(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 111159288 B
(45)授权公告日 2023.04.28

(21)申请号 201911294794.6

(22)申请日 2019.12.16

(65)同一申请的已公布的文献号
申请公布号 CN 111159288 A

(43)申请公布日 2020.05.15

(73)专利权人 郑杰骞
地址 558000 贵州省黔南布依族苗族自治州都匀市开发区虹桥派出所旁剑水村楼202号

(72)发明人 郑杰骞

(74)专利代理机构 北京安信方达知识产权代理有限公司 11262
专利代理师 解婷婷 栗若木

(51)Int.Cl.
G06F 16/27(2019.01)
G06F 16/22(2019.01)
G06F 21/64(2013.01)

(56)对比文件
CN 107180350 A,2017.09.19
CN 108647964 A,2018.10.12
CN 108681943 A,2018.10.19
CN 109242500 A,2019.01.18
CN 109255056 A,2019.01.22
CN 109726202 A,2019.05.07
CN 110177109 A,2019.08.27
US 10460120 B1,2019.10.29
US 2003093695 A1,2003.05.15
US 2016098730 A1,2016.04.07
US 2018232526 A1,2018.08.16
US 2018359089 A1,2018.12.13
US 2019058593 A1,2019.02.21
US 2019058595 A1,2019.02.21
US 2019103958 A1,2019.04.04
US 2019236726 A1,2019.08.01
US 2019379646 A1,2019.12.12
WO 2019170617 A1,2019.09.12

审查员 李玥

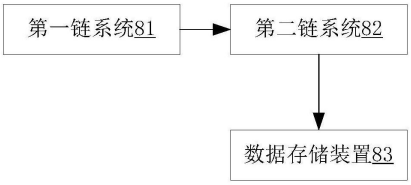
权利要求书7页 说明书32页 附图5页

(54)发明名称

链式结构数据存储、验证、实现方法、系统、装置及介质

(57)摘要

本公开提供了一种链式结构数据存储方法及装置、链式结构验证方法及装置、链式结构系统及实现方法、存储介质及计算机装置。所述链式结构实现方法包括:第一链系统对账本数据签名后,将签名后的账本数据写入第一链的区块数据;第二链系统在验证第一链的区块数据后,将所述第一链的一个或多个连续的区块数据写入第二链的区块数据;数据存储装置将所述第二链中链的部分或全部连接信息作为第一键,验证所述第一键关联的数据无错误后,将所述数据存储为输入数据或输出数据,同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。采用本公开提供的连接存储可以将庞大的链式结构数据分散存储在多个装置上,便于存储和验证。



CN 111159288 B

1.一种链式结构数据存储方法,其特征在于,所述方法包括:

将所述链式结构中链的部分或全部连接信息作为第一键,所述第一键关联的所述链式结构中的数据作为值,将所述数据存储为输入数据或输出数据,同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

2.根据权利要求1所述的数据存储方法,其特征在于:所述将所述数据存储为输入数据或输出数据,包括:

所述第一键作为所述链式结构中输出信息的部分或全部时,将所述第一键关联的数据作为输入数据存储;所述第一键作为所述链式结构中输入信息的部分或全部时,将所述第一健关联的数据作为输出数据存储。

3.一种链式结构验证方法,其特征在于,所述方法包括:

将所述链式结构中链的部分或全部连接信息作为第一键,所述第一键关联的所述链式结构中的数据作为值,验证所述第一键关联的数据是否有错误,验证所述第一键关联的数据无错误后,将所述数据存储为输入数据或输出数据。

4.根据权利要求3所述的链式结构验证方法,其特征在于,所述将所述数据存储为输入数据或输出数据,包括:

所述第一键作为所述链式结构中的输出信息的部分或全部时,将所述第一键关联的数据作为输入数据存储;所述第一键作为所述链式结构中的输入信息的部分或全部时,将所述第一键关联的数据作为输出数据存储;本装置上存储的输入数据和输出数据根据相同的连接信息相关联。

5.根据权利要求3所述的链式结构验证方法,其特征在于,所述验证所述第一键关联的数据是否有错误,包括以下验证中的一种或多种:

验证所述数据是否为所述链式结构上的数据;

所述第一键作为所述链式结构中的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

所述第一键作为所述链式结构中的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

所述第一键作为所述链式结构中的输入信息时,且所述第一键关联的数据包含交易数额时,验证所述交易数额是否正确;

所述第一键作为所述链式结构中的输入信息时,且所述第一键关联的数据包含签名信息时,验证所述签名信息是否正确。

6.根据权利要求4所述的链式结构验证方法,其特征在于,所述方法还包括:

根据数据验证装置的请求,向所述数据验证装置返回本装置存储的与所述链的连接信息关联的数据,包括以下数据中的一种或多种:输入数据、输出数据、默克尔树认证路径、附加验证数据。

7.一种链式结构验证方法,其特征在于,所述方法包括:

将所述链式结构中链的部分或全部连接信息作为第二键,从数据存储装置获取所述数据存储装置存储的与所述第二键关联的数据,根据所述获取的数据验证所述链式结构中的数据是否有错误。

8.根据权利要求7所述的链式结构验证方法,其特征在于,所述从数据存储装置获取的

与所述第二键关联的数据包括以下一种或多种:与所述第二键关联的数据、默克尔树认证路径、附加验证数据,其中,所述第二键作为所述链式结构中输出信息的部分或全部,或者所述第二键作为所述链式结构中输入信息的部分或全部。

9.根据权利要求7所述的链式结构验证方法,其特征在于,所述根据所述获取的数据验证所述链式结构中的数据是否有错误,包括:

同步所述链式结构中的区块头数据,结合所述区块头数据与从所述数据存储装置获取的与所述第二键关联的数据,进行以下验证中的一种或多种:

验证所述从数据存储装置获取的与所述第二键关联的数据是否为所述链式结构上的数据;

所述第二键作为所述获取的数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

所述第二键作为所述获取的数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含交易数额时,验证所述交易数额是否正确;

所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含签名信息时,验证所述签名信息是否正确。

10.一种链式结构实现方法,其特征在于,所述链式结构系统包括第一链系统、第二链系统和数据存储装置,所述方法包括:

所述第一链系统对账本数据签名后,将签名后的账本数据写入第一链的区块数据;

所述第二链系统在验证第一链的区块数据后,将所述第一链的一个或多个连续的区块数据写入第二链的区块数据;

所述数据存储装置将所述第二链中链的部分或全部连接信息作为第一键,所述第一键关联的数据作为值,验证所述第一键关联的数据无错误后,将所述数据存储为输入数据或输出数据,同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

11.根据权利要求10所述的链式结构实现方法,其特征在于,所述方法还包括:所述第一链系统将所述第一链中链的部分或全部连接信息作为第三键,所述第三键关联的数据作为值,将所述第三键关联的值分配到与所述值具有相同第三键的共识组;所述共识组对所述第三键关联的数据进行验证,其中:

所述第三键关联的数据包括交易数据,所述验证包括以下一种或多种:

所述第三键作为所述交易数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

所述第三键作为所述交易数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

所述第三键作为所述交易数据的输入信息时,验证所述交易数据的交易数额是否正确;

所述第三键作为所述交易数据的输入信息时,验证所述交易数据的签名信息是否正确。

12.根据权利要求11所述的链式结构实现方法,其特征在于,所述共识组对所述第三键

关联的数据进行验证后，所述方法还包括：

所述共识组将所述第三键关联的数据存储为输入数据或输出数据，其中，当所述第三键作为所述第一链中的输出信息的部分或全部时，将所述第三键关联的数据作为输入数据存储，当所述第三键作为所述第一链中的输入信息的部分或全部时，将所述第三键关联的数据作为输出数据存储；同一共识组上存储的输入数据和输出数据根据相同的连接信息相关联。

13.根据权利要求10所述的链式结构实现方法，其特征在于，所述数据存储装置将所述数据存储为输入数据或输出数据，包括：

所述第一键作为所述第二链中的输出信息的部分或全部时，将所述第一键关联的数据作为输入数据存储；所述第一键作为所述第二链中的输入信息的部分或全部时，将所述第一键关联的数据作为输出数据存储；本数据存储装置上存储的输入数据和输出数据根据相同的连接信息相关联。

14.根据权利要求10所述的链式结构实现方法，其特征在于，所述数据存储装置验证所述第一键关联的数据是否有错误，包括以下验证中的一种或多种：

验证所述第二链的完整性；

验证所述数据是否为所述第二链上的数据；

所述第一键作为所述第二链中账本数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

所述第一键作为所述第二链中账本数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

所述第一键作为所述第二链中账本数据的输入信息时，且所述第一键关联的数据包含交易数额时，验证所述交易数额是否正确；

所述第一键作为所述第二链中账本数据的输入信息时，且所述第一键关联的数据包含签名信息时，验证所述签名信息是否正确。

15.根据权利要求10或14所述的链式结构实现方法，其特征在于，

所述方法还包括：所述第一链系统按顺序为账本数据编排序号，所述账本数据对应的区块数据的区块头中包含累计账本数据量；所述第二链系统在第二链的区块数据的区块头中包含所述区块数据中的累计账本数据量。

16.根据权利要求15所述的链式结构实现方法，其特征在于，

所述数据存储装置验证所述链式结构的完整性，包括：所述数据存储装置同步所述第二链的区块头，用所述区块头中的累计账本数据量生成编号地址，根据所述编号地址对所述第二链的完整性进行验证。

17.根据权利要求15所述的链式结构实现方法，其特征在于，所述方法还包括：

所述数据存储装置根据第一链区块头数据验证第二链区块头数据中的累计账本数据量是否正确。

18.根据权利要求10所述的链式结构实现方法，其特征在于，所述方法还包括：

所述第一链系统在第一链的区块头中包含当前共识公钥集合映射的值；

所述数据存储装置根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明，验证第二链区块头中的共识公钥是否有效。

19.根据权利要求10所述的链式结构实现方法,其特征在于,

所述第二链系统将所述第一链的一个或多个连续的区块数据写入第二链的区块数据,包括:所述第二链系统将所述第一链的一个或多个连续的区块数据按照交易数据序号顺序和控制数据序号顺序重新组合分别生成默克尔树根哈希值,并在生成的第二链的区块头中包含所述默克尔树根哈希值。

20.根据权利要求10所述的链式结构实现方法,其特征在于,所述系统还包括:数据验证装置;所述方法还包括:所述数据验证装置将所述链式结构中链的部分或全部连接信息作为第二键,从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据,根据所述获取的数据验证所述第二链中的数据是否有错误。

21.根据权利要求20所述的链式结构实现方法,其特征在于,所述数据验证装置将所述链式结构中链的部分或全部连接信息作为第二键,从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据,包括:

所述数据验证装置同步所述第二链的区块头,用所述区块头中的累计账本数据量生成编号地址,根据所述编号地址获取链的部分或全部连接信息作为第二键,从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据。

22.根据权利要求20所述的链式结构实现方法,其特征在于,

所述数据验证装置从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据包括以下数据中的一种或多种:与所述第二键关联的数据、默克尔树认证路径、附加验证数据,其中,所述第二键作为所述链式结构中输出信息的部分或全部,或者所述第二键作为所述链式结构中输入信息的部分或全部;

所述数据验证装置根据所述获取的数据验证所述第二链中的数据是否有错误,包括:

所述数据验证装置同步所述第二链中的区块头,结合所述区块头与从所述数据存储装置获取的与所述第二键关联的数据,进行以下验证中的一种或多种:

验证所述从数据存储装置获取的与所述第二键关联的数据是否为所述第二链上的数据;

所述第二键作为所述获取的数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

所述第二键作为所述获取的数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含交易数额时,验证所述交易数额是否正确;

所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含签名信息时,验证所述签名信息是否正确。

23.根据权利要求20所述的链式结构实现方法,其特征在于,

所述第一链系统在第一链的区块头中包含当前共识公钥集合映射的值;

所述数据验证装置根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明,验证第二链区块头中的共识公钥是否有效。

24.根据权利要求10所述的链式结构实现方法,其特征在于,所述第一链为私有链或联盟链,所述方法还包括:第一链系统向用户颁发以下密钥的一个或多个:管理地址主密钥、

交易地址主密钥、机密交易主密钥和对称加密主密钥，其中：

所述管理地址主密钥用于与当前第一生成参数生成所述用户的下一个管理地址，所述用户的所有管理地址形成逻辑链；

所述交易地址主密钥用于与当前第二生成参数生成所述用户的下一个接收交易地址，所述用户的所有接收交易地址形成逻辑链；

所述机密交易主密钥用于与当前第三生成参数生成当前加解密密文交易数额的工作密钥；

所述对称加密主密钥用于与当前第四生成参数生成所述用户的下一个加解密管理数据的对称加密工作密钥。

25.根据权利要求10或24所述的链式结构实现方法，其特征在于，所述方法还包括：

所述第一链系统使用为用户颁发的管理地址主密钥与所述用户的上一个管理数据中的生成参数生成所述用户的当前管理地址，并在所述当前管理数据中写入用于生成所述用户的下一个管理地址的生成参数。

26.一种链式结构数据存储装置，其特征在于，所述数据存储装置包括第一存储模块和第二存储模块，其中：

所述第一存储模块，用于将所述链式结构中链的部分或全部连接信息作为第一键存储；

所述第二存储模块，用于将所述第一键关联的所述链式结构中的数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

27.根据权利要求26所述的链式结构数据存储装置，其特征在于，所述数据存储装置还包括第一验证模块，所述第一验证模块用于验证所述第一键关联的所述链式结构中的数据是否有错误。

28.根据权利要求27所述的链式结构数据存储装置，其特征在于，所述第一验证模块验证所述第一键关联的数据是否有错误，包括：

所述第一验证模块执行以下验证操作中的一种或多种：

验证所述数据是否为所述链式结构上的数据；

所述第一键作为所述链式结构中的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

所述第一键作为所述链式结构中的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

所述第一键作为所述链式结构中的输入信息时，且所述第一键关联的数据包含交易数额时，验证所述交易数额是否正确；

所述第一键作为所述链式结构中的输入信息时，且所述第一键关联的数据包含签名信息时，验证所述签名信息是否正确。

29.一种链式结构数据验证装置，其特征在于，所述数据验证装置包括键值查找模块、第二验证模块，其中：

所述键值查找模块，用于查找所述链式结构中链的部分或全部连接信息作为第二键；

所述第二验证模块，用于从数据存储装置获取所述数据存储装置存储的与所述第二键关联的数据，根据所述获取的数据验证所述链式结构中的数据是否有错误。

30.根据权利要求29所述的链式结构数据验证装置，其特征在于，所述第二验证模块根据所述获取的数据验证所述链式结构中的数据是否有错误，包括：

所述数据验证装置同步所述链式结构中的区块头数据，所述第二验证模块结合所述区块头数据与从所述数据存储装置获取的与所述第二键关联的数据，进行以下验证中的一种或多种：

验证所述从数据存储装置获取的与所述第二键关联的数据是否为所述链式结构上的数据；

所述第二键作为所述获取的数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

所述第二键作为所述获取的数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

所述第二键作为所述获取的数据的输入信息时，且所述获取的数据包含交易数额时，验证所述交易数额是否正确；

所述第二键作为所述获取的数据的输入信息时，且所述获取的数据包含签名信息时，验证所述签名信息是否正确。

31.一种链式结构系统，其特征在于，包括：第一链系统、第二链系统和数据存储装置，其中：

所述第一链系统，用于对账本数据签名后，将签名后的账本数据写入第一链的区块数据；

所述第二链系统，用于在验证第一链的区块数据后，将所述第一链的一个或多个连续的区块数据写入第二链的区块数据；

所述数据存储装置，用于将所述第二链中链的部分或全部连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据是否有错误，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

32.根据权利要求31所述的链式结构系统，其特征在于，所述第一链系统还包括共识组；

所述第一链系统还用于将所述第一链中链的部分或全部连接信息作为第三键，所述第三键关联的数据作为值，将所述第三键关联的值分配到与所述值具有相同第三键的共识组，所述第三键关联的数据包括交易数据；

所述共识组用于对所述第三键关联的数据进行验证，包括以下一种或多种验证：

所述第三键作为所述交易数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

所述第三键作为所述交易数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

所述第三键作为所述交易数据的输入信息时，验证所述交易数据的交易数额是否正确；

所述第三键作为所述交易数据的输入信息时，验证所述交易数据的签名信息是否正确。

33.根据权利要求32所述的链式结构系统,其特征在于,所述共识组还用于对所述第三键关联的数据进行验证后,将所述第三键关联的数据存储为输入数据或输出数据,其中,当所述第三键作为所述第一链中的输出信息的部分或全部时,将所述第三键关联的数据作为输入数据存储,当所述第三键作为所述第一链中的输入信息的部分或全部时,将所述第三键关联的数据作为输出数据存储;同一共识组上存储的输入数据和输出数据根据相同的连接信息相关联。

34.根据权利要求31所述的链式结构系统,其特征在于,所述第一链系统还用于按顺序为账本数据编排序号,所述账本数据对应的区块数据的区块头中包含累计账本数据量;所述第二链系统还用于在第二链的区块数据的区块头中包含所述区块数据中的累计账本数据量。

