(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112989405 A
(43)申请公布日 2021.06.18

(21)申请号 202110161013.7

(22)申请日 2021.02.05

(71)申请人 武汉北大高科软件股份有限公司
地址 430000 湖北省武汉市洪山区野芷湖西路创意天地10号楼13层1306室

(72)发明人 卢赓 罗铮 王涛 邓昕 颜培耀

(74)专利代理机构 深圳国新南方知识产权代理有限公司 44374
代理人 周雷

(51)Int.Cl.
G06F 21/62(2013.01)
G06F 21/64(2013.01)
G06F 16/61(2019.01)
G06F 16/71(2019.01)
G06F 16/901(2019.01)

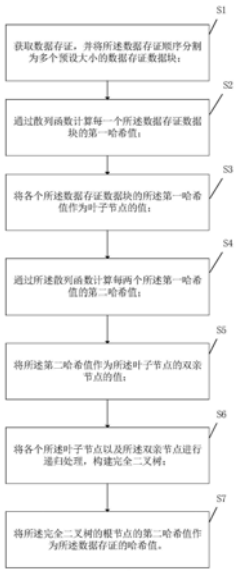
权利要求书2页 说明书10页 附图6页

(54)发明名称

一种数据存证的可信化存储方法、装置、设备和存储介质

(57)摘要

本发明公开了一种数据存证的可信化存储方法，应用于数据加密技术领域领域，用于防止证据数据被篡改，并能够分布式存储音频视频证据以防止证据灭失。本发明提供的方法包括：获取数据存证，并将数据存证顺序分割为多个预设大小的数据存证数据块；通过散列函数计算每一个数据存证数据块的第一哈希值；将各个数据存证数据块的第一哈希值作为叶子节点的值；通过散列函数计算每两个第一哈希值的第二哈希值；将第二哈希值作为叶子节点的双亲节点的值；将各个叶子节点以及双亲节点进行递归处理，构建完全二叉树；将完全二叉树的根节点的第二哈希值作为数据存证的哈希值。每一个分片内容的修改都会导致根节点哈希值的改变，达到文件无法被篡改的目的。



CN 112989405 A

1.一种数据存证的可信化存储方法,其特征在于,包括以下步骤:

获取数据存证,并将所述数据存证顺序分割为多个预设大小的数据存证数据块;

通过散列函数计算每一个所述数据存证数据块的第一哈希值;

将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值;

通过所述散列函数计算每两个所述第一哈希值的第二哈希值;

将所述第二哈希值作为所述叶子节点的双亲节点的值;

将各个所述叶子节点以及所述双亲节点进行递归处理,构建完全二叉树;

将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

2.根据权利要求1所述的可信化存储方法,其特征在于,在所述通过散列函数计算每一个所述数据存证数据块的第一哈希值的步骤中,包括:

对所述散列函数进行常量初始化,得到8个32位的哈希初值和4个32位的哈希常量;

将所述数据存证数据块转换成二进制形式以得到二进制数据块,并对所述二进制数据块的比特位进行填充,得到目标消息;

根据所述8个32位的哈希初值和4个32位的哈希常量和预设的逻辑函数对所述目标消息进行计算,得到所述第一哈希值。

3.根据权利要求2所述的可信化存储方法,其特征在于,在所述将所述数据存证数据块转换成二进制形式以得到二进制数据块,并对所述二进制数据块的比特位进行填充,得到目标消息的步骤中,包括:

获取所述二进制数据块的长度;

在所述二进制数据块的末尾先补第一个比特位1;

在补1后的所述二进制数据块的末尾补比特位0,直到长度满足对256取模后余数是192;

在所述二进制数据块的末尾添加一个64位的比特串,得到所述目标消息,所述比特串是以二进制形式表示的所述二进制数据块的长度。

4.根据权利要求1所述的可信化存储方法,其特征在于,若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同,所述可信化存储方法还包括:

遍历所述双亲节点,定位到所述第二哈希值发生变化的所述双亲节点;

遍历所述第二哈希值发生变化的所述双亲节点中的所述叶子节点,定位所述第一哈希值发生变化的所述叶子节点;

定位所述第一哈希值发生变化的所述叶子节点对应的所述数据存证数据块。

5.根据权利要求4所述的可信化存储方法,其特征在于,所述可信化存储方法还包括:

将所述数据存证数据块根据所述完全二叉树保存在点对点网络中;

若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同,则根据所述完全二叉树从所述点对点网络中提取被篡改的所述数据存证数据块。

6.根据权利要求4所述的可信化存储方法,其特征在于,所述可信化存储方法还包括:

当需要播放所述数据存证时,遍历所有的所述叶子节点;

依次获取所有的所述叶子节点所对应的所述数据存证数据块。

7.根据权利要求1所述的可信化存储方法,其特征在于,在所述获取数据存证,并将所述数据存证顺序分割为多个预设大小的数据存证数据块的步骤中,若所述数据存证数据块

的数量为奇数时，复制最后一个所述数据存证数据块的副本，使得所述数据存证数据块的数量为偶数。

8.一种数据存证的可信化存储装置，其特征在于，包括以下模块：

数据存证分割模块，用于获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块；

第一哈希值计算模块，用于通过散列函数计算每一个所述数据存证数据块的第一哈希值；

叶子节点构建模块，用于将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值；

第二哈希值计算模块，用于通过所述散列函数计算每两个所述第一哈希值的第二哈希值；

双亲节点构建模块，用于将所述第二哈希值作为所述叶子节点的双亲节点的值；

完全二叉树构建模块，用于将各个所述叶子节点以及所述双亲节点进行递归处理，构建完全二叉树；

根哈希值生成模块，用于将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

9.一种计算机设备，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，所述处理器执行所述计算机程序时实现如权利要求1至7中任一项所述数据存证的可信化存储方法的步骤。

