(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110210254 B
(45)授权公告日 2023.06.02

(21)申请号 201910509326.X

(22)申请日 2019.06.13

(65)同一申请的已公布的文献号
申请公布号 CN 110210254 A

(43)申请公布日 2019.09.06

(73)专利权人 东华大学
地址 201600 上海市松江区人民北路2999号

(72)发明人 徐光伟 赖淼麟 史春红 韩松桦

(74)专利代理机构 上海申汇专利代理有限公司 31001
专利代理师 翁若莹 王文颖

(51)Int.Cl.
G06F 21/64(2013.01)
G06F 21/60(2013.01)

(56)对比文件
CN 104994069 A,2015.10.21
CN 104598569 A,2015.05.06
CN 106650503 A,2017.05.10
CN 109286490 A,2019.01.29
WO 2014191057 A1,2014.12.04
US 2010114665 A1,2010.05.06
AU 2013207274 A1,2014.08.21
CA 2479343 A1,2003.10.02

审查员 王轩

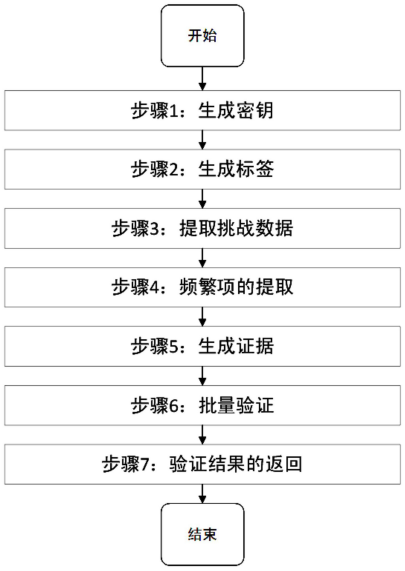
权利要求书1页 说明书4页 附图2页

(54)发明名称

一种多数据完整性验证中重复数据的优化验证方法

(57)摘要

本发明公开一种多数据完整性验证中重复数据的优化验证方法，其特征在于，首先对挑战集合计算频繁项集，在计算数据证据以及标签证据之前，先找出挑战集合重叠的部分，进行批处理，减少证据计算时的计算量；接着处理挑战集合中非重复的部分，计算这部分数据的数据证据以及标签证据；最后，将重叠部分和非重叠部分的数据合并，将证据发送给第三方验证者。本发明通过多挑战时云服务器在计算证据时的计算方法，减少了证据计算时的计算量，既可以保证验证者能够获得正确的数据完整性的验证结果，又可以减少云存储提供商因重复计算相同验证数据所造成的验证开销浪费。本发明提高验证效率的同时，保证了验证的安全性和正确性。



CN 110210254 B

1.一种重复挑战数据取出的完整性验证方法，其特征在于，首先对挑战集合计算频繁项集，在计算数据证据以及标签证据之前，先找出挑战集合重复的部分，进行批处理，减少证据计算时的计算量；接着处理挑战集合中非重复的部分，计算这部分数据的数据证据以及标签证据；最后，将重复部分和非重复部分的数据合并，将证据发送给第三方验证者；具体步骤如下：

步骤1：生成链密钥，为了确保不同版本文件之间的关联，在密钥的生成中应用链密钥；在基本密钥的基础上，通过其前一版本文件的散列密钥来计算每个版本文件的处理密钥；

步骤2：生成数据标签，用户为加密文件中的每个数据块生成一个标签，标签生成方法为由n个数据块即$m_1,\cdots,m_n$组成的文件M中的每个数据块m_i生成标签t_i，最后输出一组数据标签$T=\{t_i\},i\in[1,n]$；

步骤3：批量挑战，由验证者执行，从域Z_p中选择一个随机数生成挑战C；

步骤4：频繁项的提取，通过对所有的挑战集合计算频繁项集，找出这些集合具有的相同项，计算这些频繁项的数据证据以及标签证据；

步骤5：生成证据，对原始挑战集合，计算非频繁项的数据证据以及标签证据；最后，将频繁项的证据和非频繁项的证据进行合并，计算验证者TPA_k发送过来的挑战集合的证据P_i，其中包括标签证据TP_i和数据证据DP_i；