35.根据权利要求31所述的链式结构系统,其特征在于,所述数据存储装置验证所述第一键关联的数据是否有错误,包括以下验证中的一种或多种:

验证所述第二链的完整性;

验证所述数据是否为所述第二链上的数据;

所述第一键作为所述第二链中账本数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

所述第一键作为所述第二链中账本数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

所述第一键作为所述第二链中账本数据的输入信息时,且所述第一键关联的数据包含交易数额时,验证所述交易数额是否正确;

所述第一键作为所述第二链中账本数据的输入信息时,且所述第一键关联的数据包含签名信息时,验证所述签名信息是否正确。

36.根据权利要求31所述的链式结构系统,其特征在于,所述链式结构系统还包括:数据验证装置;所述数据验证装置用于将所述链式结构中链的部分或全部连接信息作为第二键,从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据,根据所述获取的数据验证所述第二链中的数据是否有错误。

37.根据权利要求31或36所述的链式结构系统,其特征在于,所述第一链系统还用于在第一链的区块头中包含当前共识公钥集合映射的值,以使所述数据存储装置或数据验证装置根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明,验证第二链区块头中的共识公钥是否有效。

38.根据权利要求31所述的链式结构系统,其特征在于,所述第一链系统还用于使用为用户颁发的管理地址主密钥与所述用户的上一个管理数据中的生成参数生成所述用户的当前管理地址,并在所述当前管理数据中写入用于生成所述用户的下一个管理地址的生成参数。

39.一种计算机可读存储介质,存储有计算机可执行指令,所述计算机可执行指令用于执行权利要求1-2或权利要求3-6或权利要求7-9或权利要求10-25中任一项所述的方法。

40.一种计算机装置,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述程序时实现如权利要求1-2或权利要求3-6或权利要求7-9或权利要求10-25中任一项所述方法的步骤。

链式结构数据存储、验证、实现方法、系统、装置及介质

技术领域

[0001] 本申请涉及但不限于计算机数据处理技术领域，尤指一种链式结构数据存储方法及装置、链式结构验证方法及装置、链式结构系统及实现方法、存储介质及计算机装置。

背景技术

[0002] 目前的去中心化系统例如公有链系统，因为有着公开、透明、可追溯、不可篡改等特点，所以能降低交易参与方之间的信任成本，可作为信任的基础实现价值传递。

[0003] 然而，目前的去中心化系统存在以下问题：节点需要存储全部数据才能验证整个账本数据。

发明内容

[0004] 以下是对本文详细描述的主题的概述。本概述并非是为了限制权利要求的保护范围。

[0005] 本文提供一种数据存储方法、链式结构验证方法、链式结构实现方法及系统，节点只需存储链式结构中的部分数据。

[0006] 第一方面，本公开提供一种数据存储方法，用于存储链式结构中的数据，所述方法包括：

[0007] 将所述链式结构中链的部分或全部连接信息作为第一键，所述第一键关联的所述链式结构中的数据作为值，将所述数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0008] 第二方面，本公开还提供一种链式结构验证方法，所述方法包括：

[0009] 将所述链式结构中链的部分或全部连接信息作为第一键，所述第一键关联的所述链式结构中的数据作为值，验证所述第一键关联的数据是否有错误，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据。

[0010] 第三方面，本公开还提供一种链式结构验证方法，所述方法包括：

[0011] 将所述链式结构中链的部分或全部连接信息作为第二键，从数据存储装置获取所述数据存储装置存储的与所述第二键关联的数据，根据所述获取的数据验证所述链式结构中的数据是否有错误。

[0012] 第四方面，本公开还提供一种链式结构实现方法，所述链式结构系统包括第一链系统、第二链系统和数据存储装置，所述方法包括：

[0013] 所述第一链系统对账本数据签名后，将签名后的账本数据写入第一链的区块数据；

[0014] 所述第二链系统在验证第一链的区块数据后，将所述第一链的一个或多个连续的区块数据写入第二链的区块数据；

[0015] 所述数据存储装置将所述第二链中链的部分或全部连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据无错误后，将所述数据存储为输入数

据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0016] 第五方面，本公开还提供了一种链式结构数据存储装置，所述数据存储装置包括第一存储模块和第二存储模块，其中：

[0017] 所述第一存储模块，用于将所述链式结构中链的部分或全部连接信息作为第一键存储；

[0018] 所述第二存储模块，用于将所述第一键关联的所述链式结构中的数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0019] 第六方面，本公开还提供了一种链式结构数据验证装置，所述数据验证装置包括键值查找模块、第二验证模块，其中：

[0020] 所述键值查找模块，用于查找所述链式结构中链的部分或全部连接信息作为第二键；

[0021] 所述第二验证模块，用于从数据存储装置获取所述数据存储装置存储的与所述第二键关联的数据，根据所述获取的数据验证所述链式结构中的数据是否有错误。

[0022] 第七方面，本公开还提供一种链式结构系统，包括：第一链系统、第二链系统和数据存储装置，其中：

[0023] 所述第一链系统，用于对账本数据签名后，将签名后的账本数据写入第一链的区块数据；

[0024] 所述第二链系统，用于在验证第一链的区块数据后，将所述第一链的一个或多个连续的区块数据写入第二链的区块数据；

[0025] 所述数据存储装置，用于将所述第二链中链的部分或全部连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据是否有错误，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0026] 第八方面，本公开还提供一种计算机可读存储介质，存储有计算机可执行指令，所述计算机可执行指令用于实现上述任意一种方法。

[0027] 第九方面，本公开还提供一种计算机装置，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，所述处理器执行所述程序时实现上述任意一种方法中的步骤。

[0028] 采用本公开提供的连接存储可以将庞大的链式结构数据分散存储在多个装置上，以减轻数据存储压力。该种存储方式还便于进行验证。

[0029] 本发明的其它特征和优点将在随后的说明书中阐述，并且，部分地从说明书中变得显而易见，或者通过实施本发明而了解。本发明的目的和其他优点可通过在说明书、权利要求书以及附图中所特别指出的结构来实现和获得。

[0030] 在阅读并理解了附图和详细描述后，可以明白其他方面。

附图说明

[0031] 附图用来提供对本发明技术方案的进一步理解，并且构成说明书的一部分，与本

申请的实施例一起用于解释本发明的技术方案，并不构成对本发明技术方案的限制。

[0032] 图1为一种示例性数据存储方法流程图；

[0033] 图2为一种示例性数据存储装置结构示意图；

[0034] 图3为一种示例性数据验证方法流程图；

[0035] 图4为一种示例性具有验证功能的数据存储装置结构示意图；

[0036] 图5为另一种示例性数据验证方法流程图；

[0037] 图6为一种示例性数据验证装置结构示意图；

[0038] 图7为一种示例性链式结构实现方法流程图；

[0039] 图8为一种示例性链式结构系统示意图；

[0040] 图9为一种示例性连接存储图，连接信息为Bd1；

[0041] 图10为一种示例性三层二链系统架构图；

[0042] 图11为一种示例性计算机设备的结构示意图。

具体实施方式

[0043] 本申请描述了多个实施例，但是该描述是示例性的，而不是限制性的，并且对于本领域的普通技术人员来说显而易见的是，在本申请所描述的实施例包含的范围内可以有更多的实施例和实现方案。尽管在附图中示出了许多可能的特征组合，并在具体实施方式中进行了讨论，但是所公开的特征的许多其它组合方式也是可能的。除非特意加以限制的情况以外，任何实施例的任何特征或元件可以与任何其它实施例中的任何其他特征或元件结合使用，或可以替代任何其它实施例中的任何其他特征或元件。

[0044] 本申请包括并设想了与本领域普通技术人员已知的特征和元件的组合。本申请已经公开的实施例、特征和元件也可以与任何常规特征或元件组合，以形成由权利要求限定的独特的发明方案。任何实施例的任何特征或元件也可以与来自其它发明方案的特征或元件组合，以形成另一个由权利要求限定的独特的发明方案。因此，应当理解，在本申请中示出和/或讨论的任何特征可以单独地或以任何适当的组合来实现。因此，除了根据所附权利要求及其等同替换所做的限制以外，实施例不受其它限制。此外，可以在所附权利要求的保护范围内进行各种修改和改变。

[0045] 此外，在描述具有代表性的实施例时，说明书可能已经将方法和/或过程呈现为特定的步骤序列。然而，在该方法或过程不依赖于本文所述步骤的特定顺序的程度上，该方法或过程不应限于所述的特定顺序的步骤。如本领域普通技术人员将理解的，其它的步骤顺序也是可能的。因此，说明书中阐述的步骤的特定顺序不应被解释为对权利要求的限制。此外，针对该方法和/或过程的权利要求不应限于按照所写顺序执行它们的步骤，本领域技术人员可以容易地理解，这些顺序可以变化，并且仍然保持在本申请实施例的精神和范围内。

[0046] 在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行。并且，虽然在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤。

[0047] 本文提出一种私有链与公有链的结合方案，解决私有链数据的可信问题，并且解决部分公平性问题，以及每个用户端只需要存储部分账本数据即可等价验证全部账本数据。

[0048] 下面先对本文涉及的概念进行说明。

[0049] token,指区块链上的通证。

[0050] CA(Certificate Authority),认证授权。

[0051] eID(electronic Identity),指公民网络电子身份标识,是可信的实名认证方式

[0052] SPV(Simplified Payment Verification):简单支付验证,通过默克尔树认证路径验证某交易是否包含在链的区块中,并且经过多少个确认。

[0053] POA(Proof of Authority):权威证明,一种基于身份和声誉的共识算法。

[0054] Gossip网络协议:也叫 Epidemic Protocol (流行病协议),是非结构化对等网络协议。

[0055] DHT(Distributed Hash Table):分布式哈希表,是一种分布式存储方法。在不需要服务器的情况下,每个客户端负责一小范围的路由,并负责存储一小部分数据,从而实现整个DHT网络的寻址和存储。

[0056] Kademlia网络协议:是一种分布式哈希表(DHT)网络协议,是结构化对等网络协议,是IPFS等系统采用的网络协议。

[0057] 一致性哈希算法:DHT的一种实现。能够在动态变化的环境中满足平衡性、单调性、分散性和负载。

[0058] UTXO(Unspent Transaction Output),未花费的交易输出,引用一个或多个未花费交易输出变为已花费,并且创建一个或多个新的未花费交易输出,并且不断循环向后延伸。根据其连接的方式,形成DAG(Directed Acyclic Graph有向无环图)的链式结构。本文称为UTXO链。

[0059] 公有链,指任何人都可以读取、发送交易和参与共识的区块链,属于完全去中心化的系统。本文实施例中采用的是一种除任何人都可以发送交易外,其余都与公有链系统相同的类公有链系统,满足交易中心化,账本数据去中心化。在本文所述的类公有链系统中,任何人都可以读取、验证交易和参与共识,能够实现可追溯、不可篡改。

[0060] 私有链,指写入权限在一个组织手里的数据链,属于中心化的系统。

[0061] 联盟链,指写入权限在多个组织手里的数据链,属于部分去中心化的系统。

[0062] 本文一示例性实施例提供了一种用于存储链式结构中数据的数据存储方法,如图1所示,包括步骤11-12。

[0063] 步骤11,将所述链式结构中链的部分或全部连接信息作为第一键;

[0064] 步骤12,所述第一键关联的所述链式结构中的数据作为值,将所述数据存储为输入数据或输出数据,同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0065] 一个或一组数据中包含前一个或一组数据中的信息,该信息即为链的连接信息,该连接信息使得前述数据在逻辑上形成一链式结构。

[0066] 用于进行数据存储的装置可采用DHT技术实现。所述装置上存储有作为第一键(也可称为键值或者关键值)的所述链式结构中的连接信息。作为第一键的连接信息可以是该链式结构连接信息的部分也可以是全部。所述第一键例如可以是地址或者地址的哈希值。

[0067] 对于所述链式结构中的任一个或一组数据,每个或每组数据包含输入信息和输出信息,其中当前数据的输入信息为数据M的输出信息,该数据M在逻辑上为当前数据的前一

数据(上一数据),该当前数据的输入信息或数据M的输出信息属于链的连接信息,当前数据的输出信息为数据N的输入信息,该数据N在逻辑上为当前数据的后一数据(下一数据),该当前数据的输出信息或数据N的输入信息也属于链的连接信息。该链式结构例如为UTXO链,每个交易数据包括引用的未花费输出作为输入信息和新的未花费输出作为输出信息。

[0068] 数据存储装置当接收到与第一键值相匹配的数据时,根据所述数据在链式结构中所表示的意义,将所述数据存储为输入数据或输出数据。同一数据存储装置上存储的输入数据和输出数据根据相同的键(即链的连接信息)相关联,此种存储方式可称为连接存储或者数据结对存储。同一数据存储装置上可能存储一组或多组输入数据和输出数据,对于任意一组输入数据和输出数据,该输入数据和输出数据根据相同的连接信息相关联。同一装置上存储的输入数据可以有一个或多个,存储的输出数据也可以有零个(即无输出数据)、或一个或多个。所述链式结构可以是任意具有链式结构特征的链。所述数据存储装置可以是用户端节点也可以是共识组,也可以是其他需要进行连接存储的装置。待存储的数据可以是交易数据,也可以是控制数据,例如可以将控制数据中的token发行数据存储为输入数据,将控制数据中的token回收数据存储为输出数据。其中对应的键分别为token发行预设地址、token回收预设地址。

[0069] 采用此种连接存储可以将庞大的链式结构数据分散存储在多个装置上,以减轻数据存储压力。该种存储方式还便于进行验证。

[0070] 在一示例性实施例中,所述第一键作为所述链式结构中输出信息的部分或全部时,将所述第一键关联的数据作为输入数据存储;所述第一键作为所述链式结构中输入信息的部分或全部时,将所述第一健关联的数据作为输出数据存储。

[0071] 例如,如果第一键为所述链式结构中的输出地址时,则将该第一键关联的数据存储为输入数据,如果第一键为所述链式结构中的输入地址时,则将该第一键关联的数据存储为输出数据。

[0072] 本实施例中的数据存储装置可以和其他系统结合使用,例如与一个去中心化系统结合,或者与一个中心化系统结合。

[0073] 上述数据存储装置例如可以如图2所示,包括第一存储模块21和第二存储模块22,其中:

[0074] 第一存储模块21,用于将所述链式结构中链的部分或全部连接信息作为第一键存储;

[0075] 第二存储模块22,用于将所述第一键关联的所述链式结构中的数据存储为输入数据或输出数据,同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0076] 例如,当所述第一键作为所述链式结构中输出信息的部分或全部时,所述第二存储模块22将所述第一键关联的数据作为输入数据存储,当所述第一键作为所述链式结构中输入信息的部分或全部时,所述第二存储模块22将所述第一健关联的数据作为输出数据存储。

[0077] 采用此种存储方式,使得整个链式结构可以被多个数据存储装置分段存储,便于检索和验证。

[0078] 本文一示例性实施例还提供了一种链式结构验证方法,由数据存储装置实现验

证，如图3所示，所述方法包括步骤31-33。

[0079] 步骤31，将所述链式结构中链的部分或全部连接信息作为第一键；

[0080] 步骤32，所述第一键关联的所述链式结构中的数据作为值，验证所述第一键关联的数据是否有错误；

[0081] 步骤33，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据。

[0082] 如前述实施例所述，所述链式结构可以是任意具有链式结构特征的链。所述数据存储装置可以是用户端节点也可以是共识组，也可以是其他需要进行连接存储的装置。待存储的数据可以是交易数据，也可以是控制数据。作为第一键的连接信息可以是该链式结构连接信息的部分也可以是全部。在本实施例中，由进行数据存储的装置(例如用户端节点)进行链式结构验证。该数据存储装置上存储有作为第一键的该链式结构中的连接信息，因此该数据存储装置在接收与本装置对应的第一键匹配的数据时，对所述数据进行验证，在验证无错误后，再进行连接存储。

[0083] 以数据为账本数据为例，数据存储装置验证所述账本数据无错误后，判断所述账本数据的输出信息与所述第一键匹配时，将所述账本数据存储为输入数据，判断所述账本数据的输入信息与所述第一键匹配时，将所述账本数据存储为输出数据，同一装置上存储的输入数据和输出数据根据相同的连接信息相关联。同一装置上存储的输入数据可以有一个或多个，存储的输出数据也可以有零个(即无输出数据)、或一个或多个。

[0084] 本实施例在连接存储时进行验证，可以将对整个链式结构的验证转化为局部验证，由多个数据存储装置分别进行，相当于等价验证全部数据的正确性，采用此种等价验证方式，将验证工作分散到多个装置上，以减轻数据验证的工作量压力。

[0085] 在一示例性实施例中，将所述数据存储为输入数据或输出数据，可以采取以下方式：所述第一键作为所述链式结构中的输出信息的部分或全部时，将所述第一键关联的数据作为输入数据存储；所述第一键作为所述链式结构中的输入信息的部分或全部时，将所述第一键关联的数据作为输出数据存储；本装置上存储的输入数据和输出数据根据相同的连接信息相关联。