10.一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中任一项所述数据存证的可信化存储方法的步骤。

一种数据存证的可信化存储方法、装置、设备和存储介质

技术领域

[0001] 本发明涉及数据加密技术领域，尤其涉及一种数据存证的可信化存储方法、装置、计算机设备以及存储介质。

背景技术

[0002] 当前随着音频视频数据越发广泛地用于司法领域中，其重要性也日渐增强，例如监控录像、执法记录仪等音频视频采集设备所固定的证据，往往在案情的确定中，起到非常重要的作用。

[0003] 但是，音频视频数据是非常容易进行编辑的，因此不法分子往往会篡改音频视频数据，或者通过黑客手段入侵存储有音频视频证据的服务器并将其直接删除，以达到自身的不法目的。因此需要一种防止音频视频证据被篡改，并且能够分布式存储音频视频证据的方法。

发明内容

[0004] 本发明提供了一种数据存证的可信化存储方法、装置、计算机设备以及存储介质，用于防止音频视频证据被篡改，并且能够分布式存储音频视频证据以防止证据灭失。

[0005] 在本发明的第一方面，提供了一种数据存证的可信化存储方法，包括以下步骤：

[0006] 获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块；

[0007] 通过散列函数计算每一个所述数据存证数据块的第一哈希值；

[0008] 将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值；

[0009] 通过所述散列函数计算每两个所述第一哈希值的第二哈希值；

[0010] 将所述第二哈希值作为所述叶子节点的双亲节点的值；

[0011] 将各个所述叶子节点以及所述双亲节点进行递归处理，构建完全二叉树；

[0012] 将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

[0013] 在本发明第一方面的一种可能的实现方式中，在所述通过散列函数计算每一个所述数据存证数据块的第一哈希值的步骤中，包括：

[0014] 对所述散列函数进行常量初始化，得到8个32位的哈希初值和4个32位的哈希常量；

[0015] 将所述数据存证数据块转换成二进制形式以得到二进制数据块，并对所述二进制数据块的比特位进行填充，得到目标消息；

[0016] 根据所述8个32位的哈希初值和4个32位的哈希常量和预设的逻辑函数对所述目标消息进行计算，得到所述第一哈希值。

[0017] 在本发明第一方面的一种可能的实现方式中，在所述将所述数据存证数据块转换成二进制形式以得到二进制数据块，并对所述二进制数据块的比特位进行填充，得到目标消息的步骤中，包括：

[0018] 获取所述二进制数据块的长度；

[0019] 在所述二进制数据块的末尾先补第一个比特位1；

[0020] 在补1后的所述二进制数据块的末尾补比特位0，直到长度满足对256取模后余数是192；

[0021] 在所述二进制数据块的末尾添加一个64位的比特串，得到所述目标消息，所述比特串是以二进制形式表示的所述二进制数据块的长度。

[0022] 在本发明第一方面的一种可能的实现方式中，若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同，所述可信化存储方法还包括：

[0023] 遍历所述双亲节点，定位到所述第二哈希值发生变化的所述双亲节点；

[0024] 遍历所述第二哈希值发生变化的所述双亲节点中的所述叶子节点，定位所述第一哈希值发生变化的所述叶子节点；

[0025] 定位所述第一哈希值发生变化的所述叶子节点对应的所述数据存证数据块。

[0026] 在本发明第一方面的一种可能的实现方式中，所述可信化存储方法还包括：

[0027] 将所述数据存证数据块根据所述完全二叉树保存在点对点网络中；

[0028] 若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同，则根据所述完全二叉树从所述点对点网络中提取被篡改的所述数据存证数据块。

[0029] 在本发明第一方面的一种可能的实现方式中，所述可信化存储方法还包括：

[0030] 当需要播放所述数据存证时，遍历所有的所述叶子节点；

[0031] 依次获取所有的所述叶子节点所对应的所述数据存证数据块。

[0032] 在本发明第一方面的一种可能的实现方式中，在所述获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块的步骤中，若所述数据存证数据块的数量为奇数时，复制最后一个所述数据存证数据块的副本，使得所述数据存证数据块的数量为偶数。

[0033] 本发明的第二方面提供了一种数据存证的可信化存储装置，包括以下模块：

[0034] 数据存证分割模块，用于获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块；

[0035] 第一哈希值计算模块，用于通过散列函数计算每一个所述数据存证数据块的第一哈希值；

[0036] 叶子节点构建模块，用于将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值；

[0037] 第二哈希值计算模块，用于通过所述散列函数计算每两个所述第一哈希值的第二哈希值；

[0038] 双亲节点构建模块，用于将所述第二哈希值作为所述叶子节点的双亲节点的值；

[0039] 完全二叉树构建模块，用于将各个所述叶子节点以及所述双亲节点进行递归处理，构建完全二叉树；

[0040] 根哈希值生成模块，用于将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

[0041] 本发明的第三方面提供了一种计算机设备，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上述任一数据存证的可信化存储方法的步骤。

[0042] 本发明的第四方面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上述任一数据存证的可信化存储方法的步骤。

[0043] 本发明的有益技术效果包括：将存证文件分割为多个分片，并利用散列函数计算出各个分片的哈希值，然后将各个分片的哈希值作为叶子节点并且最终构建出完全二叉树，每一个分片内容的修改都会导致根节点哈希值的改变，达到文件无法被篡改的目的。此外，由于存证文件分割为多个分片，能够将其在点对点网络中进行分布式存储，可以避免由于单一存储服务器被入侵而造成的存证文件被篡改或者删除，并且通过将存证文件的各个分片存储在点对点网络，能够达到并行的认证和播放，不需要完整的下载。

附图说明

[0044] 为了更清楚地说明本发明实施例的技术方案，下面将对本发明实施例的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发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动性的前提下，还可以根据这些附图获得其他的附图。