步骤6：批量验证，第三方验证者根据从步骤5中获得的TP_i和数据证据DP_i以及数据块的哈希值来验证存储在云上的数据的完整性；

步骤7：验证结果返回，云服务器将自己验证后的结果通过安全通道反馈给验证者。

2.如权利要求1所述的重复挑战数据取出的完整性验证方法，其特征在于，为了保护数据的隐私性，在所述第三方验证者进行数据验证时，还应用双线性映射和同态技术来保证验证的安全性和验证结果的可靠性，同时降低网络通信的流量成本。

一种多数据完整性验证中重复数据的优化验证方法

技术领域

[0001] 本发明涉及一种多数据完整性验证中重复数据的优化验证方法，属于云计算、信息安全技术领域，适用于云存储。

背景技术

[0002] 随着云计算的快速发展，云存储作为新一代计算基础设施得到越来越多的关注。与此同时，越来越多的云存储服务涌现出来，为用户提供低成本且庞大的数据存储空间。尽管云存储可以随时提供便捷的存储和快速的数据访问等，但是当用户将拥有的数据上传到云服务器后，便失去了数据的绝对控制，所存储数据的完整性和安全性问题无法得到有效的保证。且存在云服务提供商为了节省存储空间，对未被访问或访问频率较少的用户数据执行删除操作，并对用户反馈假的数据完整性验证结果。为避免云存储中数据的损失，需要使用户在有限的计算能力下确保大规模数据存储的完整性。

[0003] 现有技术中为了解决上述问题，提出了数据完整性验证的方法。但是现有云存储中完整性验证方法都是只针对单一验证者对多个文件提出完整性验证请求，并没有考虑多个验证者对多个文件提出完整性验证的情况。当多个用户对多个数据文件提出完整性验证请求时，很可能会对相同的数据文件进行验证。对于热门文件、计算机程序以及其他信息会有多个用户对同一个文件提出完整性验证的情况。因此这导致了多个验证者会对同一文件提出完整性验证，但是云服务器并不能对同一挑战请求进行批处理，造成额外的开销。因此，一种高效的云存储数据完整性验证方法是亟待解决的问题。

发明内容

[0004] 本发明所要解决的技术问题是：如何提高云服务器在计算重复挑战数据效率，在保护用户隐私的同时验证远程存储版本数据的完整性。具体地说，当存在多个验证者对同一个数据块提出完整性验证请求时，如何减少云服务器在计算数据证据以及标签证据时的计算量的同时验证远程存储版本数据的完整性。

[0005] 为了解决上述问题，本发明的技术方案是提供了一种重复挑战数据取出的完整性验证方法，其特征在于，首先对挑战集合计算频繁项集，在计算数据证据以及标签证据之前，先找出挑战集合重叠的部分，进行批处理，减少证据计算时的计算量；接着处理挑战集合中非重复的部分，计算这部分数据的数据证据以及标签证据；最后，将重叠部分和非重叠部分的数据合并，将证据发送给第三方验证者。

[0006] 优选地，为了保护数据的隐私性，在所述第三方验证者进行数据验证时，还应用双线性映射和同态技术来保证验证的安全性和验证结果的可靠性，同时降低网络通信的流量成本。

[0007] 优选地，具体步骤如下：

[0008] 步骤1：生成链密钥，为了确保不同版本文件之间的关联，在密钥的生成中应用链密钥；在基本密钥的基础上，通过其前一版本文件的散列密钥来计算每个版本文件的处理

密钥；

[0009] 步骤2：生成数据标签，用户为加密文件中的每个数据块生成一个标签，最后输出一组数据标签$T=\{t_i\}, i\in[1,n]$；

[0010] 步骤3：批量挑战，由验证者执行，从域Z_p中选择一个随机数生成挑战C；

[0011] 步骤4：频繁项的提取，通过对所有的挑战集合计算频繁项集，找出这些集合具有的相同项，计算这些频繁项的数据证据以及标签证据；

[0012] 步骤5：生成证据，对原始挑战集合，计算非频繁项的数据证据以及标签证据；最后，将频繁项的证据和非频繁项的证据进行合并，计算验证者TPA_k发送过来的挑战集合的证据P_i，其中包括标签证据TP_i和数据证据DP_i；