[0086] 在一示例性实施例中，上述验证所述第一键关联的数据是否有错误，包括以下验证中的一种或多种：

[0087] 验证11，验证所述数据是否为所述链式结构上的数据；

[0088] 验证12，所述第一键作为所述链式结构中的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

[0089] 验证13，所述第一键作为所述链式结构中的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

[0090] 验证14，所述第一键作为所述链式结构中的输入信息时，且所述第一键关联的数据包含交易数额时，验证所述交易数额是否正确。

[0091] 验证15，所述第一键作为所述链式结构中的输入信息时，且所述第一键关联的数据包含签名信息时，验证所述签名信息是否正确。所述签名信息是产生交易数据的用户端的解锁签名。所述验证签名信息是否正确即验证签名信息是否有效。

[0092] 上述验证序号仅为说明方便，不代表验证顺序。对于上述验证11，如果数据是链式

结构上的数据，则验证为无错误；对于验证12，如果存在与所述输入信息具有相同连接信息的输出信息，则验证为无错误；对于验证13，所述与输入信息具有相同连接信息的输出信息未被使用过，则验证为无错误；对于验证14，交易数额正确，则验证为无错误；对于验证15，验证签名信息有效，则验证为无错误。

[0093] 一种示例性的验证过程：所述数据存储装置同步所述链式结构中的区块头数据，所述装置查找本装置是否存储有与所述第一键相关联的输入数据(还可以从所述链式结构中查找)，如果没有，则判断为连接错误，如果有，判断所述输入数据的关联连接是否被引用，如果已被引用过，则判断为连接错误；判断所述账本数据的交易数额是否正确，如果正确，则将所述账本数据存储为输出数据，并将输入数据的关联连接标识为已被引用，也即相关联的输出信息将标记为已使用，如果不正确，判断为数据错误，如果正确，如果数据中包含签名信息时，验证所述签名信息是否正确，如果不正确判断为数据错误。所述签名信息例如是产生交易数据的用户端的解锁签名。当验证成功后，与输入信息相关联的输出信息将标记为已使用。

[0094] 在一示例性实施例中，可采用以下方式验证所述数据是否为所述链式结构上的数据：

[0095] 同步所述链式结构的区块头数据，根据所述区块头数据中的默克尔树根哈希值和所述数据的认证路径验证所述数据是否为链式结构上的数据。

[0096] 在一示例性实施例中，所述方法还包括：根据数据验证装置的请求，向所述数据验证装置返回本装置存储的与所述链的连接信息关联的数据，包括但不限于以下数据中的一种或多种：输入数据、输出数据、默克尔树认证路径、附加验证数据。

[0097] 其中，输入数据可能有一个或多个。输出数据可能没有(即零个)或者有一个或者多个，无输出数据表示输入数据的连接未被使用或未花费。正常情况下，有输入数据才有输出数据，但也不排除出错的情况，只有输出数据，没有输入数据。默克尔树认证路径与输入数据或输出数据对应存在。附加验证数据可能有，也可能没有。附加验证数据用于验证交易数额是否正确，例如UTXO中需要累加全部输入数额和全部输出数额，所以可能需要其它的交易数据才能完成交易数额的验证。附加验证数据也有对应的默克尔树认证路径。

[0098] 为了加强数据的安全性，设置其他装置的验证机制，即由非数据存储装置作为数据验证装置验证所述链式结构中的数据有无错误，被选为数据验证装置的节点从存储装置处获取相关数据，以完成验证。

[0099] 本实施例中的数据存储装置可以和其他系统结合使用，例如与一个去中心化系统结合，或者与一个中心化系统结合。

[0100] 上述具有验证功能的数据存储装置例如可以如图4所示，包括第三存储模块41、第一验证模块42和第四存储模块43，其中：

[0101] 第三存储模块41，用于存储所述链式结构中链的部分或全部连接信息作为第一键；

[0102] 第一验证模块42，用于验证所述第一键关联的所述链式结构中的数据是否有错误；

[0103] 所述第四存储模块43，用于在所述验证模块42验证无错误后，将所述第一键关联的所述链式结构中的数据存储为输入数据或输出数据。

[0104] 例如,所述第三存储模块41的存储方式可与图2中的第一存储模块21相同,所述第四存储模块43的存储方式可与图2中的第二存储模块22相同,该具有验证功能的数据存储装置可在上述图2所示数据存储装置基础上增加第一验证模块42。当所述第一键作为所述链式结构中输出信息的部分或全部时,所述第四存储模块43将所述第一键关联的数据作为输入数据存储,当所述第一键作为所述链式结构中输入信息的部分或全部时,所述第四存储模块43将所述第一健关联的数据作为输出数据存储。

[0105] 所述第一验证模块42可以进行上述验证11-15中的任意一种或多种,此处不再赘述。

[0106] 在一示例性实施例中,所述数据存储装置还可以包括发送模块,其用于根据其他数据验证装置的请求,向所述数据验证装置返回本装置存储的与所述链的连接信息关联的数据,包括以下数据中的一种或多种:输入数据、输出数据、默克尔树认证路径、附加验证数据。

[0107] 采用此种存储和验证方式,使得整个链式结构可以被多个数据存储装置分段存储和等价验证。

[0108] 本文一示例性实施例还提供一种链式结构验证方法,由数据验证装置实现,如图5所示,所述方法包括步骤51-52。

[0109] 步骤51,将所述链式结构中链的部分或全部连接信息作为第二键;

[0110] 步骤52,从数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据(即键对应的值),根据所述获取的数据验证所述链式结构中的数据是否有错误。

[0111] 为了加强数据的安全性,设置其他装置验证机制,即由非数据存储装置验证所述链式结构中的数据有无错误,作为数据验证装置的节点从数据存储装置处获取相关数据,完成验证。所述第二键例如可以是地址或者地址的哈希值。有关链式结构、链式结构中连接信息、输入信息和输出信息的说明参见前述实施例中的描述,此处不再赘述。

[0112] 在一示例性实施例中,所述从数据存储装置获取的与所述第二键关联的数据包括但不限于以下一种或多种:所述数据存储装置存储的:与所述第二键关联的数据、默克尔树认证路径、附加验证数据,其中,所述第二键作为所述链式结构中输出信息的部分或全部,或者所述第二键作为所述链式结构中输入信息的部分或全部。当所述第二键作为所述链式结构中输出信息的部分或全部时,所述第二键关联的数据即为所述数据存储装置存储的输入数据;所述第二键作为所述链式结构中输入信息的部分或全部时,所述第二键关联的数据即为所述数据存储装置存储的输出数据。

[0113] 所述数据验证装置可同步链式结构的区块头,所述区块头中包含累计账本数据量,所述数据验证装置用所述区块头中的累计账本数据量生成编号地址(用累计控制数据量生成控制数据编号地址,用累计交易数据量生成交易数据编号地址),根据所述编号地址获取链的部分或全部连接信息作为第二键,从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据。控制数据编号地址作为键获取的是该控制数据编号地址对应的控制数据,还可以包括该控制数据的默克尔树认证路径;交易数据编号地址作为键获取的是该交易数据编号地址对应的交易数据,还可以包括该交易数据的默克尔树认证路径。例如,可将控制数据的token发行数据的预设地址(简称token发行地址)、或token回收数据的预设地址(简称token回收地址)作为键,用该键从数据存储装置获取其上存储的与

键关联的数据;可将交易数据的交易地址作为键,用该键从数据存储装置获取其上存储的与键关联的数据。

[0114] 在一示例性实施例中,上述根据所述获取的数据验证所述链式结构中的数据是否有错误,包括:

[0115] 同步所述链式结构中的区块头数据,结合所述区块头数据与从所述数据存储装置获取的与所述第二键关联的数据,验证所述链式结构中的数据是否有错误。例如包括进行以下验证中的一种或多种:

[0116] 验证21,验证所述从数据存储装置获取的与所述第二键关联的值是否为所述链式结构上的数据;

[0117] 验证22,所述第二键作为所述获取的数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

[0118] 验证23,所述第二键作为所述获取的数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

[0119] 验证24,所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含交易数额时,验证所述交易数额是否正确;

[0120] 验证25,所述第二键作为所述获取的数据的输入信息时,且所述获取的数据包含签名信息时,验证所述签名信息是否正确。

[0121] 上述验证序号仅为说明方便,不代表验证顺序。对于上述验证21,如果所述值是链式结构上的数据,则验证为无错误;对于验证22,如果存在与所述输入信息具有相同连接信息的输出信息,则验证为无错误;对于验证23,所述与输入信息具有相同连接信息的输出信息未被使用过,则验证为无错误;对于验证24,交易数额正确,则验证为无错误;对于验证25,验证签名信息有效,则验证为无错误。

[0122] 一种示例性的验证过程:使用默克尔树认证路径验证数据存储装置上存储的输入数据,或者输入数据和输出数据只要有一个数据不为链上的数据,则判断为数据错误;判断如果只有输出数据,则判断为连接错误;判断所述输入数据的关联连接如果被引用过,则判断为连接错误;判断所述输出数据的交易数额如果不正确,则判断为数据错误;判断所述数据中的签名信息(例如产生数据的用户端的解锁签名)如果不正确,则判断为数据错误。

[0123] 通过非存储装置验证可以提高系统的安全性,避免仅由固定存储装置验证带来的风险。

[0124] 上述数据验证装置例如可以如图6所示,包括键值查找模块41、第二验证模块62,其中:

[0125] 键值查找模块61,用于查找所述链式结构中链的部分或全部连接信息作为第二键;

[0126] 第二验证模块62,用于从数据存储装置获取所述数据存储装置存储的与所述第二键关联的数据(即键对应的值),根据所述获取的数据验证所述链式结构中的数据是否有错误。

[0127] 所述第二验证模块62可以进行上述验证21-25中的任意一种或多种,此处不再赘述。

[0128] 采用此种验证机制,通过增加数据验证装置,提高了系统的安全性。

[0129] 本文一示例性实施例还提供了一种链式结构实现方法，所述链式结构系统包括第一链系统、第二链系统和数据存储装置，如图7所示，所述方法包括步骤71-73。

[0130] 步骤71，所述第一链系统对账本数据签名后，将签名后的账本数据写入第一链的区块数据；

[0131] 步骤72，所述第二链系统在验证第一链的区块数据后，将所述第一链的一个或多个连续的区块数据写入第二链的区块数据；

[0132] 步骤73，所述数据存储装置将所述第二链中链的部分或全部连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0133] 所述账本数据包括交易数据和/或控制数据。

[0134] 所述第一链为私有链或者联盟链，为由管理端控制的链，第二链与公有链类似，区别在于并非任意数据可以上链。当第一链产生新的区块数据后，会同步到第二链的链生成节点，区块数据验证通过后由第二链的链生成节点根据第二链的共识算法产生新的区块数据。第二链中顺序记录第一链提交的区块数据，第二链上每个区块数据包含的第一链的区块数据的数量可以不固定，也即第二链上每个区块可以包含1至n个第一链的区块数据，数量由第二链的共识算法决定。所以对外的第二链系统上的每个区块数据是由第一链系统的一个或多个区块数据顺序组合而成，因此第一链系统和第二链系统在逻辑状态上是相同的。本实施例使用先生成链后验证的方式，也即先生成第二链，再由第三层的数据存储装置进行验证的方式。实际的账本数据是由第一层即第一链系统上链的，第二层即第二链系统可验证默克尔树和管理端的签名，并不修改账本数据，由第三层的数据存储装置验证账本数据是否有错误。实现交易中心化，账本数据去中心化。第三层的数据存储装置采用连接存储的方式存储并验证数据，每个数据存储装置只需要存储和验证少量数据，即可实现整个链的等价验证，减少每个装置的负担。且第二层的链生成节点可以只验证第一链系统的管理端签名和默克尔树，降低了第二链系统的计算量。

[0135] 在一示例性实施例中，所述方法还包括：所述第一链系统将所述第一链中链的部分或全部连接信息作为第三键，所述第三键关联的数据作为值，将所述第三键关联的值分配到与所述值具有相同第三键的共识组；例如第一链系统可在对交易数据签名之前进行上述分配。所述共识组对所述第三键关联的数据进行验证，其中：

[0136] 所述第三键关联的数据包括交易数据，所述验证包括以下一种或多种：

[0137] 所述第三键作为所述交易数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

[0138] 所述第三键作为所述交易数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

[0139] 所述第三键作为所述交易数据的输入信息时，验证所述交易数据的交易数额是否正确；

[0140] 所述第三键作为所述交易数据的输入信息时，验证所述交易数据的签名信息(用户端的解锁签名)是否正确。

[0141] 上述验证的顺序不限。

[0142] 一种示例性的验证过程:共识组查找与所述交易数据相连接的输入数据,即第三键作为所述交易数据的输入信息时,查找是否存在与所述输入信息具有相同连接信息的输出信息,如果未查找到,判断为连接错误,如果查找到,判断所述输入数据的关联连接是否被引用,即第三键作为所述交易数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过,如果已被引用过或被使用过,则判断为连接错误,如果未被引用过,判断所述交易数据的交易数额是否正确,如果不正确,判断为数据错误,如果正确,判断数据中包含的签名信息(例如产生交易数据的用户端的解锁签名)是否正确,如果不正确,判断为数据错误,如果正确,则将所述交易数据存储为输出数据(即共识组也可以采用上述连接存储方式存储数据),并将所述输入数据的关联连接标识为已被引用,也即相关联的输出信息将标记为已使用。

[0143] 在一示例性实施例中,所述共识组对所述第三键关联的数据进行验证后,进行连接存储时,根据所述数据在第一链中所表示的意义,将所述第三键关联的数据存储为输入数据或输出数据,其中,当所述第三键作为所述第一链中的输出信息的部分或全部时,将所述第三键关联的数据作为输入数据存储,当所述第三键作为所述第一链中的输入信息的部分或全部时,将所述第三键关联的数据作为输出数据存储;同一共识组上存储的输入数据和输出数据根据相同的连接信息相关联。

[0144] 例如,所述共识组如果判断所述交易数据或控制数据的输出信息与所述第三键匹配,则将所述交易数据或控制数据存储为输入数据;如果判断所述交易数据或控制数据的输入信息与所述第三键匹配,将所述交易数据或控制数据存储为输出数据。

[0145] 第一链系统中包括多个共识组,同一个共识组中的节点对应的健是相同的,如果接收到的交易数据被验证,则连接存储所述交易数据与所述键,如果接收到的控制数据被验证,则连接存储所述键值与控制数据。如果所述数据为所述链式结构中的输出信息,则将所述数据作为输入数据存储;如果所述数据为所述链式结构中的输入信息,则将所述数据作为输出数据存储。同一共识组上存储的输入数据和输出数据具有相同的键,以形成数据连接存储。

[0146] 在一示例性实施例中,所述数据存储装置对数据的存储与前述实施例中的处理相同:当所述第一键作为所述第二链中的输出信息的部分或全部时,将所述第一键关联的数据作为输入数据存储;当所述第一键作为所述第二链中的输入信息部分或全部时,将所述第一键关联的数据作为输出数据存储;本数据存储装置上存储的输入数据和输出数据根据相同的连接信息相关联。

[0147] 在一示例性实施例中,所述数据存储装置验证所述第一键关联的数据是否有错误包括以下验证中的一种或多种:

[0148] 验证31,验证所述第二链的完整性;

[0149] 验证32,验证所述数据是否为所述第二链上的数据;

[0150] 验证33,所述第一键作为所述第二链中账本数据的输入信息时,验证是否存在与所述输入信息具有相同连接信息的输出信息;

[0151] 验证34,所述第一键作为所述第二链中账本数据的输入信息时,验证与所述输入信息具有相同连接信息的输出信息是否被使用过;

[0152] 验证35,所述第一键作为所述第二链中账本数据的输入信息时,且所述第一键关

联的数据包含交易数额时，验证所述交易数额是否正确。

[0153] 验证36，所述第一键作为所述第二链中账本数据的输入信息时，且所述第一键关联的数据包含签名信息时，验证所述签名信息是否正确。

[0154] 本实施例中的验证32-36与前述实施例中的验证11-15类似，区别在于本实施例中的验证对象为第二链。

[0155] 在一示例性实施例中，所述方法还包括：所述第一链系统按顺序为账本数据编排序号，所述账本数据对应的区块数据的区块头中包含累计账本数据量。所述第二链系统在第二链的区块数据的区块头中包含所述区块数据中的累计账本数据量。