[0045] 图1是本发明一个实施例中数据存证的可信化存储方法的应用环境示意图；

[0046] 图2是本发明一个实施例中数据存证的可信化存储方法的流程图；

[0047] 图3是本发明另一个实施例中数据存证的可信化存储方法的流程图；

[0048] 图4是本发明一个实施例中可信化存储方法所构建的二叉树结构图；

[0049] 图5是本发明一个实施例中数据存证的可信化存储装置的结构示意图；

[0050] 图6是本发明一个实施例中计算机设备的示意图。

具体实施方式

[0051] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有作出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0052] 本申请提供的数据存证的可信化存储方法，可应用在如图1的应用环境中，其中，计算机设备或终端设备通过网络与服务器进行通信。其中，计算机设备或终端设备可以但不限于各种个人计算机、笔记本电脑、智能手机、平板电脑和便携式可穿戴设备。服务器可以用独立的服务器或者是多个服务器组成的服务器集群来实现。

[0053] 在一实施例中，如图2所示，提供一种数据存证的可信化存储方法，以该方法应用在图1中的服务器为例进行说明，包括如下步骤：

[0054] S1：获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块。

[0055] 为了构建完全二叉树，首先将数据存证分割成多个数据块，每个数据块对应完全二叉树的一个叶子节点，后述会详细说明。具体地，上述数据存证顺序地分割为多个预设大小的数据存证数据块，即数据存证数据块是从头到尾的顺序，无需另外进行调整。数据存证数据块的大小可以根据视频的大小以及系统的处理能力，尤其是后述的通过散列函数计算

各个数据存证数据块哈希值的能力，优选地，将需要存证的数据进行IO读取操作并按照128KB的大小进行分割，不足128KB部分按单个切片处理，得到多个数据存证数据块。

[0056] 在一个具体的实施例中，若所述数据存证数据块的数量为奇数时，复制最后一个所述数据存证数据块的副本，使得所述数据存证数据块的数量为偶数。

[0057] 具体地，为了提高数据的一致性，若分割数据存证后得到数据存证数据块的数量为奇数时，复制最后一个数据存证数据块的副本，使得数据存证数据块的数量为偶数。在后续计算双亲节点的第二哈希值时，可以保证哈希值的一致性，即都是由两个数据存证数据块的第一哈希值所计算二得到的。

[0058] S2：通过散列函数计算每一个所述数据存证数据块的第一哈希值。

[0059] 具体地，如图3所示，步骤S2还包括：

[0060] S201：对所述散列函数进行常量初始化，得到8个32位的哈希初值和4个32位的哈希常量；进一步地，包括：

[0061] 1)初始化8个32比特的哈希初值L0＝0xf7e1499，L1＝0x1a5c3162，L2＝0x2a5ff421，L3＝0x2a515f8a，L4＝0x366a396e，L5＝0x8580a649，L6＝

[0062] 0x220d1062，L7＝0x27d6fff0。

[0063] 2)初始化4个32比特的常量：C1:0x27292b63，C2:0x28fb26f3，C3:0x2c8d5346，C4:0x2e4de275。

[0064] S202：将所述数据存证数据块转换成二进制形式以得到二进制数据块，并对所述二进制数据块的比特位进行填充，得到目标消息；

[0065] 具体地，包括：

[0066] S2021：获取所述二进制数据块的长度；

[0067] S2022：在所述二进制数据块的末尾先补第一个比特位1；

[0068] S2023：在补1后的所述二进制数据块的末尾补比特位0，直到长度满足对256取模后余数是192；

[0069] S2024：在所述二进制数据块的末尾添加一个64位的比特串，得到所述目标消息，所述比特串是以二进制形式表示的所述二进制数据块的长度。需要指出的是，若上述以二进制形式表示的二进制数据块的长度不足64位时，在该比特串的前方开始补比特位0，直至该比特串的长度达到64位。

[0070] 然后，S203：所述8个32位的哈希初值和4个32位的哈希常量和预设的逻辑函数对所述目标消息进行计算，得到所述第一哈希值。具体包括以下步骤：

[0071] 1)定义以下逻辑函数：

[0072] 1.常量函数T_j接受输入变量j，当j大于等于和0小于等于15时，T_j等于常量C1，当j大于等于16小于等于31时，T_j等于常量C2，当j大于等于33小于等于48时，T_j等于常量C3，当j大于等于49小于等于64时，T_j等于常量C4。

[0073] 2.Fa(x,y,z)接受三个输入变量x,y,z均为32比特，先对x和y进行”与“运算，再对x和z进行”与“运算，再对两个进过进行“异或”运算得到32为的比特值后依次与x,y,z进行异或在得到2个与运算的32比特值后对这三个值进行依次异或运算得到32比特的结果。

[0074] 3.Ga(x,y,z)接受三个输入变量x,y,z均为32比特，先对x和y进行“与”运算，再对x进行求补码后和z进行”与“运算，然后对上述两个结果进行“异或”运算得到32比特值后依

次对x,y,z进行“异或”操作得到32比特结果。

[0075] 4.P_0(x)接受输入变量x为32比特,先将x与将x循环右移7位的结果进行“异或”操作,再将x循环右移19位,然后将这两个结果进行异或操作,得到32比特的结果。

[0076] 5.P_1(x)接受输入变量x为32比特,将x与x循环右移9位的结果进行“异或”操作,然后将x循环右移27位,将这两个结果进行“异或”操作,得到32比特的结果。

[0077] 2)按256比特进行分组,分组长度为n。

[0078] 构造64个字,每个字为16比特,对于W_0,W_1.....W_{15}可由每个分组的长度通过16比特划分得到,而W_{16},W_{17}.....W_{63}由以下公式得到:$W_t=P_1(W_{t-7} \oplus (W_{t-3} <<< 11)) \oplus P_0(W_{t-13}) \oplus W_{t-15}$,其中⊕为异或操作,<<<为循环左移操作。

[0079] 3)初始化A,B,C,D,E,F,G,H,将步骤1定义的8个哈希常量进行依次赋值,定义QQ1,WW2,EE1,YY2为中间变量,进行64次迭代,按照以下公式对A,B,C,D,E,F,G,H进行运算(其j为迭代次数-1,¬为求补码)。