[0013] 步骤6：批量验证，第三方验证者根据从步骤5中获得的TP_i和数据证据DP_i以及数据块的哈希值来验证存储在云上的数据的完整性；

[0014] 步骤7：验证结果返回，云服务器将自己验证后的结果通过安全通道反馈给验证者。

[0015] 与现有技术相比，本发明的有益效果在于：

[0016] 1、本发明使用频繁项集，对不同验证者发送过来的挑战集合计算重复项，改进了现有方法逐个计算证据的方法。它可以通过对多个验证者发送过来的挑战集合计算频繁项集，提取多个验证任务中的相同数据对象，使得云服务器不需要重复计算不同验证者对相同数据对象提出的验证任务。这样，在验证成本有限的情况下，本发明提高了证据计算时的效率，与此同时能够有效地验证数据完整性；

[0017] 2、本发明将频繁项集和完整性验证结合，设计了一种对多个挑战集合选取频繁项集的方法，即根据提取多个挑战集合中的相同的部分，进行批处理；

[0018] 3、本发明改进了证据生成方法，在验证中，通过提取不同挑战集合中的频繁项，并进行批处理后，将提取出来的频繁项合并到原始挑战集合中，减少了云服务器在计算相同数据对象的计算开销。

[0019] 4、本方法改进了云服务器在证据生成的方法。对于到达云服务器的任务集合，提取该任务集中的相同的数据对象。云服务器将优先计算所提取出来的频繁项的证据，再计算剩余数据的证据。最后将频繁项的证据和非频繁项的证据合并。这样，在验证成本有限的情况下，本方法提高了云服务器在多验证者时计算证据时的效率，有效地保护了数据的完整性。

[0020] 本发明通过多挑战时云服务器在计算证据时的计算方法，减少了证据计算时的计算量。本发明提高验证效率的同时，保证了验证的安全性和正确性。

附图说明

[0021] 图1为实施例提供的重复挑战数据去除的完整性验证的整体流程；

[0022] 图2为集合拆分和合并的过程。

具体实施方式

[0023] 为使本发明更明显易懂，兹以优选实施例，并配合附图作详细说明如下。

[0024] 实施例

[0025] 在本实施例中设G_1和G_t为具有素数p的乘法群，并且$e:G_1\rightarrow G_t$为双线性映射。令g_1和g_2分别为G_1和G_t的生成元。

[0026] 以下内容具体说明本发明提供的一种重复挑战数据去除的完整性验证方法：

[0027] 步骤1：数据所有者随机选择一个私钥sk，并计算一个公钥$pk=g^{sk}$。

[0028] 步骤2：生成数据标签方法TagGen(M,sk)→T，设M为外包数据集。标签生成方法为由n个数据块即$m_1,\cdots,m_n$组成的文件M中的每个数据块m_i生成标签t_i，其中$i\in[1,n]$。首先为每个文件选择随机值$x_i\in Zp$。对于每个数据块m_i计算其数据标签t_i为：

[0029] $$t_i=\left(h(m_{i\cdot id})\times g^{am_i}\right)^{sk}$$

[0030] 其中，$m_{i\cdot id}$是数据块m_i的标识，a为选择的随机数。它输出一组数据标签$T=\{t_i\}i\in[1,n]$。

[0031] 步骤3：提取挑战数据。验证者TPA_k选取数据文件M中的$c\leqslant n$个数据块发起挑战，产生c个索引号，组成索引集合Q_i，并为每个待验证的数据块索引j_i在Z_p中任意选取一个随机数v_{ji}与之对应，即产生二元组(j_i,v_{ji})。

[0032] 步骤4：频繁项的提取，对于步骤3中得到的任务集，云服务器将通过FP-Growth算法提取任务集中多个任务的频繁项集，集合的拆分和合并的过程如图2所示。同时，保存数据频繁项集的数据对象所对应的每一个随机数。它首先计算所有受挑战数据块的线性组合$MP_{F_i}=\sum_{j\in S_i}m_j\cdot\sum_{v_i\in V_j}v_i$，计算所有属于同一个挑战索引的随机数的平均值$\overline{v}=\sum_{v_i\in V_{jF_i}}v_i/num_{f_i}$，然后计算频繁项集的数据证据$DP_{F_i}=e(u,pk)^{MP_{F_i}}$和标签证据$TP_{F_i}=\prod_{f_i\in F_i}\sigma_{f_i}^{\overline{v}}$。