[0156] 例如，在将账本数据写入第一链的区块数据时，按第一顺序为交易数据编排序号，按第二顺序为控制数据编排序号，在区块数据的区块头中写入最后一个交易数据的序号和/或最后一个控制数据的序号，使得所述区块头数据中包含累计交易数据量和累计控制数据量。例如，每一共识组接收到前一共识组发送的累计交易数据量后，在所述累计交易数据量的基础上对本共识组待上链的交易数据编排序号，重新计算新的累计交易数据量并发送给下一个共识组；共识组对所述待上链的交易数据编排序号后，所述共识组请求对编排序号后的交易数据进行签名，将被签名的交易数据写入第一链，所述交易数据所在区块的区块头中包含累计了所述交易数据的累计交易数据量。第一链的累计账本数据量包括第一累计交易数据量和/或第一累计控制数据量，第二链中的累计账本数据量包括第二累计交易数据量和/或第二累计控制数据量。所述第二链与第一链异步同态，由于第一链的区块头中包含累计交易数据量和累计控制数据量，因此所述第二链的区块头中也包含相应的累计交易数据量和累计控制数据量。通过在区块头数据中携带累计账本数据量以使数据存储装置能够根据累计账本数据量对第二链进行完整性验证。

[0157] 所述数据存储装置可采用以下方式验证所述链式结构的完整性：所述数据存储装置同步所述第二链的区块头，根据所述区块头中的累计账本数据量（累计交易数据量和累计控制数据量）对所述第二链的完整性进行验证。验证完整性时，通过编号地址为键来进行验证。例如，装置根据当前区块头数据中的累计交易数据量和前一区块头数据中的累计交易数据量计算得到当前区块的所有交易数据编号地址，根据当前区块头数据中的累计控制数据量和前一区块头数据中的累计交易数据量计算得到当前区块的所有控制数据编号地址，判断每个交易数据编号地址如果与本节点分布式哈希表的网络标识匹配，则查找与所述交易数据编号地址对应的交易数据，判断每个控制数据编号地址如果与本节点分布式哈希表的网络标识匹配，则查找与所述控制数据编号地址对应的控制数据，如果找到，则完整性验证通过，如果未找到，完整性验证失败。

[0158] 完整性验证是由存储编号地址的数据存储装置进行的验证，可在存储时验证。因为分布式哈希表的特点就是要把数据存储在对应键的节点上，以提供键的检索，所以本身就需要存储键(key)与值(value)。这里因为编号地址是序号，是已知的，所以键已知，也就知道键应该存储在哪些节点上。所以节点只需要同步区块头数据，就知道哪些编号地址的键应该由自己存储，如果节点判断自己需要存储该键的数据，但是没有存储该数据，则在第三层(用户端)或第二层(类公有链)检索该数据，然后进行完整性验证，这样通过节点自己验证，也就能够验证数据的完整性。因为通过区块头的累计数量，节点就能够知道有哪些编号地址。

[0159] 除了完整性验证外，还可以进行随机验证。随机验证可以是任意节点，不只是存储编号地址的节点，可以随机挑选某些编号地址进行验证，并不需要存储。随机验证时根据随机编号地址获取该编号地址对应的交易数据，将交易数据的输入地址作为键进行验证。本质上与完整性验证类似。

[0160] 在一示例性实施例中，所述数据存储装置根据第一链区块头数据验证第二链区块头数据，包括验证累计账本数据量是否正确。数据存储装置可验证第一链区块头的累计交易数量和累计控制数量与第二链区块头的累计交易数量和累计控制数量是否相等，如果相等则说明数量正确并且第一链区块头是第二链区块头产生时间片段内的最后一个区块头数据。

[0161] 在一示例性实施例中，通过在第二链区块头数据中携带默克尔树根哈希值以使数据存储装置验证数据是否为链上数据。所述第二链系统将所述第一链的一个或多个连续的区块数据写入第二链的区块数据，包括：所述第二链系统将所述第一链的一个或多个连续的区块数据按照交易数据序号顺序和控制数据序号顺序重新组合分别生成默克尔树根哈希值，并在生成的第二链的区块头中包含所述默克尔树根哈希值。

[0162] 所述数据存储装置验证所述数据是否为所述第二链上的数据，包括：所述数据存储装置同步所述第二链的区块头，根据所述区块头中的默克尔树根哈希值以及所述数据对应的认证路径验证所述数据是否为第二链上的数据。

[0163] 在一示例性实施例中，除了数据存储装置可实现数据验证外，为提高安全性，增加数据验证装置进行数据验证，即前述实施例中的数据验证装置。所述链式结构系统还包括：数据验证装置，所述方法还包括：所述数据验证装置将所述链式结构中链的部分或全部连接信息作为第二键，从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据，根据所述获取的数据验证所述第二链中的数据是否有错误。例如，所述数据验证装置可同步所述第二链的区块头，用所述区块头中的累计账本数据量生成编号地址(用累计控制数据量生成控制数据编号地址，用累计交易数据量生成交易数据编号地址)，根据所述编号地址获取链的部分或全部连接信息作为第二键，从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据。控制数据编号地址作为键获取的是该控制数据编号地址对应的控制数据，还可以包括该控制数据的默克尔树认证路径；交易数据编号地址作为键获取的是该交易数据编号地址对应的交易数据，还可以包括该交易数据的默克尔树认证路径。例如，可将控制数据的token发行数据的预设地址(简称token发行地址)、或token回收数据的预设地址(简称token回收地址)作为键，用该键从数据存储装置获取其上存储的与键关联的数据；可将交易数据的交易地址作为键，用该键从数据存储装置获取其上存储的与键关联的数据。

[0164] 例如，数据验证装置从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据，包括以下数据中的一种或多种：与所述第二键关联的数据、默克尔树认证路径、附加验证数据，其中，所述第二键作为所述链式结构中输出信息的部分或全部，或者所述第二键作为所述链式结构中输入信息的部分或全部；所述数据验证装置同步所述第二链中的区块头，结合所述区块头与从所述数据存储装置获取的与所述第二键关联的数据，进行以下验证中的一种或多种：

[0165] 验证所述从数据存储装置获取的与所述第二键关联的数据是否为所述第二链上

的数据；

[0166] 所述第二键作为所述获取的数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

[0167] 所述第二键作为所述获取的数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

[0168] 所述第二键作为所述获取的数据的输入信息时，且所述获取的数据包含交易数额时，验证所述交易数额是否正确；

[0169] 所述第二键作为所述获取的数据的输入信息时，且所述获取的数据包含签名信息时，验证所述签名信息是否正确。

[0170] 此处验证部分的说明参见前述实施例中描述，此处不再赘述。

[0171] 在一示例性实施例中，所述第一链为私有链或联盟链，所述方法还包括：第一链系统向用户颁发以下密钥的一个或多个：管理地址主密钥、交易地址主密钥、机密交易主密钥和对称加密主密钥，其中：

[0172] 所述管理地址主密钥用于与当前第一生成参数生成所述用户的下一个管理地址，所述用户的所有管理地址形成一逻辑链；

[0173] 所述交易地址主密钥用于与当前第二生成参数生成所述用户的下一个接收交易地址，所述用户的所有接收交易地址形成一逻辑链；

[0174] 所述机密交易主密钥用于与当前第三生成参数生成当前加解密密文交易数额的工作密钥；

[0175] 所述对称加密主密钥用于与当前第四生成参数生成所述用户的下一个加解密管理数据的对称加密工作密钥，该对称加密主密钥还可以用于生成其它数据的对称加密工作密钥。

[0176] 例如，所述第一链系统使用为用户颁发的管理地址主密钥与所述用户的上一个管理数据中的生成参数生成所述用户的当前的管理地址，并在所述当前的管理数据中写入用于生成所述用户的下一个管理地址的生成参数。所述第一链系统可以使用为用户颁发的对称加密主密钥与所述用户的上一个管理数据中的生成参数生成对称加密工作密钥加密所述用户的当前的管理数据。所述用户可以使用相同的密钥生成方式生成对称加密工作密钥解密所述用户的当前的管理数据。

[0177] 在一示例性实施例中，所述方法还包括：所述第一链系统在第一链的区块头中包含当前共识公钥集合映射的值；所述数据存储装置根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明，验证第二链区块头中的共识公钥是否有效。除数据存储装置可进行上述验证外，所述数据验证装置也可以根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明，验证第二链区块头中的共识公钥是否有效。

[0178] 本文一示例性实施例还提供了一种链式结构系统，如图8所示，包括：第一链系统81、第二链系统82和数据存储装置83，其中：

[0179] 所述第一链系统81，用于对账本数据签名后，将签名后的账本数据写入第一链的区块数据；

[0180] 所述第二链系统82，用于在验证第一链的区块数据后，将所述第一链的一个或多

个连续的区块数据写入第二链的区块数据；

[0181] 所述数据存储装置83，用于将所述第二链中链的部分或全部连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据是否有错误，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据，同一数据存储装置存储的输入数据和输出数据根据相同的连接信息相关联。

[0182] 在一示例性实施例中，所述第一链系统还包括共识组；

[0183] 所述第一链系统还用于将所述第一链中链的部分或全部连接信息作为第三键，所述第三键关联的数据作为值，将所述第三键关联的值分配到与所述值具有相同第三键的共识组，所述第三键关联的数据包括交易数据；

[0184] 所述共识组用于对所述第三键关联的数据进行验证，包括以下一种或多种验证：

[0185] 所述第三键作为所述交易数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

[0186] 所述第三键作为所述交易数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

[0187] 所述第三键作为所述交易数据的输入信息时，验证所述交易数据的交易数额是否正确；

[0188] 所述第三键作为所述交易数据的输入信息时，验证所述交易数据的签名信息是否正确。

[0189] 在一示例性实施例中，所述共识组还用于对所述第三键关联的数据进行验证后，将所述第三键关联的数据存储为输入数据或输出数据，其中，当所述第三键作为所述第一链中的输出信息的部分或全部时，将所述第三键关联的数据作为输入数据存储，当所述第三键作为所述第一链中的输入信息的部分或全部时，将所述第三键关联的数据作为输出数据存储；同一共识组上存储的输入数据和输出数据根据相同的连接信息相关联。

[0190] 在一示例性实施例中，所述第一链系统还用于按顺序为账本数据编排序号，所述账本数据对应的区块数据的区块头中包含累计账本数据量；所述第二链系统还用于在第二链的区块数据的区块头中包含所述区块数据中的累计账本数据量。

[0191] 在一示例性实施例中，所述数据存储装置83例如可以是如图2所示的数据存储装置，所述数据存储装置将所述数据存储为输入数据或输出数据，包括：所述第一键作为所述第二链中的输出信息的部分或全部时，将所述第一键关联的数据作为输入数据存储；所述第一键作为所述第二链中的输入信息的部分或全部时，将所述第一键关联的数据作为输出数据存储；本数据存储装置上存储的输入数据和输出数据根据相同的连接信息相关联。

[0192] 在一示例性实施例中，所述数据存储装置验证所述第一键关联的数据是否有错误，包括以下验证中的一种或多种：

[0193] 验证所述第二链的完整性；

[0194] 验证所述数据是否为所述第二链上的数据；

[0195] 所述第一键作为所述第二链中账本数据的输入信息时，验证是否存在与所述输入信息具有相同连接信息的输出信息；

[0196] 所述第一键作为所述第二链中账本数据的输入信息时，验证与所述输入信息具有相同连接信息的输出信息是否被使用过；

[0197] 所述第一键作为所述第二链中账本数据的输入信息时，且所述第一键关联的数据包含交易数额时，验证所述交易数额是否正确；

[0198] 所述第一键作为所述第二链中账本数据的输入信息时，且所述第一键关联的数据包含签名信息时，验证所述签名信息是否正确。

[0199] 在一示例性实施例中，所述数据存储装置采用以下方式验证所述链式结构的完整性：所述数据存储装置同步所述第二链的区块头，用所述区块头中的累计账本数据量生成编号地址，根据所述编号地址对所述第二链的完整性进行验证。

[0200] 在一示例性实施例中，所述数据存储装置还用于根据第一链区块头数据验证第二链区块头数据中的累计账本数据量是否正确。

[0201] 在一示例性实施例中，所述第一链为私有链或联盟链，所述第一链系统还用于向用户颁发以下密钥的一个或多个：管理地址主密钥、交易地址主密钥、机密交易主密钥和对称加密主密钥。

[0202] 在一示例性实施例中，所述系统还可包括数据验证装置，所述数据验证装置例如可以是如图6所示的数据验证装置。所述数据验证装置用于将所述链式结构中链的部分或全部连接信息作为第二键，从所述数据存储装置获取所述数据存储装置上存储的与所述第二键关联的数据，根据所述获取的数据验证所述第二链中的数据是否有错误。

[0203] 在一示例性实施例中，所述第一链系统还用于在第一链的区块头中包含当前共识公钥集合映射的值，以使所述数据存储装置或数据验证装置根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明，验证第二链区块头中的共识公钥是否有效。

[0204] 在一示例性实施例中，所述第一链系统还用于使用为用户颁发的管理地址主密钥与所述用户的上一个管理数据中的生成参数生成所述用户的当前管理地址，并在所述当前管理数据中写入用于生成所述用户的下一个管理地址的生成参数。

[0205] 所述链式结构系统中第一链系统、第二链系统、数据存储装置、数据验证装置的功能和效果参见方法中描述，此处不再赘述。

[0206] 下面对本文中提到的键(key，或称为键值、关键值)进行说明：

[0207] 第一键值为数据存储装置进行连接存储时使用的键值，所述第二键值为数据验证装置从数据存储装置上检索数据时使用的键值，所述第三键值为第一链系统中进行共识组查询，以及共识组进行连接存储时使用的键值。第一键值、第二键值和第三键值可以依据对应的链式结构的不同而有所差别，但均使用链式结构中的连接信息的部分或者全部。例如可以是以下类型中的一种或多种：交易数据的交易地址，控制数据的token发行数据的预设地址(简称token发行地址)、和token回收数据的预设地址(简称token回收地址)。键值关联的数据为对应的交易数据或控制数据，还可以包括数据对应的默克尔树认证路径(简称认证路径)。键值可以存储于管理端或客户端。以上述实施例中的链式结构系统为例，该链式结构系统包括第一链系统、第二链系统和多个用户端(或称客户端)节点，用户端节点可以包括作为数据存储装置的用户端节点，还可以包括作为数据验证装置的用户端节点。对于用户端节点，键值关联的数据除交易数据或控制数据外，还包含对应的默克尔树认证路径，该认证路径为该交易数据或控制数据在第二链中的默克尔树认证路径。

[0208] 本公开还提供一类键值(第四键值)，可用于数据存储装置验证账本数据的完整

性,还可以用于数据验证装置进行键值的查询。第四键值包括以下信息的一种或多种:交易数据的编号地址(根据交易数据编号生成)和控制数据的编号地址(根据控制数据编号生成)。可以存储于客户端。第四键值关联的数据为对应的交易数据或控制数据,还可以包括对应的默克尔树认证路径。在上述链式结构系统的示例中,第四键值关联的认证路径为该交易数据或控制数据在第二链中的默克尔树认证路径。

[0209] 此外,本公开还提供一类键值(第五键值),可用于进行账户数据链的查询或检索,用户的账户数据链包括由所述用户的管理数据组成的第一账户数据链和由所述用户的接收交易数据组成的第二账户数据链。用于存储管理数据的管理地址为隐性链式结构,由此构成第一账户数据链。用于存储交易数据的交易地址也为隐性链式结构,由此构成第二账户数据链。第五键值包括用于查询管理数据的管理地址或用于查询交易数据的交易地址。

[0210] 本文中作为键值的地址可以是地址或者是地址的哈希值。

[0211] 下面介绍默克尔树和区块链结构。默克尔树是一种哈希二叉树,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。默克尔树的叶子节点保存着数据集合的单元数据的哈希值,而节点之间通过哈希运算得到父节点的哈希值,通过一层层往上层计算,最终会形成根节点的哈希值。其中叶子节点可以根据根哈希值,以及对应的认证路径,验证该叶子节点是否属于数据集合中的元素。

[0212] 可以看出,如果默克尔树的叶子节点数量(也即数据集合元素的数量)已知,叶子节点所在的序号(即位置)也已知,则该叶子节点所对应的认证路径的高度和方向是固定且已知的,其中方向是指路径的左右方向。所以也即表示该元素是有序的,不能用不同的认证路径以及高度和方向来替代,也就能加强数据的安全性。

[0213] 默克尔树的叶子节点数量最多是树深度的2^n,即使包含大量的数据,也能通过固定的路径快速验证某个叶子节点。比如树深度为30,则最大能包含1073741824个叶子节点。如果平均10分钟产生一个数据块,则每秒可包含1789569笔交易。但验证某个叶子节点的数据,只需要30个hash值即可完成验证,如果每个hash值为32字节,则需要960字节。如果每个客户端每个区块需要保存10笔数据,每个客户端一年大约需要保存525600笔数据,每个区块都按树深度为30计算,则需要保存的认证路径大小为481MB,如果每笔数据大小为1KB,则需要保存的总数据量大小为994MB。但实际并不需要保存这么多数据,并且一定年限后客户端就可以清除之前的数据,使需要保存的数据量一直保持在可控的范围,即使移动设备也可以接受。