[0080] PP1=((A<<<7))+E+(T_j<<<3)

[0081] $KK2 = SS1 \oplus (B <<< 13)$

[0082] $EE1 = Fa(A,B,C) + E + PP1 + (W_j \neg W_j)$

[0083] $YY2 = Ga(F,G,H) + E + KK2 + (W_j \neg W_j)$

[0084] D=C

[0085] C=B<<<9

[0086] B=A

[0087] A=KK2

[0088] G=H

[0089] H=E

[0090] F=H

[0091] $E = P_0(KK2) \oplus P_1(PP1)$

[0092] 4)根据分组长度n,进行n次迭代,根据步骤(6)与步骤(7)的算法对A,B,C,D,E,F,G,H进行迭代更新,最终将8个值进行拼凑得到最终256比特的哈希值,即数据存证数据块的第一哈希值。

[0093] S3:将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值。

[0094] 具体地,如图4所示,首先将各个数据存证数据块D_i的第一哈希值N_i作为叶子节点的值,为后续步骤做准备。

[0095] S4:通过所述散列函数计算每两个所述第一哈希值的第二哈希值。

[0096] 具体地,将每两个步骤S3中所得的第一哈希值N_i,通过步骤S2中所提供的散列函数,再次计算哈希值,即第二哈希值。详细地,如图4所示,即是将两个叶子节点的哈希值合并,然后再通过散列函数计算第二哈希值。

[0097] S5:将所述第二哈希值作为所述叶子节点的双亲节点的值。

[0098] 具体地,将所得到的第二哈希值作为叶子节点的双亲节点的值,其结构如图4所示,为后续步骤做准备。

[0099] S6:将各个所述叶子节点以及所述双亲节点进行递归处理,构建完全二叉树。

[0100] 具体地,将各个叶子节点以及双亲节点逐层递归,由下而上地构建完全二叉树,最终得到完全的完全二叉树。

[0101] S7:将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

[0102] 最终,如图4所示,生成的完全二叉树的根节点的第二哈希值即是数据存证的哈希值。

[0103] 在一个具体的实施例中,若某一数据存证通过执行上述方法后,得到的根节点的第二哈希值与原始文件的根节点的第二哈希值不相同,则说明该数据存证遭到篡改,为了定位具体被篡改的部分,所述可信化存储方法还包括:

[0104] S711:遍历所述双亲节点,定位到所述第二哈希值发生变化的所述双亲节点;

[0105] S712:遍历所述第二哈希值发生变化的所述双亲节点中的所述叶子节点,定位所述第一哈希值发生变化的所述叶子节点;

[0106] S713:定位所述第一哈希值发生变化的所述叶子节点对应的所述数据存证数据块。

[0107] 根据上述说明,完全二叉树的根节点的第二哈希值即为正确的数据存证的哈希值,若检测到数据存证的哈希值与上述第二哈希值不相同,则说明该数据存证遭到篡改,通过本方法的一个实施例,能够从构建的完全二叉树中定位并被篡改的数据块。具体地,由于双亲节点所记录的第二哈希值都是由叶子节点的第一哈希值所生成的,因此首先遍历各个双亲节点,定位到第二哈希值发生变化的双亲节点,然后再遍历该双亲节点中的各个叶子节点,定位第一哈希值发生变化的叶子节点,最后定位到被篡改的数据存证数据块。

[0108] 在一个具体的实施例中,所述可信化存储方法还包括:

[0109] S721:将所述数据存证数据块根据所述完全二叉树保存在点对点网络中;

[0110] S722:若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同,则根据所述完全二叉树从所述点对点网络中提取被篡改的所述数据存证数据块。

[0111] 具体地,为了避免服务器被入侵进而导致数据存证灭失,在一个具体的实施例中,还可以将数据存证数据块根据完全二叉树保存在点对点网络中。点对点网络具有多个节点,每一个节点都存储了全部或者部分的数据存证,且在一些具体的实施例中,各个节点是加密的,因此黑客是难以入侵所有节点的,从而保证了数据的安全性。进一步地,若检测到数据存证的哈希值与完全二叉树根节点的第二哈希值不相同,则说明该数据存证遭到篡改,此时,可以根据完全二叉树从点对点网络的节点中提取被篡改的所述数据,结合上述实施例,可以定位到具体被篡改的数据存证数据块,然后仅从点对点网络的节点中提取被篡改的所述数据存证数据块即可。

[0112] 在一个具体的实施例中,所述可信化存储方法还包括:

[0113] S731:当需要播放所述数据存证时,遍历所有的所述叶子节点;

[0114] S732:依次获取所有的所述叶子节点所对应的所述数据存证数据块。

[0115] 具体地,如需要播放音频视频证据的数据存证,可以通过专用的程序,遍历完全二叉树中的所有叶子节点,并逐个获取数据块,然后进行播放,需要注意的是,上述方法不仅可以播放存储在本地的数据存证,也可以从上述实施例中提供的点对点网络中下载数据存证然后播放。进一步地,从上述实施例中提供的点对点网络中下载数据存证时,不需要将全

部的数据存证全部下载之后再播放，可以一边播放一边遍历叶子节点并获取对应的数据存证数据块，以提高播放效率。

[0116] 应理解，上述实施例中各步骤的序号的大小并不意味着执行顺序的先后，各过程的执行顺序应以其功能和内在逻辑确定，而不应对本发明实施例的实施过程构成任何限定。

[0117] 在一实施例中，提供一种数据存证的可信化存储装置，该数据存证的可信化存储装置与上述实施例中数据存证的可信化存储方法一一对应。如图5所示，该数据存证的可信化存储装置包括数据存证分割模块101、第一哈希值计算模块102、叶子节点构建模块103、第二哈希值计算模块104、双亲节点构建模块105、完全二叉树构建模块106以及根哈希值生成模块107。各功能模块详细说明如下：

