, 包括:

后置提案区块生成模块 401, 用于在第 $n+x_2$ 次共识轮次中, 引用多个提案区块, 并结合客户端的当前交易, 生成一后置提案区块; 其中, 多个提案区块是在第 n 次共识轮次中, 由多个验证节点引用一前置提案区块所对应产生的, 所述前置提案区块在第 $n-x_1$ 次共识轮次中产生;

25 共识模块 402, 用于将所述后置提案区块作为所述前置提案区块的证书, 以共识所述前置提案区块。

需要说明的是, 本申请提供的共识装置 400, 用引用关系代替了传统共识方法中的签名消息, 即后置提案区块若是引用多个提案区块, 并且多个提案区块均引用了某一前置提案区块, 则可以用后置提案区块作为前置提案区块的证书, 并依据证书完成共识过程。故在利用本申请提供的共识装置 400 实现共识的过程中, 不在需要传递大量的签名消息, 而是通过引用关系, 即可确定某个前置提案区块是否被足够数量的验证节点验证通过。如此一来, 减少了共识过程中传输和验证签名消息的开销。

基于相同的发明构思, 在一种可能的实现方式中, x_1 和 x_2 相同或不同, 且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

35 需要说明的是, x_1 可以为 1 或者大于 1 的正整数, 若 x_1 为 1, 则表示产生提案区块的轮次与产生前置提案区块的轮次是相邻的, 若 x_1 不为 1, 则表示产生提案区块的轮次与产生前置提案区块的轮次是不相邻的。同理, x_2 也可以为 1 或者大于 1 的正整数, 若 x_2 为 1, 则表示产生提案区块的轮次与产生后置提案区块的轮次是相邻的, 若 x_2 不为 1, 则表示产生提案区块的轮次与产生后置提案区块的轮次是不相邻的。

基于相同的发明构思，本验证节点用于连同其他验证节点，引用一前置提案区块对应产生提案区块。

需要说明的是，在第 n 次共识轮次中，包括引用一前置提案区块以产生提案区块的过程，该过程的执行主体可以包含本验证节点，也可以不包含。

5 基于相同的发明构思，后置提案区块生成模块具体用于引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

需要说明的是，提案区块包含：交易列表、当前共识轮次、提案签署人、提案签名以及与之前轮次中生成的提案区块之间的引用关系。参阅图 5，每个提案区块包含交易列表 transactions、当前共识轮次 round、提案签署人 author、提案签名 signature 以及对之前轮次提案区块的引用关系 parents。
10

在一种可能的实现方式中，哈希索引可以根据当前共识轮次、提案签署人、交易列表以及引用关系生成。参阅图 5，在图示情况中，哈希索引包括 h1、h2 和 h3，其中 h1 根据当前共识轮次 r、提案签署人 v1、交易列表 txs1 以及引用关系 p1 通过哈希函数计算得到；h2 根据当前共识轮次 r、提案签署人 v2、交易列表 txs2 以及引用关系 p2 通过哈希函数计算得到；h3 根据当前共识轮次 r、提案签署人 v3、交易列表 txs3 以及引用关系 p3 通过哈希函数计算得到。
15

基于相同的发明构思，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

需要说明的是，在上述实施例 1、实施例 2 和实施例 3 中，前置提案区块与后置提案区块均由相同的验证者产生。而在一些可能的实现方式中，对应上述三种情况，前置提案区块与后置提案区块也可由不同的验证者产生。下面以具体的实施例对此进行解释。
20

基于相同的发明构思，共识模块具体用于连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

需要说明的是，证书代表了验证节点对待提交提案区块的确认和接受。每个验证节点都可以为一个或多个待提交提案区块生成证书，这些证书累积起来，形成了一个提案区块被接受程度的量化指标。为了确保区块链的稳定性和安全性，只有当待提交提案区块的证书数量达到或超过预设值时，该提案区块才会被认定为得到了足够的支持，从而被提交到目标区块链中。
25

还需要说明的是，证书数量的预设置可以根据区块链网络的具体情况进行调整。例如，在区块链网络规模较小或验证节点数量较少的情况下，可以适当降低证书数量的预设置；而在区块链网络规模较大或验证节点数量较多的情况下，则可以适当提高证书数量的预设置。
30

在本申请实施例中，对于同一个前置提案区块，可以有多个后置提案区块作为它的证书，且多个证书可以由不同的验证节点在同一个共识轮次中产生，也可以由相同的验证节点在不同轮次中产生。
35