[0214] 区块链是由连续分片的时间片段内产生的区块头数据和区块体数据组成。其中后一个区块头包含了前一个区块头的哈希值,从而形成反向连接的链式结构。并且区块头还包含了对应区块体数据的默克尔树根哈希值,从而能唯一映射该区块体数据。而区块体数据则包含实际的账本数据。并且因为区块头唯一映射了区块体数据,也即区块头的一致性可以映射区块体数据的一致性。所以只需要同步区块头数据,就可以根据账本数据和对应的认证路径,验证该账本数据是否在链的区块中,并且经过多少个确认,也即SPV简单支付验证。

[0215] 下面介绍底层数据结构。账本数据分为交易数据集和控制数据集两部分,交易数据集包括实际的交易数据;控制数据集则包括但不限于以下数据的一种或多种类型:用户的管理数据、token的发行数据、token的回收数据、系统发布的公告数据。交易数据主要由

客户端产生,包含客户端的解锁签名,并由管理端验证上链,并且上链时,被管理端赋予顺序编排(例如递增)的唯一序号,该序号连同交易数据一起经过管理端的背书签名。控制数据由链式结构系统(例如第一链系统)的管理端产生,产生时赋予顺序编排(例如递增)的唯一序号并经过管理端签名。交易数据集和控制数据集分别生成默克尔树的根哈希值记录在区块头中。区块头中也会包含对应区块体数据中的最后一个交易数据编号和最后一个控制数据编号,并且该编号是顺序递增的,也等同于区块头中包含当前所有的累计交易数据量和累计控制数据量。

[0216] 第三层用户端(包括数据存储装置)使用的是结构化对等网络,比如可以使用Kademlia网络协议。每个用户端(节点)只需要同步区块头数据,而区块体中的账本数据,则会根据分布式哈希表(DHT)由整个网络上的节点分散存储,每个节点存储部分账本数据和对应的认证路径。由于每个交易数据和控制数据都具有一个递增的唯一序号,并且区块头中会包含最后一个序号,相当于包含当前的累计交易数据量和累计控制数据量,所以能够快速查找到每个交易数据和控制数据所在的区块,再结合认证路径,即可使用默克尔树验证该数据。并且因为该区块的默克尔树的叶子节点数量和该数据所对应的序号(即位置)都是已知的,所以认证路径的高度和方向是固定且已知的,不能用不同的认证路径以及高度和方向来替代,加强了数据的安全性。

[0217] 用户则是通过账户数据链查找自己的账户数据。而第三方用户和监管者也可以在被授权用户主密钥后通过账户数据链获取该用户的账户数据。账户数据链是指通过用户主密钥与当前生成参数得到一中间值K,再由K经过其它运算得到下一个数据的地址,从而形成正向连接的逻辑链式结构。账户数据链使用户能够在隐私情况下根据用户主密钥进行检索。一个用户有两条账户数据链,其中一条账户数据链即第一账户数据链由用户的管理数据组成,该用户通过管理地址主密钥进行检索。所述第一链系统使用为用户颁发的管理地址主密钥与所述用户的上一个管理数据中的生成参数生成所述用户的当前管理地址,该当前管理地址被包含在当前管理数据中,并在所述当前管理数据中写入用于生成所述用户的下一个管理地址的生成参数,以使所述用户的所有管理地址形成逻辑链。控制数据中用户管理数据的管理地址可以作为该第一账户数据链的查询键值,该键值关联的数据为管理地址对应的用户管理数据。

[0218] 初始管理地址生成参数可以是一个预设值,比如用户ID。该管理地址是一次性的,并且都是唯一的,从而可以达到保护用户身份隐私的目的。所述第一链系统使用为用户颁发的对称加密主密钥与所述用户的上一个管理数据中的生成参数生成对称加密工作密钥加密所述用户的当前的管理数据。所述用户可以使用相同的密钥生成方式生成对称加密工作密钥解密所述用户的当前的管理数据。

[0219] 另一条账户数据链即第二账户数据链由用户的接收交易数据组成,该用户通过交易地址主密钥进行检索。所述接收交易数据上链后,同一交易接收端的所有接收交易数据将会形成逻辑链式结构。该逻辑链隐含在生成的账本数据中。

[0220] 交易数据由用户端提交给第一链系统的管理端,管理端对交易数据进行验证。所述交易数据包括交易接收端的交易地址和本次交易时生成的地址参数,交易地址利用所述交易接收端上次接收交易时生成的地址参数生成,所述本次交易时生成的地址参数用于生成所述交易接收端下次接收交易的交易地址。管理端对交易数据的验证主要包括有效性的

验证，例如包括验证用户状态的有效性，解锁脚本的有效性，交易数额的有效性，以及交易地址是否是有效的地址等。交易数据验证通过后，管理端对交易数据进行背书签名，背书签名后的交易数据将被写入第一链的区块数据中。所述交易数据上链后，同一交易接收端的所有接收交易数据将会形成逻辑链式结构。该逻辑链隐含在生成的账本数据中。

[0221] 用户在注册时管理端为其生成初始地址参数(或称为nonce值)，并生成起始接收交易地址，该用户作为交易接收端发生交易时，该用户的所有接收交易数据会形成一个逻辑链即第二账户数据链。当同一用户被管理端颁发了用于生成交易地址的新密钥后，管理端会重新为该用户生成一个初始地址参数，并由重新生成的初始地址参数生成新的起始接收交易地址，所述重新生成的初始地址参数与颁发的新密钥是配套的，或者说是相关联的。此后，当该用户作为交易接收端发生交易时，该用户的所有接收交易数据会形成一个新的第二账户数据链。可见同一交易接收端的接收交易数据可有一个或多个逻辑链式结构。每次新生成的初始地址参数会存储在用户的管理数据中，用户可以根据管理数据中的初始地址参数自行查找第二账户数据链。

[0222] 如果有多个针对同一交易接收端的并发交易，即存在多个包含相同交易地址的交易数据，则包含相同交易地址的交易数据在第二账户数据链中互为兄弟节点。

[0223] 通过交易数据中包含的用上一次接收交易中的地址参数生成的交易地址和用于生成下一次交易地址的地址参数，使得同一用户端的接收交易数据形成或有兄弟节点的逻辑链条，从而能够快速检索用户端的所有接收交易数据，且由于发送交易数据是引用接收交易数据，从而能够实现快速获取到所有交易数据。该交易地址是一次性的，对不同用户是唯一的，从而可以达到保护用户身份隐私的目的。

[0224] 由于交易地址可作为第一键值，所以第二账户数据链的交易地址连接信息是隐藏在连接存储的键值里，用户可以通过将交易地址作为第二键值检索获取所述用户的第二账户数据链。作为第二键值的交易地址可以为所述用户的当前接收交易地址，由所述用户的交易地址主密钥与所述用户的上一个接收交易数据里的生成参数，以及所述用户的签名公钥生成。

[0225] 用户端的密钥分为两类，一类是用户端的签名密钥，另一类是由管理端颁发的用户主密钥。签名密钥可由用户端本地生成并管理，也可由可信第三方进行托管。如果需要实名认证，则该签名密钥的公钥需经过可信CA机构颁发身份证书或经过eID认证。管理端颁发的用户主密钥包括管理地址主密钥、交易地址主密钥、机密交易主密钥和对称加密主密钥。其中管理地址主密钥用于生成由用户的管理数据组成的第一账户数据链的地址；交易地址主密钥用于生成由用户的接收交易数据组成的第二账户数据链的地址；机密交易主密钥用于生成机密交易中的工作密钥，可用于保护盲化因子，使用户可以解密密文交易数额；对称加密主密钥用于生成对称加密工作密钥，可用于保护用户的管理数据和用户的其它数据。

[0226] 用户主密钥与当前生成参数通过第一单向不可逆函数运算得到一中间值K，其中，管理地址主密钥与第一生成参数生成中间值K1，交易地址主密钥与第二生成参数生成中间值K2，机密交易主密钥与第三生成参数生成中间值K3，对称加密主密钥与第四生成参数生成中间值K4。每个交易数据里的生成参数可以不同，初始生成参数可以是用户ID。可由K1通过第二单向不可逆函数运算得到所述用户的下一个管理数据的地址；由K2与用户的签名公钥运算得到一新的公钥，比如可以使用椭圆曲线上的标量乘法运算，再用新的公钥通过第

三单向不可逆函数运算得到所述用户的下一个接收交易的地址，并且该新公钥所对应的私钥可由K2与用户的签名私钥运算得到，比如有限域的乘法运算；由K3通过第四单向不可逆函数运算得到下一个管理数据的对称加密工作密钥；由K4通过第五单向不可逆函数运算得到当前机密交易中的工作密钥，可用于保护盲化因子。用户主密钥与当前生成参数还可以通过多个单向不可逆函数运算得到多个不同的中间值，然后中间值一对一与多个签名公钥运算得到多个新的公钥，再由这些新的公钥生成下一个接收交易的多重签名地址。

[0227] 上述及本文中的单向不可逆函数都可以是散列函数或散列函数的组合，下文使用散列函数阐述。

[0228] 通过用户主密钥与不同的生成参数得到不同的地址或密钥，还能加强数据的安全性，比如上述散列函数都使用具有能抵抗量子计算破解的散列函数。即使量子计算破解了椭圆曲线上的私钥，但由于该私钥的公钥是通过用户签名公钥与中间值K运算得到的，也即还需要破解中间值K才能把密钥和用户签名公钥关联上，因为任意用户签名公钥都能找到一个K’值运算得到该公钥，也就不能找到是由哪个用户签名公钥运算得到的，签名私钥也同理。或者量子计算破解了对称加密工作密钥，但由于对称加密工作密钥也都是通过中间值K运算得到，并且每个数据中的K值都是不同的，也就无法找到数据之间的关联。所以通过用户主密钥与不同的生成参数，再结合使用具有能抵抗量子计算破解的散列函数，就能使数据或密钥被破解后无法找到与用户的关联，也无法找到数据之间的关联，也就能够不泄漏用户隐私。所以系统未来升级具有安全保障的抗量子计算的密码算法，并不会由于之前公开的数据被破解而泄露用户的隐私。

[0229] 本系统中的交易数据使用UTXO模型，所以每个交易数据都包含未花费输出的引用作为输入和新的未花费输出。引用的方式可以是未花费输出的地址加上所引用的交易标识(ID)，其中交易ID为所引用交易数据的哈希值。但在本系统中，每个交易数据都具有一个唯一的交易数据编号，所以也可以使用交易数据编号替换所引用的交易ID，根据输出地址加上交易数据编号就能唯一确认所引用的未花费输出。以下使用交易数据编号为例说明。新的未花费输出的地址则是根据对应用户的上一个接收交易的生成参数与用户的交易地址主密钥，以及用户的签名公钥生成的新地址。交易数额则是通过加法同态承诺或佩德森承诺实现的机密交易进行保护，使任何用户都可以在密文情况下验证交易数额的输入之和等于输出之和，并结合范围证明验证交易数额都不小于零且不会溢出，也即验证交易数额的有效性。机密交易中的盲化因子则是通过当前交易数据的生成参数与用户的机密交易主密钥生成的工作密钥加密保护，所以用户可以解密自己的密文交易数额。

[0230] 用户端的结构化对等网络是通过分布式哈希表(DHT)，根据索引的key(键)将对应的value(值)分散存储在网络的节点中，可以使用索引key在网络中快速检索对应的value。在本实施例中，将系统中的交易地址、管理地址、token发行或回收使用的预设地址等都通过地址前缀有明确区别，再把交易数据编号和控制数据编号也通过类似地址方式扩展(如补0)并通过前缀明确区别(以下称为编号地址)。比如控制数据编号地址前缀用E表示，序号为1的控制数据用E001作为控制数据编号地址；交易数据编号地址前缀用F表示，序号为1的交易数据用F001作为交易数据编号地址。可以将上述这些地址的哈希值作为索引的key，比如某个账本数据具有一个或多个地址，则将每个地址的哈希值都作为索引的key，对应的账本数据和认证路径数据作为value，存储在结构化对等网络中。任何用户就可以通过key(上

述地址)检索到相应的账本数据和认证路径,再结合区块头数据,根据编号找到账本数据所在的区块,就可以使用默克尔树验证该数据。为简单描述,后文使用地址作为索引的key,这与使用地址的哈希值作为索引的key是等价的。并且与通常DHT的key-value不同,本系统允许有多个value具有相同的key值,并且规定了相同key的不同value的存储与检索方式,也即下面介绍的连接存储(或称结对存储)方式。

[0231] 下面介绍一种通过连接存储结构,将链式结构转化为节点验证的方式。

[0232] 以区块头形成的链为例说明,已知区块头连接(包含)前一个区块头的哈希值。假设第一个区块头H1所包含的值为0000,设H1的哈希值为hash(00H1);则第二个区块头H2所包含的哈希值为hash(00H1),设H2的哈希值为hash(00H2);则第三个区块头H3所包含的哈希值为hash(00H2),设H3的哈希值为hash(00H3)。以区块头的哈希值作为索引key,所包含的前一个区块头的哈希值也作为索引key。则存储hash(00H1)的节点,会存储H1和H2;存储hash(00H2)的节点,会存储H2和H3;以此类推,每个节点都会存储链上的一对数据,并且索引key为该数据对连接的全部或部分信息。例如对于连接是交易ID和输出索引,如果将交易ID作为key,就是部分信息;如果区块头连接的是前一个区块头的哈希值,将哈希值作为key,就是全部信息。下面以哈希值作为key为例进行说明。H1的哈希值为hash(00H1),H2所连接的哈希值为hash(00H1),所以连接存储还需要区分输入数据(以下简称输入)和输出数据(以下简称输出),比如这里输入的H1对应的key(hash(00H1))为数据的哈希值(可看作为输出地址),而输出的H2对应的key为数据连接的哈希值(可看作为输入地址)。这里可以归纳为key对应的是输出地址,则为连接存储的输入数据;key对应的是输入地址,则为连接存储的输出数据。如果链在一直增长,假设当前的最后一个数据为Hn,所以存储hash(00Hn)的节点,只会存储输入数据Hn,而输出数据为空;当链新增加H[n+1]时,该节点才会包含输出的数据H[n+1];而存储hash(00H[n+1])的节点,也只会存储输入的数据H[n+1],输出数据为空。存储0000的节点,只会存储H1,并且由于0000是H1所连接的输入地址,所以H1为该节点的输出数据,而输入数据为空。但由于0000是特殊的起始数据,也即能标识为起始连接存储,所以起始连接存储能使用一个特殊的0000数据(或空数据)作为输入。

[0233] 区块头链有分叉和连接叔区块头的可能,也即区块头链可能出现有多个输入和多个输出的情况。因为叔区块头的哈希值与父区块头的哈希值是不同的,通过哈希值存储也是在不同的节点上,所以区块头链的连接存储不会有多个输入数据,但分叉却会使相关的节点有多个输出数据。根据选择的作为连接的key的不同,如果是数据的哈希值,因为哈希值具有唯一性(不考虑碰撞),则连接存储只会有一个输入数据;如果是地址,根据选择的作为key的地址的不同,可能有多个相同地址的情况,则此时连接存储可能会有多个输入数据。

[0234] 如果节点只有输出数据而没有输入数据(除特殊的起始连接存储外),则该链的连接错误;如果节点只有输入数据而没有输出数据,则该输入数据必定会在上一个节点中连接存储且为输出数据。连接存储是指节点存储一个或多个输入数据和对应的一个或多个输出数据,并且该节点能验证输入、输出数据是否正确;也可以没有输出数据,表示输入数据的连接未被使用或未花费,也即输出数据为零个。其中验证输入输出数据的正确性包括验证一个或多个输入和对应的一个或多个输出的连接是否正确。如果输出数据包含交易数额,还需要验证输出数据的交易数额是否正确;如果输出数据包含签名信息,还可以验证输

出数据的签名信息是否有效。比如UTXO结构中，只有输出的数额，输入是引用的地址，所以验证输出数据的交易数额，其中输入数额的部分就可以来自输入数据的输出数额，也即可以减少部分数据的获取。所以也就把链式结构转换为节点的连接存储，并且由节点验证链的连接是否正确。而且连接存储的方式还能使链具有双向检索的能力，比如上述区块头链查询hash(OOH2)，能获取H2所包含的前一个区块头的哈希值hash(OOH1)，以及H3的哈希值hash(OOH3)。并且检索数据者也可以通过检索的key和节点返回的数据判断是输入数据还是输出数据，从而判断链的连接是否正确。