[0118] 数据存证分割模块101，用于获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块；

[0119] 第一哈希值计算模块102，用于通过散列函数计算每一个所述数据存证数据块的第一哈希值；

[0120] 叶子节点构建模块103，用于将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值；

[0121] 第二哈希值计算模块104，用于通过所述散列函数计算每两个所述第一哈希值的第二哈希值；

[0122] 双亲节点构建模块105，用于将所述第二哈希值作为所述叶子节点的双亲节点的值；

[0123] 完全二叉树构建模块106，用于将各个所述叶子节点以及所述双亲节点进行递归处理，构建完全二叉树；

[0124] 根哈希值生成模块107，用于将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

[0125] 在一个具体的实施例中，在所述第一哈希值计算模块102中，具体包括：

[0126] 常量初始化单元，用于对所述散列函数进行常量初始化，得到8个32位的哈希初值和4个32位的哈希常量；

[0127] 比特位填充单元，用于将所述数据存证数据块转换成二进制形式以得到二进制数据块，并对所述二进制数据块的比特位进行填充，得到目标消息；

[0128] 哈希值计算单元，用于根据所述8个32位的哈希初值和4个32位的哈希常量和预设的逻辑函数对所述目标消息进行计算，得到所述第一哈希值。

[0129] 在一个具体的实施例中，比特位填充单元包括：

[0130] 二进制数据块长度获取单元，用于二进制数据块的长度；

[0131] 第一补位单元，用于在所述二进制数据块的末尾先补第一个比特位1；

[0132] 第二补位单元，用于在补1后的所述二进制数据块的末尾补比特位0，直到长度满足对256取模后余数是192；

[0133] 比特串添加单元，在所述二进制数据块的末尾添加一个64位的比特串，得到所述目标消息，所述比特串是以二进制形式表示的所述二进制数据块的长度。

[0134] 在一个具体的实施例中，所述可信化存储装置还包括：

[0135] 双亲节点模块，用于遍历所述双亲节点，定位到所述第二哈希值发生变化的所述双亲节点；

[0136] 叶子节点定位模块，用于遍历所述第二哈希值发生变化的所述双亲节点中的所述叶子节点，定位所述第一哈希值发生变化的所述叶子节点；

[0137] 数据存证数据块定位模块，用于定位所述第一哈希值发生变化的所述叶子节点对应的所述数据存证数据块。

[0138] 在一个具体的实施例中，所述可信化存储装置还包括：

[0139] 点对点网络存储模块，用于将所述数据存证数据块根据所述完全二叉树保存在点对点网络中；

[0140] 点对点网络提取模块，用于若所述数据存证的哈希值与所述根节点的所述第二哈希值不相同，则根据所述完全二叉树从所述点对点网络中提取被篡改的所述数据存证数据块。

[0141] 在一个具体的实施例中，所述可信化存储装置还包括：

[0142] 叶子节点遍历模块，用于当需要播放所述数据存证时，遍历所有的所述叶子节点；

[0143] 数据存证数据块获取模块，用于依次获取所有的所述叶子节点所对应的所述数据存证数据块。

[0144] 在一个具体的实施例中，数据存证分割模块101还包括：

[0145] 副本复制单元，用于若所述数据存证数据块的数量为奇数时，复制最后一个所述数据存证数据块的副本，使得所述数据存证数据块的数量为偶数。

[0146] 其中上述模块/单元中的“第一”和“第二”的意义仅在于将不同的模块/单元加以区分，并不用于限定哪个模块/单元的优先级更高或者其它的限定意义。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或模块的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或模块，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或模块，本申请中所出现的模块的划分，仅仅是一种逻辑上的划分，实际应用中实现时可以有另外的划分方式。

[0147] 关于数据存证的可信化存储装置的具体限定可以参见上文中对于数据存证的可信化存储方法的限定，在此不再赘述。上述数据存证的可信化存储装置中的各个模块可全部或部分通过软件、硬件及其组合来实现。上述各模块可以硬件形式内嵌于或独立于计算机设备中的处理器中，也可以以软件形式存储于计算机设备中的存储器中，以便于处理器调用执行以上各个模块对应的操作。

[0148] 在一个实施例中，提供了一种计算机设备，该计算机设备可以是服务器，其内部结构图可以如图6所示。该计算机设备包括通过系统总线连接的处理器、存储器、网络接口和数据库。其中，该计算机设备的处理器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作系统、计算机程序和数据库。该内存储器为非易失性存储介质中的操作系统和计算机程序的运行提供环境。该计算机设备的数据库用于存储数据存证的可信化存储方法中涉及到的数据。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种数据存证的可信化存储方法。

[0149] 在一个实施例中，提供了一种计算机设备，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，处理器执行计算机程序时实现上述实施例中数据存证的可信化存储方法的步骤，例如图2所示的步骤S1至步骤S7及该方法的其它扩展和相关步骤的延伸。或者，处理器执行计算机程序时实现上述实施例中数据存证的可信化存储装置的各模块/单元的功能，例如图5所示模块101至模块107的功能。为避免重复，这里不再赘述。

[0150] 所述处理器可以是中央处理单元(Central Processing Unit，CPU)，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，所述处理器是所述计算机装置的控制中心，利用各种接口和线路连接整个计算机装置的各个部分。

[0151] 所述存储器可用于存储所述计算机程序和/或模块，所述处理器通过运行或执行存储在所述存储器内的计算机程序和/或模块，以及调用存储在存储器内的数据，实现所述计算机装置的各种功能。所述存储器可主要包括存储程序区和存储数据区，其中，存储程序区可存储操作系统、至少一个功能所需的应用程序(比如声音播放功能、图像播放功能等)等；存储数据区可存储根据手机的使用所创建的数据(比如音频数据、视频数据等)等。