[0033] 步骤5：生成证据，对原始挑战集合，计算非频繁项挑战数据块的线性组合$MP_{T_i'}=\sum_{i\in\Omega_i}m_iv_i$，接着计算数据证据$DP_{T_i'}=e(u,pk)^{MP_{T_i'}}$以及标签证据$TP_{T_i'}=\prod_{i\in\Omega_i}\sigma_i^{v_i}$。最后，将频繁项的证据和非频繁项的证据进行合并，则数据证据$DP_{T_i}=DP_{T_i'}\cdot DP_{F_i}$，标签证据为$TP_{T_i}=TP_{T_i'}\cdot TP_{F_i}$，得到验证者$TPA_k$发送过来的挑战集合的证据$P_i$，其中包括标签证据$TP_{T_i}$和数据证据$DP_{T_i}$。

[0034] 步骤6：批量验证，第三方验证者根据从步骤6中获得的标签证据和数据证据以及数据块的哈希值$h(m_{i\cdot id})$来验证存储在云上的数据的完整性。当完成所有挑战文件的计算时，通过的验证方程验证证明如下所示：

[0035] $$DP_{T_i}\cdot e\left(\prod_{i\in\Omega_i}H(m_{id})^{v_i}\cdot\prod_{f_i\in F'}H(m_{id})^{\left(v_{s_{ia}}+\cdots+v_{s_{ij}}\right)/num_{s_i}},pk\right)=e(TP_{T_i},g)\quad\text{式（1）。}$$

[0036] 如果式(1)为真，则输出1并且指示所有经验证的文件是完整的；否则，输出0，表示存在损坏的文件。

[0037] 利用上述分析方法，本发明所有的测试都在云存储平台和两台笔记本电脑上进行。由两台服务器组成的云存储平台，每台配备至E5-24031.8GHzCPU和32GBRAM作为云服务提供商，配备IntelCorei5-4210M2.60GHzCPU和4GB内存的两款笔记本电脑分别作为用户和第三方验证者使用。在实验过程中，为了减少实验时间，本方法将存储文件的大小设置为

40G,设数据块的大小固定为320byte,并设置挑战数据块数为50000块,验证者个数为10人,与此同时,每个验证者之间存在10%的挑战块是重复的。在证据计算阶段,云服务器计算数据证据以及标签证据的时间减少了19%。实验结果表明本发明减少云服务器在计算数据证据以及标签证据时的计算量。而且,由于同时减少了验证成本,因此减少了验证中的传输开销。