[0235] 结合区块链，进行连接存储的节点还可以验证连接存储的数据是否为链上的数据。UTXO数据具有默克尔树认证路径，并且UTXO也属于链式结构，下面以UTXO为例说明。因为本系统使用的UTXO引用方式为地址加上交易数据编号，以地址为索引的key。由于存在并发交易的原因，所以可能有多个相同输出地址的情况，但交易数据编号是不同的，所以本系统根据交易地址的连接存储，是可能出现多个输入和多个输出的情况，但相同地址引用的是不同的交易数据编号，也即多个输入和多个输出数据之间，根据地址加上交易数据编号是一对一的引用，所以并不是双花。比如交易数据T1，其中的输入地址为Ad1和Ad2，输出地址为Bd1和Bd2；交易数据T2，输入地址为Bd1和Ed1，输出地址为Cd1和Cd2。根据上文可知，对于存储Bd1的节点，如图9所示，会存储T1和T2，其中T1为连接存储的输入数据(因为Bd1为T1的输出地址)，T2为连接存储的输出数据(因为Bd1为T2的输入地址)。而对于存储Ad1和Ad2的节点，T1数据是连接存储的输出数据；对于存储Cd1和Cd2的节点，T2数据是连接存储的输入数据。每个节点都会验证自己连接存储数据的输入输出是否正确，比如存储Bd1的节点，会验证有无输入数据T1，输出数据T2是否存在双花，输出数据T2的交易数额是否正确，但T2的输入还需要依赖Ed1，所以存储Bd1的节点还需要获取Ed1作为输出的数据才能够验证，设Ed1作为输出的数据为T3，所以存储Bd1的节点，会存储T3以及对应的认证路径作为附加验证数据，并验证Bd1和Ed1的输入数额是否等于Cd1和Cd2的输出数额，即Bd1+Ed1是否等于Cd1+Cd2。同理存储Ed1的节点，会存储输入数据T3和输出数据T2，以及附加验证数据T1，并验证输入输出数据是否正确。

[0236] 下面介绍本系统的token发行和回收也即起始交易数据和结束交易数据也是连接存储的，并且相关节点也能验证连接存储数据的输入输出是否正确。所以也就把UTXO链转化为节点的连接存储结构，每个节点验证连接存储是否正确，比如是否只有输出而没有输入(起始连接存储也会有特殊的输入)，输入数据是否有多个相同输出(即双花)，以及连接存储数据的输入输出是否正确，并且每个交易数据都可以通过默克尔树验证是否为链上的数据，因此通过分散的节点即可验证整个UTXO账本的正确性。根据上文可知，用户在网络中通过检索地址也可以获取双向检索的能力，并且能根据节点返回的数据是输入还是输出，从而判断链的连接是否正确，还可以根据返回的数据判断是否是未花费输出。比如用户C在网络中检索Ed1，节点会将Ed1相关的数据T1、T2和T3以及对应的认证路径返回给用户C。用户C根据Ed1是T3的输出地址，所以T3是输入数据；根据Ed1是T2的输入地址，所以T2是输出数据；Ed1不是T1的地址，所以T1是附加验证数据。根据相关数据的认证路径验证T1、T2和T3都是链上的数据，并且可以验证Bd1和Ed1的输入数额是否等于Cd1和Cd2的输出数额，从而可以验证输入输出数据的正确性。如果节点只返回T2数据，则只有输出而没有输入，判断为链的连接错误；如果节点只返回T3数据，则表示T3的Ed1没有被花费。

[0237] 系统token发行或回收使用的是预设地址，并且首先需要在控制数据集中明文公布。比如发行的地址为A001，回收的地址为B001。因为控制数据集中公布的token发行或回收信息中包含有该预设地址信息，所以网络中存储A001的节点，会存储控制数据集中发布的信息，以及A001作为输入引用的交易数据，也即起始交易数据；而网络中存储B001的节点，会存储控制数据集中发布的信息，以及B001作为输出的交易数据，也即结束交易数据。所以系统token发行或回收的预设地址，也是符合连接存储的特征，只是使用预设的输入或输出数据，比如不是交易数据而是控制数据集中的明文发行或回收信息，节点也可以验证输入输出数据是否正确。

[0238] 上述举例系统使用的是同质化token，如果使用非同质化token，也是可以符合连接存储的特征，并能达到验证UTXO链的目的。

[0239] 上述通过连接存储使分布式节点即可验证链的连接是否正确，此外还可以验证链的长度是否正确。为了验证链的长度是否正确，根据区块头中所包含的累计交易数据量和累计控制数据量，节点能计算出该区块所包含的所有编号地址，然后节点可以根据每个编号地址查询自己的网络标识是否匹配该key，如果有相应的编号地址key是由自己存储，但自己没有相应的数据，则通过网络查询该key的数据(例如先在用户端节点查找，如果没有再将编号地址作为key在第二层链上查找)，如果没有找到则说明该链的完整性错误。如果找到了该key对应的账本数据，并验证了该数据，则说明该编号地址的数据正确，如果该数据是交易数据，根据上述所知，交易数据通过交易地址形成连接存储，该节点可以根据交易的输入引用地址查询和验证。这样就能通过编号地址验证链的完整性，也就能通过网络查询未花费输出。

[0240] 由于本系统的账本数据是由交易数据集和控制数据集两部分组成。并且每个控制数据中都有唯一的控制数据编号，该编号是顺序递增的，根据上文可知，通过编号地址即可验证控制数据的完整性。如果某个地址只被特定节点验证，也即验证某个交易数据或控制数据的节点是固定并且已知的，这会带来一定的风险，所以在一示例性实施例中，系统可选择增加用户随机选取验证的方式。因为区块头中包含了当前所有的累计交易数量和累计控制数量，所以每个节点都能知道区块中所包含的交易数据和控制数据的起始编号与结束编号。用户端节点在同步区块头时，可以根据编号地址对该区块里的账本数据进行随机选取验证，比如可以随机选取验证m1个控制数据，也可以随机选取验证m2个交易数据。控制数据只有管理端的签名，所以用户端节点只需要通过控制数据编号地址获取到控制数据，验证管理端签名，并根据认证路径验证数据是否在链上。验证交易数据则需要先经过交易数据编号地址获取到交易数据，再根据交易数据里的输入引用地址，查询这些地址并进行验证。根据上文所知，查询输入引用的地址节点返回的是具有输入输出数据的连接存储数据，即前述具有关联关系的输入数据和输出数据，并且能够验证输入数据和输出数据的正确性，以及根据认证路径验证数据是否在链上。由于用户端使用的是分布式哈希表网络进行检索，并且每个节点都是独立随机选取账本数据进行验证，所以并不会知晓某个交易数据被哪些节点选择验证，也就能避免某个地址只被特定节点验证的缺点，增加了安全性。

[0241] 从上文可知，可以把任意链上的数据转化为节点的连接存储结构，其中节点存储一个或多个输入数据和对应的零个、一个或多个输出数据，并且节点能验证输入、输出数据是否正确，特殊的起始连接存储和结束连接存储也可以通过预设的输入和输出数据符合连

接存储特征。通过连接存储把任意链上的数据转化为节点存储，再结合证明所存储的数据是链上数据，就能验证链的连接是否正确。以及通过编号地址验证链的完整性，就能使每个节点存储部分账本数据和对应的认证路径，并验证输入、输出数据是否正确，就能等价验证全部账本数据的正确性，称之为等价验证。并且还可以选择增加用户随机选取验证的方式，避免某个地址只被特定节点验证的缺点，增加了安全性。

[0242] 再以公有链的UTXO为例说明。UTXO输入引用的是未花费交易ID和输出索引，其中交易ID为所引用交易数据的哈希值。使用交易ID作为检索的key，引用的交易ID也作为检索的key。比如交易数据ID为Tb，其中的输入引用为Ta[1]和Ta[2]，有两个输出Tb[1]和Tb[2]；交易数据ID为Tc，其中输入引用为Tb[1]；交易数据ID为Td，其中输入引用为Tb[2]；括号内数字为引用的输出索引。则存储Tb的节点，会存储输入数据Tb，以及输出数据Tc和Td。因为不允许有相同的交易ID，所以根据交易ID作为检索的key，连接存储不会有多个输入数据，但可能会有多个输出数据。这里的两个输出的输入引用并不是相同的输出索引，所以并不是双花。每个交易数据都包含对应的认证路径数据，并且每个区块体数据中的第一个交易数据的认证路径是特殊的第一个交易数据，也即起始连接存储。在不考虑交易手续费的情况下，每个节点可以容易验证输入、输出数据的正确性，起始连接存储也可以容易验证。但考虑实际交易数据中含有手续费，如果区块体内的交易数量较多，会使验证变得非常困难。如果能够仅依赖有限个交易数据验证输入、输出数据的正确性，就能发挥分布式节点连接存储的优点，能通过每个节点存储并验证部分交易数据的正确性，达到验证整个UTXO链的正确性的目的。并且可以通过增加编号地址的方式验证链的完整性。

[0243] 因为本系统使用UTXO模型，所以用户A的未花费输出可能会被同一个交易里的其他用户B知道，引用该未花费输出的交易时间戳也可能会被用户B知道，如果交易里还引用了用户A的其它未花费输出，也可能会被用户B知道。为保证尽可能不泄露任何信息，用户可以选择系统提供的辅助混淆方案。比如用户将这些未花费输出依次转入系统提供的混淆地址，该混淆地址也是一条逻辑链，所以每次转入的混淆地址都是不同的，然后由系统的另外一个混淆地址转出相同数额到用户新的接收交易地址，而新的接收交易地址在UTXO链上与用户之前的未花费输出没有任何关联，也就能不泄漏任何信息。

[0244] 用户的管理地址是通过上一个管理数据中的生成参数与管理地址主密钥生成，管理数据则由上一个管理数据中的生成参数与对称加密主密钥生成的对称加密工作密钥加密保护。而用户的第一个管理数据，也即用户的注册管理数据，使用用户的注册ID作为生成参数。用户的注册管理数据中包含接收交易数据组成的第二账户数据链的初始地址生成参数，还包含用户的身份证书散列值和用户签名公钥，可用于确认用户身份。除用户注册管理数据和用户身份证书更新管理数据中包含有用户关联信息用于确认用户身份外，其余的账本数据里都不再含有用户的关联信息。比如交易数据中可以包含用户附加信息，其中可能会含有用户的身份信息，所以用户附加信息是链下存储，链上只记录附加信息的散列值，并且用户附加信息可以使用交易的时间戳作为盐值。除用户附加信息外，交易数据中还可以包含有合约附加信息，合约附加信息是用于记录合约的关联信息，并不含有用户的身份信息，所以合约附加信息可以是链上存储的。用户的签名公钥与用户身份证书是关联的，所以用户身份证书更新管理数据中也包含用户签名公钥更新信息。用户的其它管理数据可以包含用户主密钥更新信息和普通管理信息，这些信息都不会涉及用户身份。

[0245] 用户在注册后，通过注册ID与管理地址主密钥找到注册管理数据，并由注册ID与对称加密主密钥生成的对称加密工作密钥解密数据，从而获取到用户的管理数据组成的第一账户数据链。然后根据注册管理数据中的交易数据初始生成参数、交易地址主密钥以及用户的签名公钥获取到用户接收交易数据组成的第二账户数据链。根据上文可知，在用户端网络查找接收交易地址，因为采用连接存储，所以能获取到接收交易地址作为输入引用的交易，也即用户的发送交易。所以通过获取接收交易的第二账户数据链，就能同时获取到用户所有的交易数据。再由交易数据里的生成参数与机密交易主密钥生成的工作密钥解密盲化因子，从而获取到用户的账户信息。

[0246] 比如用户Alice通过身份证书在管理端注册了ID为Alice的账户，管理端给账户Alice颁发了管理地址主密钥、交易地址主密钥、机密交易主密钥和对称加密主密钥。其中账号ID名称为管理数据的默认初始生成参数。用户通过账户ID 名称Alice和管理地址主密钥生成注册管理地址，然后在用户端网络查找该注册管理地址，即能找到账户Alice的注册管理信息，并通过账户ID名称Alice和对称加密主密钥生成的对称加密工作密钥解密数据。注册管理信息中包含了用户身份证书哈希值，可以证明用户的身份，还有生成参数用于生成下一个管理数据的地址和对称加密工作密钥，并包含接收交易地址的初始生成参数，用于生成该用户的第一个接收交易地址。当给账户Alice转账后，用户就能通过上一个接收交易的生成参数、交易地址主密钥和用户的签名公钥生成接收交易地址，然后在用户端网络查找该地址，即能找到交易信息，并通过交易数据中的生成参数和机密交易主密钥生成的工作密钥解密盲化因子，即能解密密文交易数额。如果Alice需要给Bob转账，则需要通过上一个接收交易的生成参数、交易地址主密钥和用户的签名私钥生成解锁脚本。

[0247] 当用户需要向第三方用户证明某个地址的数额时，只需要提供中间值K与用户的身份证书，以及盲化因子即可。第三方用户可以验证身份证书的有效性确认用户身份，然后用K与用户签名公钥做椭圆曲线上的标量乘法运算得到新的公钥，验证新公钥生成的地址等于该地址。因为椭圆曲线上的标量乘法运算具有单向性，因此无法找到一个K’与用户的签名公钥运算得到其他人的公钥地址(假设为量子计算破解出现之前，之后需要更新能抵抗量子计算破解的算法，才能提供有效证明)。再通过网络查询该地址是否是未花费地址，最后使用盲化因子解密密文数额。而用户提供的信息里都不包含相关密钥信息，第三方用户无法根据所提供的信息，获取到用户其它交易数据的隐私。上述查询未花费地址，是通过节点的连接存储实现的。根据上文所述，如果该地址已被花费，则相关节点会返回输入数据和输出数据；未花费则只会返回输入数据而没有输出数据。但该查询方式依赖于相关节点，根据后文将可知，第三方用户也可以通过第二链系统查询未花费地址的信息。

[0248] 系统发布的公告数据都是明文，比如token发行或回收的公告信息、密钥算法更新信息、新交易规则发布信息以及系统版本更新信息等。根据公告类型的不同，每个公告都有公告类型和递增的序号，根据公告前缀+公告类型+公告序号，组成公告的地址，任何用户都可以通过公告地址在网络中检索并验证公告。

[0249] 所以本系统根据交易地址(包括起始交易数据和结束交易数据的特殊地址)的UTXO链是连接存储的，可以验证UTXO链是否正确；编号地址等顺序递增的地址并不是链式关系，所以是直接存储的，可以用于验证链的完整性以及随机选取验证；管理地址不是显性的链式关系(具有用户主密钥才能获取该用户的第一账户数据链，管理地址形成隐性的链

式关系),以及公告地址等,也是直接存储的。直接存储的key对应的value具有唯一性,检索返回相应数据;连接存储的key可能对应多个value数据,也即连接存储的输入、输出数据,检索返回相关的所有数据,包括附加验证数据。并且用户端通过结构化对等网络,使用分布式哈希存储方式,每个用户端只需要存储部分账本数据,就可以通过地址检索到对应的账本数据,并且可以使用默克尔树验证该账本数据是否包含在链的区块中。

[0250] 公有链是指任何人都可以读取、发送交易和参与共识的区块链系统,属于完全去中心化的系统。类公有链是指不包括任何人都可以发送交易外,其余都与公有链相同的系统,满足交易中心化,账本数据去中心化。下面以公有链系统阐述类公有链的系统架构。

[0251] 假设有一个公有链系统S,其中有一个用户A,A发送的交易数据需满足UTXO的规则,也即要有一个合法的输入引用,并且不能够有双花存在。如果不考虑有多个输入输出的情况,则S链上A的交易数据将形成一条顺序单连接的UTXO链。将A假设为一个私有链系统(S仍可以将A看作为一个用户),并将A的交易数据替换为私有链系统产生的区块数据,因为可将私有链看作为后一个区块连接(花费)了前一个区块的输出,可以把私有链看作为UTXO链,所以A所产生的区块数据连接上一个区块数据,并且不能够分叉(不能够有双花),也即可以把S链上A的UTXO链与A的私有链看作为等价的。当A的区块数据上链时,S系统需要验证A区块数据的合法性,需顺序连接且不能分叉,还需要验证区块里账本数据的合法性。S系统是任何人都可以参与共识,读取账本数据以及验证账本数据,所以任何人也都可以验证A的账本数据。

[0252] S系统除A的区块数据外,还可以有扩展数据,但该扩展数据并不对用户的管理数据和交易数据产生影响,也不会影响用户的账户状态,所以用户端并不需要读取和验证该扩展数据。扩展数据的作用是使S能够与A进行交互,使S能够部分影响A数据的产生,所以该数据只有参与S共识的对象和A会读取。比如A系统在控制数据中发布token发行或回收的信息之前,需要经过S进行投票;或者由S生成可证公平随机数,通过该随机数影响A系统产生的数据,以解决A系统的部分公平性问题。用户端C并不需要同步A私有链的区块头数据,而只需要同步S系统的区块头数据,所以A私有链的区块头数据也可以存储在扩展数据中。如果系统中有不合法的数据产生,需要能够对数据进行举证时,也可以将相关的数据存储在S系统的扩展数据中。如果系统使用了POA(Proof Of Activity,权威证明)等共识算法,相关验证者选举和验证者列表的数据也可以存储在扩展数据中。