[0152] 所述存储器可以集成在所述处理器中，也可以与所述处理器分开设置。

[0153] 在一个实施例中，提供了一种计算机可读存储介质，其上存储有计算机程序，计算机程序被处理器执行时实现上述实施例中数据存证的可信化存储方法的步骤，例如图2所示的步骤S1至步骤S7及该方法的其它扩展和相关步骤的延伸。或者，计算机程序被处理器执行时实现上述实施例中数据存证的可信化存储装置的各模块/单元的功能，例如图5所示模块101至模块107的功能。为避免重复，这里不再赘述。

[0154] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程，是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一非易失性计算机可读取存储介质中，该计算机程序在执行时，可包括如上述各方法的实施例的流程。其中，本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限，RAM以多种形式可得，诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0155] 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。

[0156] 以上所述实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，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发明各实施例技术方案的精神和范围，均应包含在本发明的保护范围之内。

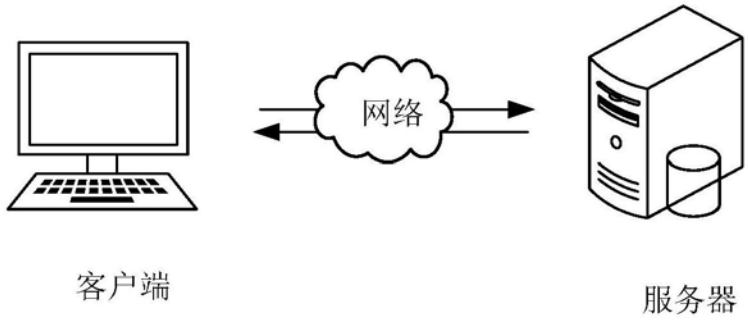


图1

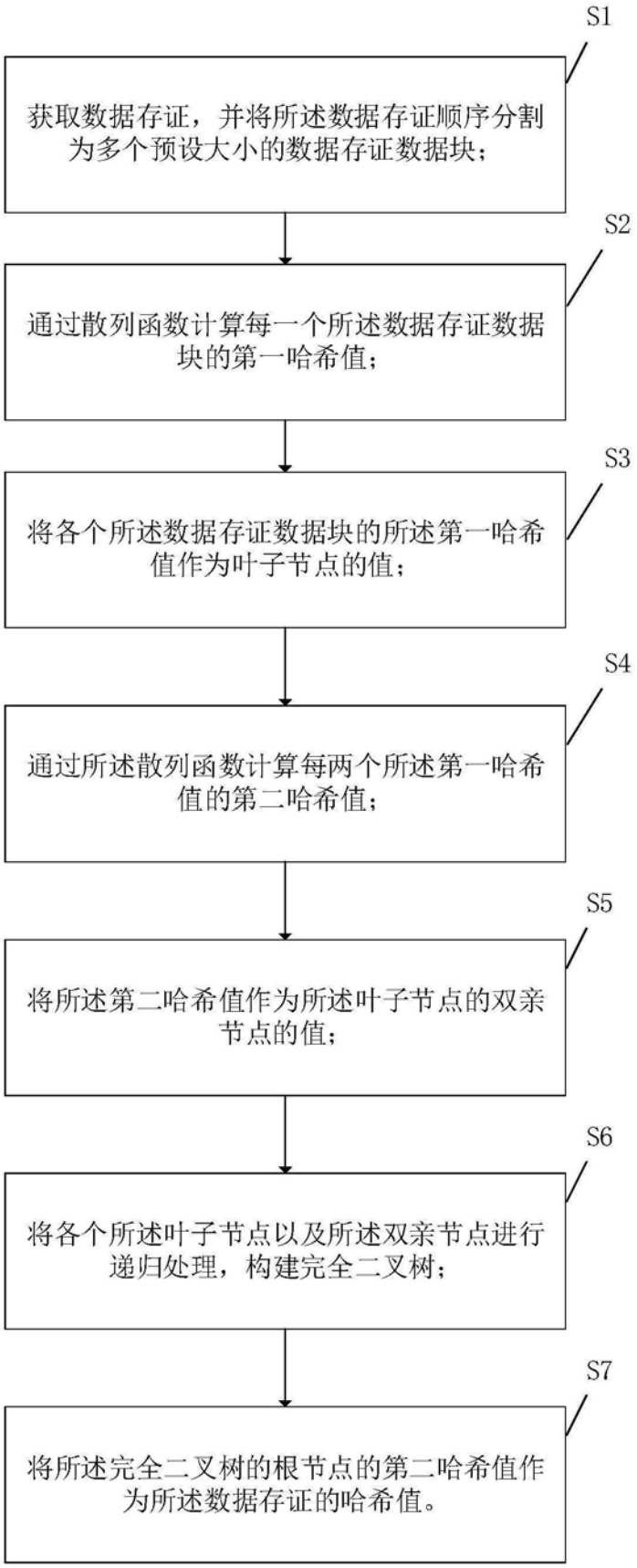

S1
获取数据存证，并将所述数据存证顺序分割为多个预设大小的数据存证数据块；
S2
通过散列函数计算每一个所述数据存证数据块的第一哈希值；
S3
将各个所述数据存证数据块的所述第一哈希值作为叶子节点的值；
S4
通过所述散列函数计算每两个所述第一哈希值的第二哈希值；
S5
将所述第二哈希值作为所述叶子节点的双亲节点的值；
S6
将各个所述叶子节点以及所述双亲节点进行递归处理，构建完全二叉树；
S7
将所述完全二叉树的根节点的第二哈希值作为所述数据存证的哈希值。