基于相同的发明构思，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

需要说明的是，将后置提案区块作为前置提案区块的证书需要满足一定的条件，即后置提案区块所引用的提案区块的数量需要满足 $2f+1$ 的条件，该条件基于拜占庭容错算法（Byzantine Fault Tolerant），拜占庭容错算法是面向拜占庭问题的容错算法，解决的是在网络通信可靠但节点可能故障和作恶情况下如何达成共识的问题。其核心理论是系统中的总节点数大于等于 $3f+1$ ， f 是允许出现故障的节点数，换句话说，对于某个提案区块，只有法定数量（不小于 $2f+1$ ）个验证节点验证通过该区块，才能保证该提案区块的正确性。故本申请实施例要求对于前置提案区块来说，引用其生成的提案区块数量不得小于 $2f+1$ 。

此外，对于引用关系的检查，可以采用深度优先遍历的方式，以第 $n+x2$ 次共识轮次的某个后置提案区块作为起始点，优先遍历与其存在引用关系的提案区块，以通过引用关系确定某个前置提案区块是否得到足够多的验证节点的支持。

图 15 为本申请实施例提供的一种终端设备的结构示意图。该终端设备 1000 包括：至少一个处理器 1001（图 15 中仅示出一个处理器）、存储器 1002 以及存储在所述存储器 1002 中并可在所述至少一个处理器 1001 上运行的计算机程序 1003，所述处理器 1001 执行所述计算机程序 1003 时实现上述 6 方法实施例中的步骤。

所述终端设备 1000 可以是桌上型计算机、笔记本、掌上电脑及云端服务器等计算设备。该终端设备可包括，但不限于，处理器 1001、存储器 1002。本领域技术人员可以理解，图 15 仅仅是终端设备 1000 的举例，并不构成对终端设备 1000 的限定，可以包括比图示更多或更少的部件，或者组合某些部件，或者不同的部件，例如还可以包括输入输出设备、网络接入设备等。

所称处理器 1001 可以是中央处理单元（Central Processing Unit，CPU），该处理器 1001 还可以是其他通用处理器、数字信号处理器（Digital Signal Processor，DSP）、专用集成电路（Application Specific Integrated Circuit，ASIC）、现成可编程门阵列（Field-Programmable Gate Array，FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

所述存储器 1002 在一些实施例中可以是所述终端设备 1000 的内部存储单元，例如终端设备 1000 的硬盘或内存。所述存储器 1002 在另一些实施例中也可以是所述终端设备 1000 的外部存储设备，例如所述终端设备 1000 上配备的插接式硬盘，智能存储卡（Smart Media Card，SMC），安全数字（Secure Digital，SD）卡，闪存卡（Flash Card）等。进一步地，所述存储器 1002 还可以既包括所述终端设备 1000 的内部存储单元也包括外部存储设备。所述存储器 1002 用于存储操作系统、应用程序、引导装载程序（BootLoader）、数据以及其他程序等，例如所述计算机程序的程序代码等。所述存储器 1002 还可以用于暂时地存储已经输出或者将要输出的数据。

所属领域的技术人员可以清楚地了解到，为了描述的方便和简洁，仅以上述各功能单元、模块的划分进行举例说明，实际应用中，可以根据需要而将上述功能分配由不同的功能单元、模块完成，即将所述装置的内部结构划分成不同的功能单元或模块，以完成以上描述的全部或者部分功能。实施例中的各功能单元、模块可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中，上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。另外，各功能单元、模块的具体名称也只是为了便于相互区分，并不用于限制本申请的保护范围。上述

系统中单元、模块的具体工作过程，可以参考前述方法实施例中的对应过程，在此不再赘述。

本申请实施例还提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被处理器执行时可实现上述各个方法实施例中的步骤。

5 本申请实施例提供了一种计算机程序产品，当计算机程序产品在移动终端上运行时，使得移动终端执行时实现可实现上述各个方法实施例中的步骤。

所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用，可以存储在一个计算机可读存储介质中。基于这样的理解，本申请实现上述实施例方法中的全部或部分流程，可以通过计算机程序来指令相关的硬件来完成，所述的计算机程序可存储于计算机可读存储介质中，该计算机程序在被处理器执行时，可实现上述各个方法实施例的步骤。其中，所述计算机程序包括计算机程序代码，所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质至少可以包括：能够将计算机程序代码携带到拍照装置/终端设备的任何实体或装置、记录介质、计算机存储器、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、电载波信号、电信信号以及软件分发介质。例如 U 盘、移动硬盘、磁碟或者光盘等。在某些司法管辖区，根据立法和专利实践，计算机可读介质不可以是电载波信号和电信信号。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中未详述或记载的部分，可以参见其它实施例的相关描述。

20 本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本申请的范围。

25 在本申请所提供的实施例中，应该理解到，所揭露的装置/网络设备和/或方法，可以通过其它的方式实现。例如，以上所描述的装置/网络设备实施例仅仅是示意性的，例如，所述模块或单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通讯连接可以是通过一些接口，装置或单元的间接耦合或通讯连接，可以是电性，机械或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