[0253] A系统的用户端C,发起的交易数据需要经过A再到S,C从S上获取到交易数据。可以将上述系统S、系统A(私有链或联盟链系统),以及A系统的用户端C,看作一种类公有链应用系统。任何人都可以读取、验证交易,并参与共识。这里是指读取S链的数据,用户端C也同步S链的区块头数据。根据上文可知,区块头的一致性能保证系统状态的一致性,所有用户端同步一致的S链区块头,就能保证所有的用户端以及系统S的状态都是一致的。并且用户端只需要从A系统获取用户主密钥(密钥不属于链数据),就能在用户端的结构化对等网络中(或S链中)自行检索账户数据,而第三方用户和监管者也可以在被授权密钥后自行检索,该过程并不依赖A系统或其它中心化的系统。所以类公有链应用系统是可验证、可追溯、不可篡改的。但因为用户的交易数据首先需要经过A才能在S中上链,所以类公有链应用系统并不解决交易的公平性问题,但对于通常的支付系统,比如在线购买音乐等,购买的顺序并不会对结果有影响,只要合规的交易都能正常上链。并且有一类可延迟选择优先的公平性问

题,类公有链应用系统也是可以解决的。

[0254] 可延迟选择优先与时间优先不同,是通过生成可证公平随机数,然后根据该随机数决定优先权的方式,因为是任何人都能够参与,所以是公平的。可延迟选择优先,是系统S生成一个可证公平随机数,延迟若干个区块公开,系统A通过该随机数来决定优先权,以解决此类问题的公平性。

[0255] 下面介绍系统的架构,系统是由三层二链架构组成,如图10所示。

[0256] 第一层也是第一链系统,可以是私有链或联盟链,属于中心化的系统(例如上述系统A),可以采用私有网络以及PBFT(Practical Byzantine Fault Tolerance,实用拜占庭容错算法)等共识算法,以满足快速确认和高频交易的需求,由具有实名的主体对象负责,是系统的管理者。包括以下管理的一种或多种:用户和机构的管理、token的发行或回收、用户主密钥的颁发和系统公告等,主要通过发布控制数据实现系统管理,然后写入第一链中。用户端的交易数据通过管理端验证后写入第一链中,第一链生成的区块数据会立即向第二层广播,也可以立即向第三层广播。因为管理端具有所有的用户主密钥,所以管理端能生成系统的状态树,管理端查询和修改用户账户状态等操作是通过系统的状态树实现的,然后转换为底层的UTXO交易数据或控制数据。管理端会验证用户端的交易数据,并且会验证相关的用户身份信息。每个交易数据或控制数据都会被分别赋予顺序递增的唯一序号,连同交易数据或控制数据一起经过管理端的背书签名。第一链的区块头数据中会包含当前的累计交易数量和累计控制数量,并且区块头数据会经过管理端的签名。

[0257] 第二层也是第二链系统,是类公有链(上述系统S),采用非结构化对等网络,比如可以使用gossip网络协议,可以采用共识时间较长的算法。第二层会对第一链的区块数据进行验证,但由于第二层并没有用户主密钥,所以只会验证交易数据,而不会验证相关的用户身份信息,也不会生成系统的状态树。根据上文可知,系统采用UTXO模型,并且任何人都可以在密文情况下验证交易数额的正确性。第二链的区块数据是由第一链的一个或多个区块数据顺序组成,第一链上任一账户的状态与第二链上所述账户的状态一致,所以系统是异步同态的。因为第二链的区块数据可能是由多个第一链的区块数据顺序组成,第一链区块体的账本数据需要重新顺序组成第二链区块体的账本数据,包括交易数据和控制数据,然后分别重新生成默克尔树的根哈希值记录在第二链的区块头中,第二链的区块头也会包含当前的累计交易数量和累计控制数量。因为第二链区块体的账本数据重新顺序组成,所以也是满足已知默克尔树的叶子节点数量和节点序号,对应的认证路径的高度和方向是固定且已知的。第二链还可以包含扩展数据,该数据可以是第二链系统自身的状态数据,并不会修改用户的状态,第三层也不会同步和读取该数据。比如该数据可以是第二链的投票过程或者是生成可证公平随机数的过程等,也可以是第二链投票选取链生成节点的过程,还可以是用于存证的数据,该扩展数据只会由第一层和第二层读取,并且可能会对第一链或第二链后续产生的区块产生影响。第一链的区块头数据也可记录在第二链的扩展数据中,并不会影响用户状态,而是作为存证的数据。第二链生成的区块数据会向第三层广播,广播的数据是第二链的区块头数据、交易数据和控制数据以及对应的认证路径数据等。

[0258] 第三层是用户端系统,采用结构化对等网络,比如可以使用kademlia网络协议,通过分布式哈希表(DHT)检索数据,每个用户端节点(以下简称节点)只需要存储部分账本数据以及对应的认证路径。根据上文可知,第三层通过等价验证,每个节点验证自己的部分账

本数据，即可等价验证全部的账本数据，再结合每个节点独立随机选取账本数据进行验证，避免某个地址只被特定节点验证，增加了安全性。第三层的每个节点都会同步第二层的区块头数据，所以第三层与第二层系统状态是一致性的。用户端、第三方用户和监管者的读取操作可以是在第二层或第三层，用户端交易数据的写入操作则是通过管理端在第一层上链，所以系统是读写分离的。并且第一链可以立即向第三层广播，所以用户端的交易数据是能及时获取的，但此时的交易数据是没有上第二链的，只是上了第一链。如果是小额交易，用户端可以立即信任管理端发布的数据；但如果交易数额较大，用户端可以等待一段时间，等该交易数据在第二链上链，并经过n个区块确认，就可以认为该交易数据不可逆且无法被篡改了。

[0259] 由于第二层系统和第三层系统状态是一致性的，并且第三层通过等价验证对全部的账本数据进行了验证，所以第二层系统可以选择只验证第一链的区块头数据、账本数据的默克尔树和账本数据的管理端签名，然后将数据上链生成第二链的区块数据，并向第三层广播，由第三层验证UTXO链的连接是否正确，以及验证每个交易数据里的用户端解锁签名和交易数额是否正确。因为第三层的每个节点只需要验证少量账本数据，所以即使用户端使用多重签名的方式，并且需要验证密文交易数额是否正确，每个节点的负担都较少，即使是移动设备也能够完成验证。而第二层验证管理端的签名，还可以使用优化的批量验证单个签名者生成的多个签名，大大降低了参与第二层共识所需要验证的计算量。并且根据默克尔树的特点，第二层的链生成节点可以将第一层产生的账本数据，顺序分散到多个物理设备上，将这些物理设备上的账本数据联合生成默克尔树的根哈希值，也即可以通过多个物理设备完成一个含有非常多账本数据的巨大区块体数据的生成和验证，并且存储也可以分散到多个物理设备上。不同于第三层使用分布式哈希表存储的方式，该方式属于线性划分存储，也即把一个巨大区块体数据，根据默克尔树的特点，某个父节点对应的叶子节点数是2的幂次方，所以把数据按2的幂次方个线性划分到多个物理设备上进行存储，然后可以联合生成对应的默克尔树根哈希值，并且还可以生成账本数据对应的认证路径。所以第二层的链生成节点，只需要多个普通的物理设备和存储设备，并且验证的计算量也可控，实际参与共识生成区块头的设备也只需要普通设备即可，大大降低了参与第二层共识的门槛。

[0260] 上述方式使用了先生成后验证的方式，也即先生成第二链，再由第三层进行验证的方式，因为实际的账本数据是由第一层系统上链的，第二层会验证默克尔树和管理端的签名，并不会修改账本数据。所以如果第三层验证账本数据错误，并且经过管理端的签名(后面还需第二层验证是否为第一链的数据)，责任人是第一层系统，并且第一层是具有实名的管理端，可由监管者进行相应处理。因为第一层是中心化的系统，如果产生了非法的账本数据，第二层和第三层也是无法阻止的，但可以立即验证出来，然后采取相应的措施。比如第三层将非法的账本数据由第二层写入类公有链扩展数据的存证数据中，而第一层系统也是无法修改第二链的数据，并且该数据是公开的任何人都可以访问，就可以由监管者进行相应处理。所以该系统虽然无法保证上链片段所处时间段内的数据一定正确(因为该数据由中心化生成)，但却可以保证错误的数据无所遁形(由第二层或第三层验证)。由第二链上链且经过n个区块确认的数据是正确的，并且是无法篡改的，所以是可信任的。而第一层系统也是采用私有链或联盟链，减少和防止数据出错的可能性。并且根据上述可知，不允许

第一链产生分叉,第一链可以采用PBFT等共识算法防止分叉。如果第一链产生分叉,第二层系统能检测出来并写入类公有链扩展数据的存证数据中,由监管者进行相应处理。

[0261] 第一层的管理端系统也可以通过连接存储的方式优化处理需要验证和存储的海量数据。比如可以使用一致性哈希算法,根据交易地址和交易ID,将交易数据分散到多个共识组中,然后再通过属性分组的PBFT(实用拜占庭容错算法)共识算法,生成第一链的区块头数据。

[0262] PBFT算法是一种状态机副本复制算法。将所有的副本组成的集合的数量设为N,假设失效的副本数量是F,则需要N>3F。每个节点具有一个状态机副本,所以PBFT算法可以容忍小于N/3个无效或者恶意的节点。但PBFT的缺点是具有O(N^2)的消息复杂度,所以通常N不会很大。

[0263] 属性分组的PBFT共识算法,是使用一致性哈希算法,根据数据的属性,将数据分散到多个共识组中,只需要在其中的某一组中进行状态机副本复制。比如将N分为M个共识组,每个共识组中有n个节点,也即N=M*n,并将这些共识组设为1到M序号。

[0264] 产生区块之前,使用一致性哈希算法将交易数据的每个输入地址都映射到1到M中的一个,将交易ID也即交易数据的哈希值也映射到1到M中的一个,然后将交易数据存储到所映射的共识组中,根据上文可知,输入地址对应的是连接存储的输出数据。产生区块之后,也即数据上链后,再用同样方式将交易数据的每个输出地址都映射到1到M中的一个,并且包括上链的控制数据中的token发行地址,然后将相应的上链数据存储在所映射的共识组中。根据上文可知,输出地址对应的是连接存储的输入数据,所以共识组存储的连接存储的输入数据是上链后的数据,共识组可以通过默克尔树验证该数据。

[0265] 如果共识组存储对应的是输入数据,则存储即可;如果共识组存储对应的是输出数据,则需要查找对应的输入数据,并且验证输入输出的正确性,然后形成连接存储;如果共识组存储对应的是交易ID,则需要根据交易数据所有的输入引用地址,向地址对应的共识组发起PBFT共识验证。可以知道这些地址对应的共识组如果验证通过,则会形成连接存储,返回验证成功,如果验证未通过则返回验证失败。根据上文可知,当每个地址返回验证成功的数量都大于n*2/3时,则该交易数据验证通过,可以上链。所以交易数据是由交易ID对应的共识组发起验证和上链的,并且因为交易ID是唯一的,所以不会重复上链。为优化验证交易数额所需的附加验证数据,交易数额可以由交易ID对应的共识组验证,输入引用地址对应的共识组只需要验证连接是否正确,以及用户端解锁签名是否有效,并返回交易数额数据,并不需要附加验证数据。

[0266] 一时间片段后,每个共识组独立发起PBFT共识上链,由每个共识组分别独立共识出自己的上链集合的交易数据以及顺序。然后由1号共识组将自己的上链集合数量加上累计数量,向2号共识组发出累加后的数量消息;然后由2号共识组将自己的上链集合数量加上累计数量,向3号共识组发出累加后的数量消息;直到最后的共识组M,M再向1号共识组发出累加后的数量消息;当1号共识组收到累计数量消息后,则完成一次循环作业,将循环内的交易数据联合上链。每个共识组根据累计数量计算出起始序号,然后将上链集合的交易数据以及对应顺序赋予递增的序号并请求签名(例如n个节点,则需要大于n*2/3的请求才给予签名),最后将签名后的数据联合生成默克尔树的根哈希值。因为是独立共识上链,所以每个共识组可以接收到数量消息后再进行共识上链,通过消息传递完成循环作业。共识

组可以独立共识上链的原因，是因为连接存储的输入数据是上链后的数据，所以同一时间片段内，不会出现两个之间相引用的合法交易。

[0267] 因此可知，属性分组的PBFT共识算法消息复杂度是O(n^2)，可以容忍小于n/3个无效或者恶意的节点。但由于是有M个分组，交易数据也是分散到M个分组中，所以最后能处理的交易数据的数量就能得到较大提高。

[0268] 系统的控制数据不是UTXO链，产生时即可赋予递增的序号并进行签名，然后根据序号线性划分存储和验证，也可以联合生成控制数据的默克尔树根哈希值。最后根据交易数据的根哈希值和控制数据的根哈希值，以及相应的累计数量生成第一链的区块头数据。所以第一链的管理端系统是可以通过连接存储的方式优化处理海量的数据。

[0269] 根据上述可知，第三层用户端存储的数据，包括区块头数据、账本数据以及对应的认证路径数据。账本数据实际是由第一层的管理端生成，区块头数据是由第二层共识生成，并且区块头数据也能映射区块体数据(账本数据)的一致性，而且区块头数据还能确认对应的认证路径，所以区块头数据的正确同步对于系统的正确性非常重要。但第三层用户端并不参与第二层的共识，如果第二层的类公有链系统采用POA(权威证明)等共识算法，需要第三层用户端能够正确同步共识者列表，这会增加用户端的负担。因为第二链是在经过验证第一链区块头的基础上生成的，所以第三层用户端可以在信任第一链区块头的基础上对第二链的区块头进行验证。

[0270] 举例来说，首先第一层的管理端需要在第三方的公有链上建立一个智能合约。该合约的功能是任何用户均可上传共识公钥(共识公钥是由用户自己产生的一非对称密钥，可用于参与第二层的共识)，合约会将当前上传的所有共识公钥组成的共识公钥集合映射为一个值，该共识公钥集合和映射方法都是公开的，能很容易找到集合中元素存在集合中的证明，但很难找到一个不在集合中元素存在集合中的证明，比如可以使用默克尔树证明或累加器证明。其中一个密码学上的累加器是一个单向的隶属函数，它可以用于识别一个候选是否为一个集合的成员，且不会在过程中暴露集合中的成员。第一链产生的区块头中包含当前该共识公钥集合映射的值，还可以包含共识公钥集合中元素的数量，并且一个密钥代表一个固定的权益，即为一密钥一票。第二链产生的区块头数据，会连同第一链对应的最后一个区块头数据对第三层广播。第二链区块头包含当前所有的累计交易数量和累计控制数量，以及对应的共识公钥，第二链区块头数据经过共识私钥的签名。第三层的用户端同步第二链的区块头数据，可根据共识公钥验证该共识私钥的签名，验证第一链区块头的管理端签名，验证第一链区块头的累计交易数量和累计控制数量与第二链区块头的累计交易数量和累计控制数量是否相等，相等则说明数量正确并且第一链区块头是第二链区块头产生时间片段内的最后一个区块头数据，所以第一链区块头中的共识公钥集合的映射值是该时间片段内最新的。用户端还可以根据第一链区块头中的共识公钥集合的映射值采用默克尔树证明或累加器证明，验证第二链区块头中的共识公钥是否有效，验证通过则说明第二链区块头有效，可以加入候选主链的区块头。这样用户端就能通过信任第一链区块头的基础上对第二链的区块头进行验证，而无需同步第二链的共识者列表，并且可以通过第一链区块头的累计交易数量和累计控制数量验证第二链区块头的累计交易数量和累计控制数量是否正确，以保证能正确同步第二链的区块头数据。

[0271] 第二层的类公有链系统，也可以采用POA等共识算法，这样可以不依赖第三方的公

有链，但需要参与的共识者对象提供身份证明。第一链区块头也可以包含POA共识算法的验证者列表的映射值和验证者的数量，以方便用户端能正确同步第二链的区块头数据。

[0272] 如果第一链区块头没有按照规则包含正确的共识者公钥集合的映射值，因为该共识者公钥集合的映射值是在第三方公有链的合约中或第二层类公钥链的扩展数据中存储的，都是公开且无法被篡改的。第二层系统可以将第一链的不合规区块头数据写入类公有链扩展数据的存证数据中，由监管者进行相应处理。

[0273] 所以本系统分别使用区块链实现账本数据可溯源、不可篡改，UTXO链实现交易数据的正确连接，确保系统的token总量是一定的，账户数据链实现用户账户数据的隐私检索。并且通过等价验证，使用户端通过结构化对等网络存储和验证部分账本数据，即可等价验证全部账本数据的正确性。

[0274] 本公开一示例性实施例还提供一种计算机存储介质，所述计算机存储介质存储有计算机程序；所述计算机程序被执行后，能够实现前述一个或多个示例性实施例提供的方法，例如，执行如图1、图3、图5及图7所示方法中的一个或多个。所述计算机存储介质包括在用于存储信息(诸如计算机可读指令、数据结构、程序模块或其他数据)的任何方法或技术中实施的易失性和非易失性、可移除和不可移除介质。计算机存储介质包括但不限于RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光盘存储、磁盒、磁带、磁盘存储或其他磁存储装置、或者可以用于存储期望的信息并且可以被计算机访问的任何其他的介质。

[0275] 本公开一示例性实施例还提供了一种计算机装置(或称计算机设备)。所述计算机设备可包括处理器、存储器及存储在所述存储器上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本公开中数据存储装置或数据验证装置所执行的操作。