图2

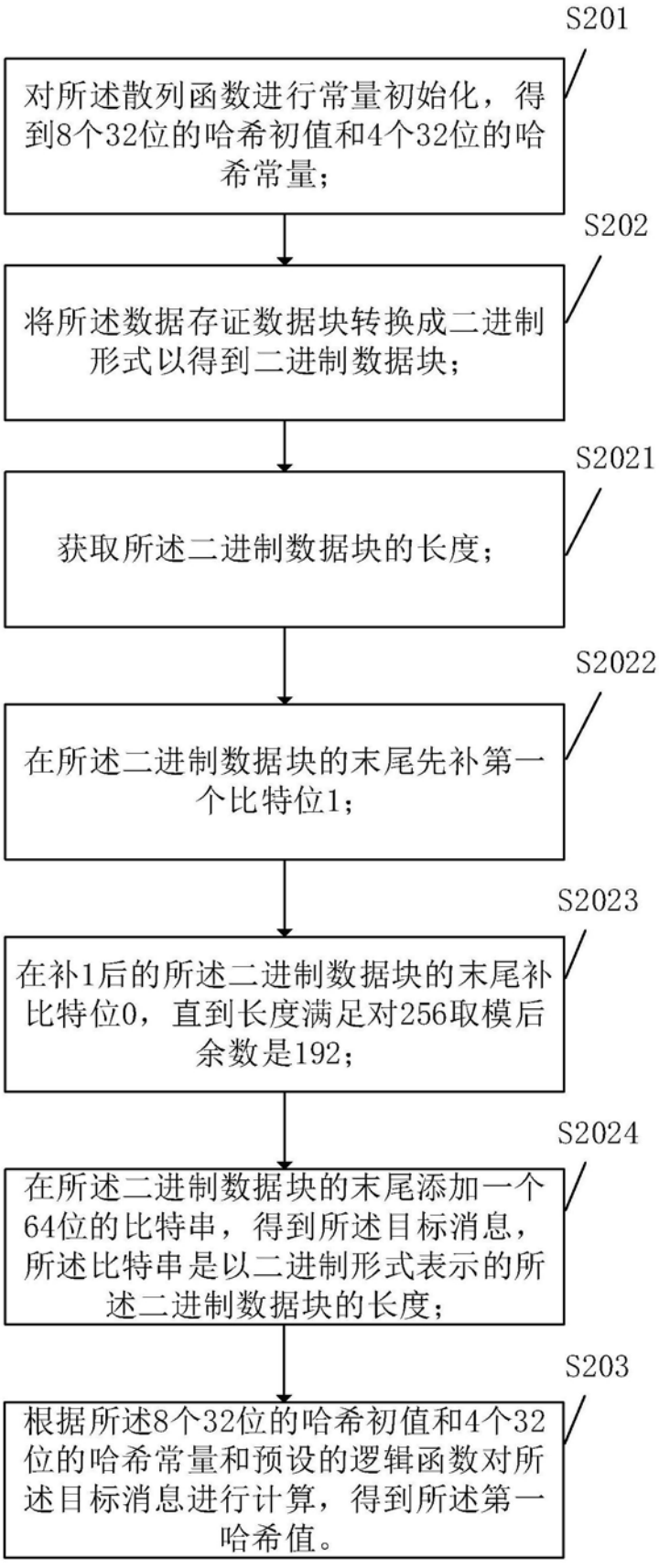

S201
对所述散列函数进行常量初始化，得到8个32位的哈希初值和4个32位的哈希常量；
S202
将所述数据存证数据块转换成二进制形式以得到二进制数据块；
S2021
获取所述二进制数据块的长度；
S2022
在所述二进制数据块的末尾先补第一个比特位1；
S2023
在补1后的所述二进制数据块的末尾补比特位0，直到长度满足对256取模后余数是192；
S2024
在所述二进制数据块的末尾添加一个64位的比特串，得到所述目标消息，所述比特串是以二进制形式表示的所述二进制数据块的长度；
S203
根据所述8个32位的哈希初值和4个32位的哈希常量和预设的逻辑函数对所述目标消息进行计算，得到所述第一哈希值。