35 以上所述实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围，均应包含在本申请的保护范围之内。

权利要求书

1. 一种区块链系统的共识方法，其特征在于，所述区块链系统包括多个验证节点，所述共识方法由一验证节点执行，所述共识方法包括：

5 在第 $n+x_2$ 次共识轮次中，引用多个提案区块，并结合客户端的当前交易，生成一后置提案区块；其中，多个提案区块是在第 n 次共识轮次中，由多个验证节点引用一前置提案区块所对应产生的，所述前置提案区块在第 $n-x_1$ 次共识轮次中产生；

将所述后置提案区块作为所述前置提案区块的证书，以共识所述前置提案区块。

2. 如权利要求 1 所述的共识方法，其特征在于， x_1 和 x_2 相同或不同，且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

10 3. 如权利要求 1 所述的共识方法，其特征在于，多个验证节点引用一前置提案区块对应产生提案区块，包括：

本验证节点连同其他验证节点，引用一前置提案区块对应产生提案区块。

4. 如权利要求 1 所述的共识方法，其特征在于，引用任意一个提案区块的步骤包括：引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

5. 如权利要求 1 所述的共识方法，其特征在于，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

6. 如权利要求 1 所述的共识方法，其特征在于，将所述后置提案区块作为所述前置提案区块的证书，包括：

20 连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

7. 如权利要求 1 所述的共识方法，其特征在于，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

8. 一种区块链系统的共识装置，其特征在于，包括：

30 后置提案区块生成模块，用于在第 $n+x_2$ 次共识轮次中，引用多个提案区块，并结合客户端的当前交易，生成一后置提案区块；其中，多个提案区块是在第 n 次共识轮次中，由多个验证节点引用一前置提案区块所对应产生的，所述前置提案区块在第 $n-x_1$ 次共识轮次中产生；

共识模块，用于将所述后置提案区块作为所述前置提案区块的证书，以共识所述前置提案区块。

9. 如权利要求 8 所述的共识装置，其特征在于， x_1 和 x_2 相同或不同，且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

35 10. 如权利要求 8 所述的共识装置，其特征在于，引用任意一个提案区块的步骤包括：引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

11. 如权利要求 8 所述的共识装置，其特征在于，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

40 12. 如权利要求 8 所述的共识装置，其特征在于，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

45 13. 一种终端设备，包括存储器、处理器以及存储在所述存储器中并可在所述处理器上运行的计算机程序，其特征在于，所述终端设备为区块链系统包括的多个验证节点中的任一验证节点；所述处理器执行所述计算机程序时实现如下步骤：

在第 $n+x_2$ 次共识轮次中，引用多个提案区块，并结合客户端的当前交易，生成一后置提案区块；其中，多个提案区块是在第 n 次共识轮次中，由多个验证节点引用一前置提

案区块所对应产生的，所述前置提案区块在第 $n-x_1$ 次共识轮次中产生；
将所述后置提案区块作为所述前置提案区块的证书，以共识所述前置提案区块。

14. 如权利要求 13 所述的终端设备，其特征在于， x_1 和 x_2 相同或不同，且 x_1 和 x_2 各自等于 1 或者各自不等于 1。

5 15. 如权利要求 13 所述的终端设备，其特征在于，多个验证节点引用一前置提案区块对应产生提案区块，包括：
本验证节点连同其他验证节点，引用一前置提案区块对应产生提案区块。

16. 如权利要求 13 所述的终端设备，其特征在于，引用任意一个提案区块的步骤包括：
引用所述任意一个提案区块的哈希索引，其中所述任意一个提案区块为所述提案区块和所述前置提案区块中的一个。

10 17. 如权利要求 13 所述的终端设备，其特征在于，所述前置提案区块由本验证节点产生，或者所述前置提案区块由其他验证节点产生。

18. 如权利要求 13 所述的终端设备，其特征在于，将所述后置提案区块作为所述前置提案区块的证书，包括：

15 连同其他至少一个验证节点生成的后置提案区块，各自独立地将所述后置提案区块作为所述前置提案区块的证书，其中，若证书数量大于一预设阈值，则验证通过所述前置提案区块。

19. 如权利要求 13 所述的终端设备，其特征在于，所述提案区块的数量不小于 $2f+1$ ，其中 f 为在第 n 次共识轮次中，不诚实的验证节点的数量，且 $3f+1$ 为第 n 次共识轮次中验证节点的总数量。

20 20. 一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现如权利要求 1 至 7 任一项所述共识方法的步骤。

25

30

35

40