[0276] 如图11所示，在一个示例中，计算机设备(节点)可包括：处理器91、存储器92、总线系统93和收发器94，其中，该处理器91、该存储器92和该收发器94通过该总线系统93相连，该存储器92用于存储指令，该处理器91用于执行该存储器92存储的指令，以控制该收发器94发送信号。例如上述数据存储装置中第二存储模块的操作可由收发器在处理器的控制下执行，第一验证模块的操作可由处理器执行。

[0277] 应理解，处理器91可以是中央处理单元(Central Processing Unit，简称为“CPU”)，处理器91还可以是其他通用处理器、数字信号处理器(DSP)、专用集成电路(ASIC)、现成可编程门阵列(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。

[0278] 存储器92可以包括只读存储器和随机存取存储器，并向处理器91提供指令和数据。存储器92的一部分还可以包括非易失性随机存取存储器。例如，存储器92还可以存储设备类型的信息。

[0279] 总线系统93除包括数据总线之外，还可以包括电源总线、控制总线和状态信号总线等。但是为了清楚说明起见，在图11中将所有总线都标为总线系统93。

[0280] 在实现过程中，该计算机设备所执行的处理可以通过处理器91中的硬件的集成逻辑电路或者软件形式的指令完成。即本公开实施例所公开的方法的步骤可以体现为硬件处理器执行完成，或者用处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机

存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等存储介质中。该存储介质位于存储器92，处理器91读取存储器92中的信息，结合其硬件完成上述方法的步骤。为避免重复，这里不再详细描述。

[0281] 本领域普通技术人员可以理解，上文中所公开方法中的全部或某些步骤、系统、装置中的功能模块/单元可以被实施为软件、固件、硬件及其适当的组合。在硬件实施方式中，在以上描述中提及的功能模块/单元之间的划分不一定对应于物理组件的划分；例如，一个物理组件可以具有多个功能，或者一个功能或步骤可以由若干物理组件合作执行。某些组件或所有组件可以被实施为由处理器，如数字信号处理器或微处理器执行的软件，或者被实施为硬件，或者被实施为集成电路，如专用集成电路。这样的软件可以分布在计算机可读介质上，计算机可读介质可以包括计算机存储介质(或非暂时性介质)和通信介质(或暂时性介质)。如本领域普通技术人员公知的，术语计算机存储介质包括在用于存储信息(诸如计算机可读指令、数据结构、程序模块或其他数据)的任何方法或技术中实施的易失性和非易失性、可移除和不可移除介质。计算机存储介质包括但不限于 RAM、ROM、EEPROM、闪存或其他存储器技术、CD-ROM、数字多功能盘(DVD)或其他光盘存储、磁盒、磁带、磁盘存储或其他磁存储装置、或者可以用于存储期望的信息并且可以被计算机访问的任何其他的介质。此外，本领域普通技术人员公知的是，通信介质通常包含计算机可读指令、数据结构、程序模块或者诸如载波或其他传输机制之类的调制数据信号中的其他数据，并且可包括任何信息递送介质。

[0282] 以上显示和描述了本申请的基本原理和主要特征和本申请的优点。本申请不受上述实施例的限制，上述实施例和说明书中描述的只是说明本申请的原理，在不脱离本申请精神和范围的前提下，本申请还会有各种变化和改进，这些变化和改进都落入要求保护的本申请范围内。

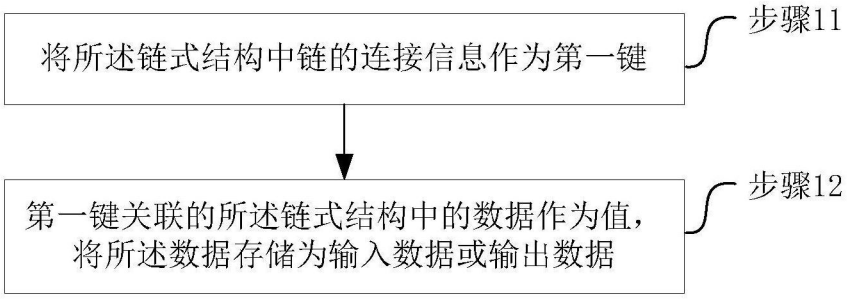

将所述链式结构中链的连接信息作为第一键
步骤11
第一键关联的所述链式结构中的数据作为值，将所述数据存储为输入数据或输出数据
步骤12

图1

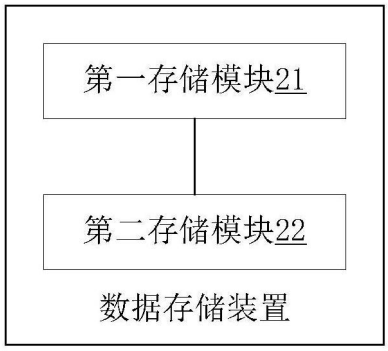

第一存储模块21
第二存储模块22
数据存储装置

图2

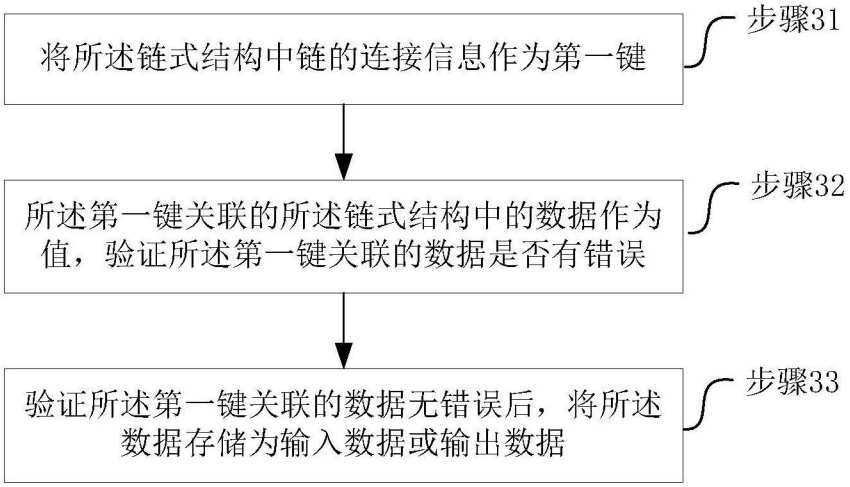

将所述链式结构中链的连接信息作为第一键
步骤31
所述第一键关联的所述链式结构中的数据作为值，验证所述第一键关联的数据是否有错误
步骤32
验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据
步骤33

图3

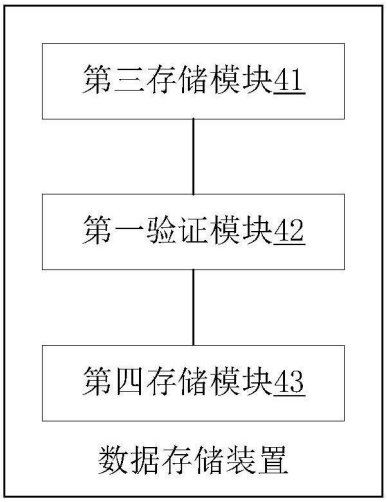

第三存储模块41
第一验证模块42
第四存储模块43
数据存储装置

图4

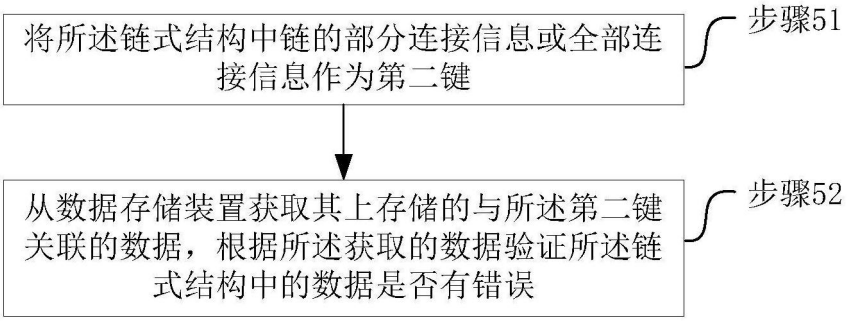

将所述链式结构中链的部分连接信息或全部连接信息作为第二键
步骤51
从数据存储装置获取其上存储的与所述第二键关联的数据，根据所述获取的数据验证所述链式结构中的数据是否有错误
步骤52

图5

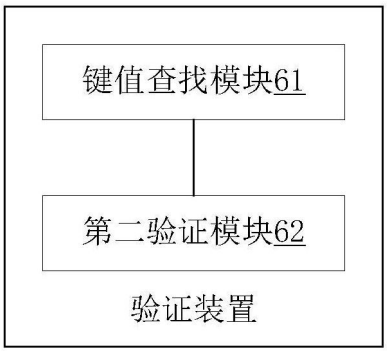

键值查找模块61
第二验证模块62
验证装置

图6

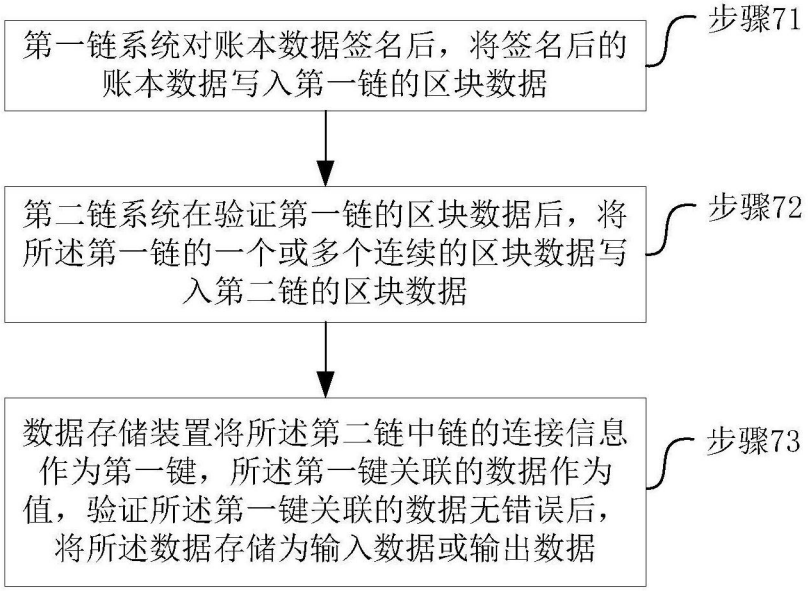

第一链系统对账本数据签名后，将签名后的账本数据写入第一链的区块数据
步骤71
第二链系统在验证第一链的区块数据后，将所述第一链的一个或多个连续的区块数据写入第二链的区块数据
步骤72
数据存储装置将所述第二链中链的连接信息作为第一键，所述第一键关联的数据作为值，验证所述第一键关联的数据无错误后，将所述数据存储为输入数据或输出数据
步骤73

图7

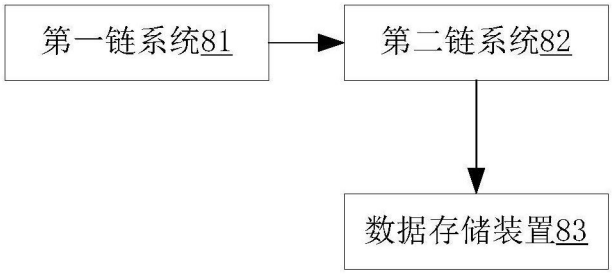

第一链系统81
第二链系统82
数据存储装置83

图8

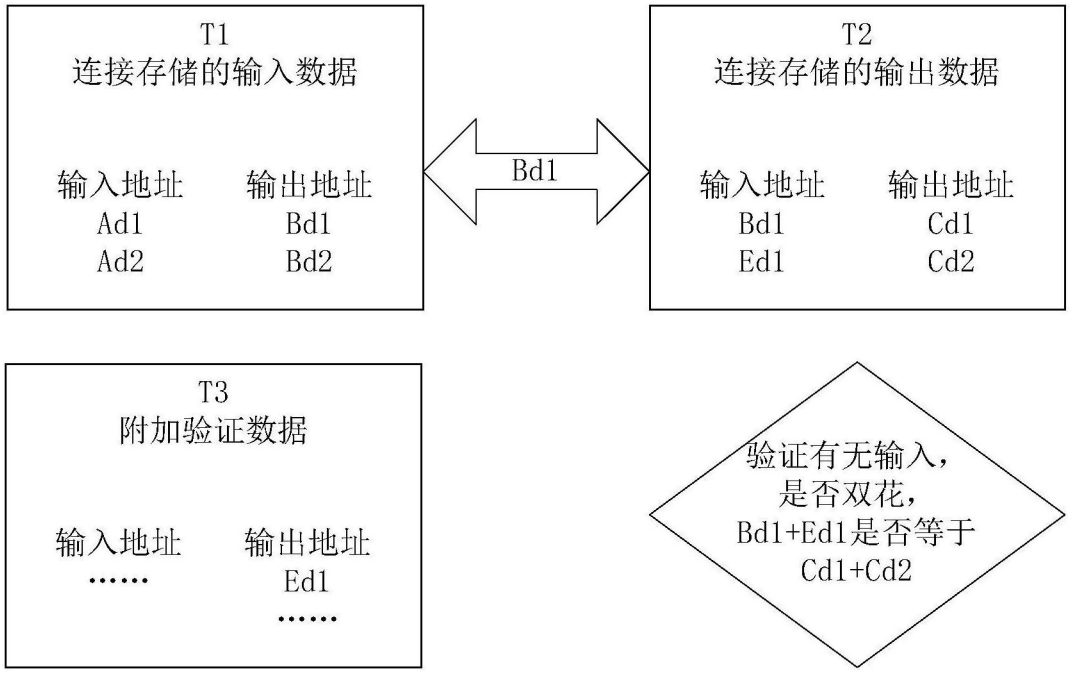

T1
连接存储的输入数据
输入地址
输出地址
Ad1
Bd1
Ad2
Bd2
Bd1
T2
连接存储的输出数据
输入地址
输出地址
Bd1
Cd1
Ed1
Cd2
T3
附加验证数据
输入地址
输出地址
……
Ed1
……
验证有无输入，
是否双花，
Bd1+Ed1是否等于
Cd1+Cd2

图9

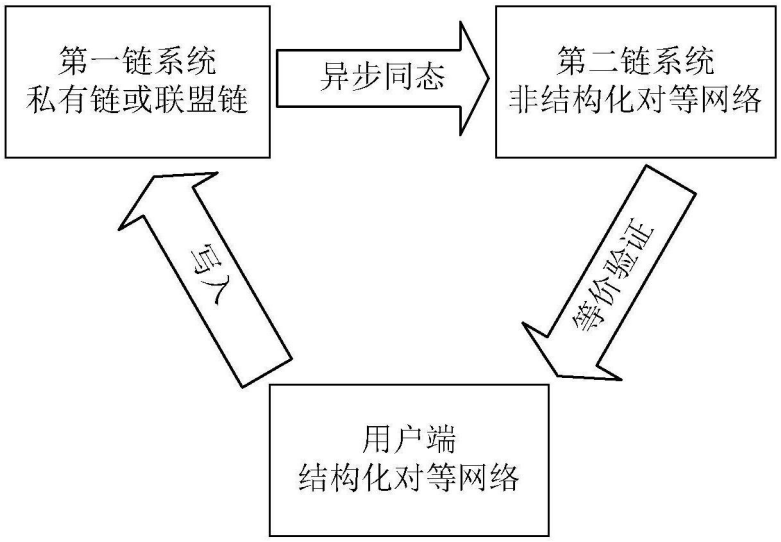

第一链系统
私有链或联盟链
异步同态
第二链系统
非结构化对等网络
写入
等价验证
用户端
结构化对等网络

图10

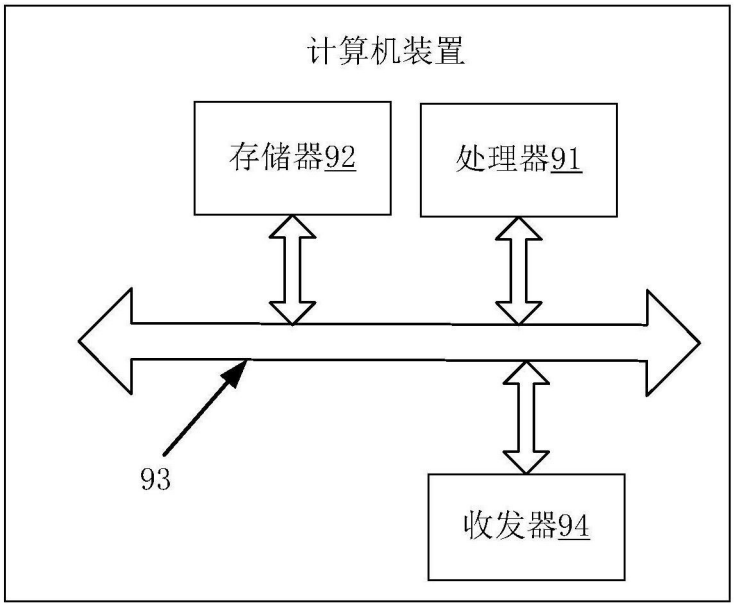


图11