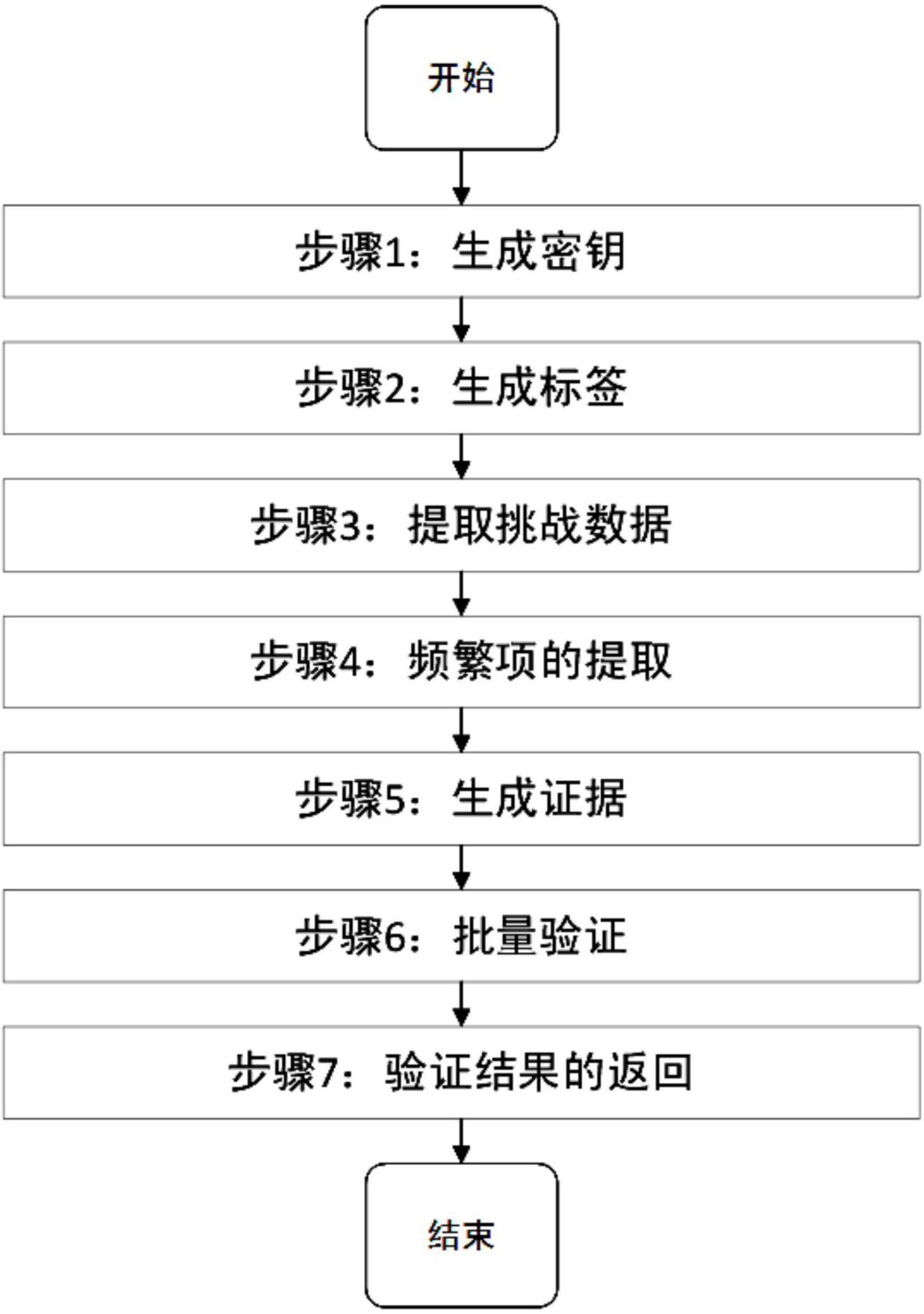

开始
步骤1：生成密钥
步骤2：生成标签
步骤3：提取挑战数据
步骤4：频繁项的提取
步骤5：生成证据
步骤6：批量验证
步骤7：验证结果的返回
结束

图1

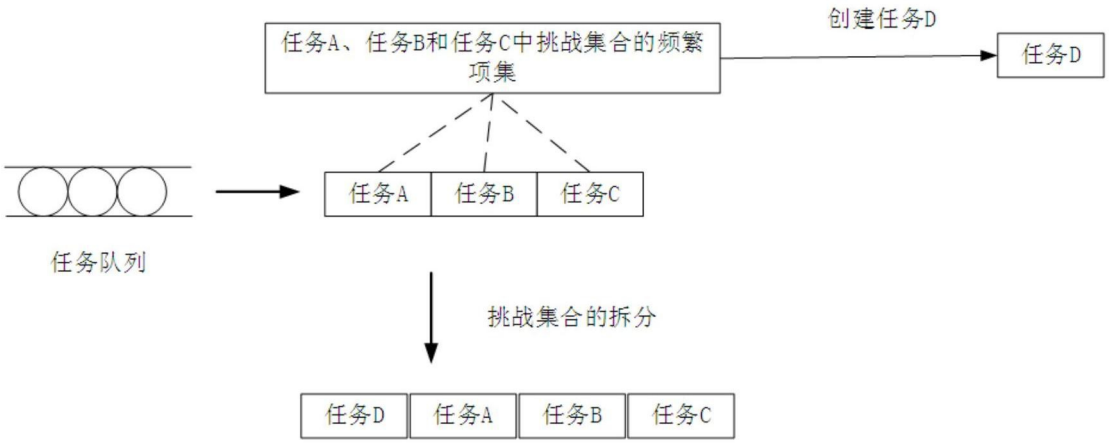

任务A、任务B和任务C中挑战集合的频繁项集
创建任务D
任务D
任务A
任务B
任务C
任务队列
挑战集合的拆分
任务D
任务A
任务B
任务C

图2