图3

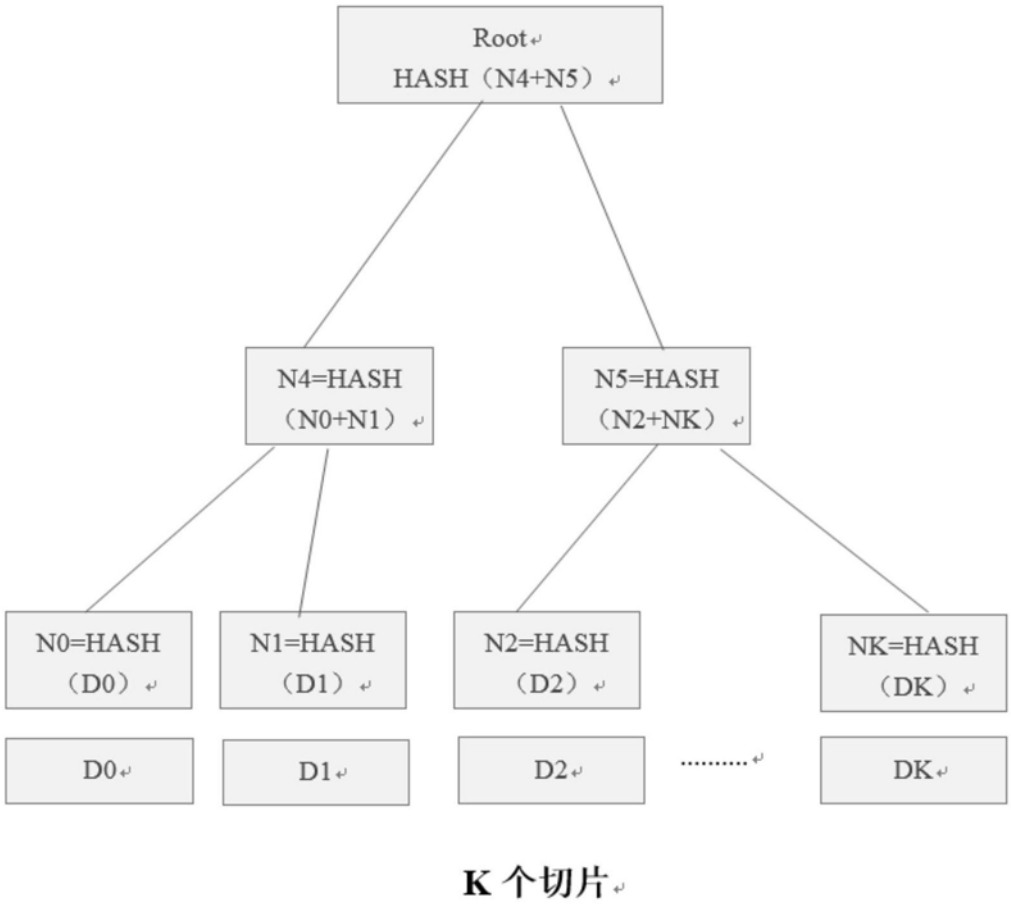

Root
HASH（N4+N5）
N4=HASH
（N0+N1）
N5=HASH
（N2+NK）
N0=HASH
（D0）
N1=HASH
（D1）
N2=HASH
（D2）
NK=HASH
（DK）
D0
D1
D2
.....
DK
K 个切片

图4

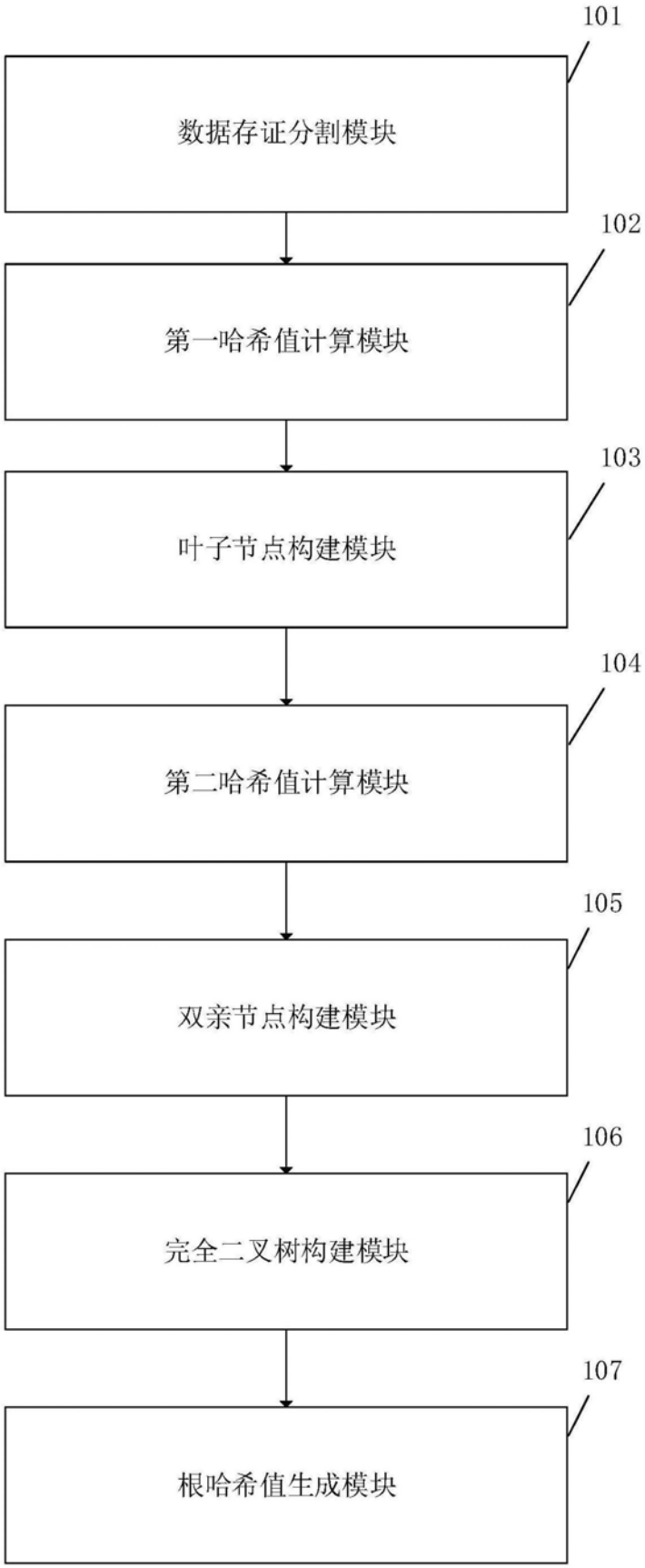

101
数据存证分割模块
102
第一哈希值计算模块
103
叶子节点构建模块
104
第二哈希值计算模块
105
双亲节点构建模块
106
完全二叉树构建模块
107
根哈希值生成模块

图5

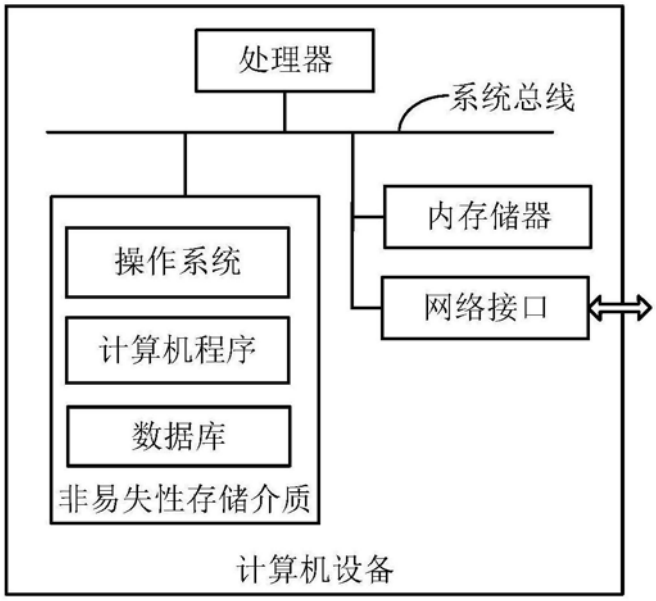


图6