

DOI:10.12154/j.qbzlgz.2020.02.011

大数据环境下数据对象的可溯源性保障方法研究*

朝乐门 李昊璟 冀佳钰 (中国人民大学信息资源管理学院 北京 100872)

摘要: [目的/意义]开展数据对象可溯源性保障研究,可降低大数据时代所凸显的跨领域、海量、异构和动态数据的失信风险,有助于自动化实现数据质量评价、数据审计及数据恢复等问题。[方法/过程]基于数据溯源、密码编码学等理论,以数据对象为操作单位,提出一种新的数据对象版本号计算方法;进一步提出了数据对象DNA的概念及其生成和回溯方法,并探讨其IT实现及应用。[结果/结论]本文提出统一溯源新方法数据对象DNA,具有易生成、防篡改、普适性强等特点,可应用于数据对象的世系检验和数据连续性自动审计等场景。

关键词: 数据对象 数据溯源 可溯源性保障 数据连续性

An Analysis of Data Object Traceability Guarantee Method in Big Data Environment

Chao Lemen Li Haojing Ji Jiayu

(School of Information Resource Management, Renmin University of China, Beijing, 100872)

Abstract: [Purpose/significance] Data object traceability assurance research can reduce exposure risk of cross-domain, massive, heterogeneous and dynamic data loss in big data era, and contribute to automated data quality evaluation, data audit and data recovery. [Method/process] Based on the theory of data provenance and cryptography, a new version number calculation method of data objects is provided. Then, the concept of data object DNA are described, and a new method for generating as well as backtracking the data object DNA are also proposed. Furthermore, The IT implementation and applications of it are discussed. [Result/conclusion] Data object DNA, which is a new method to support unified data traceability, is producing easily, tamper-proof, strong universality, and can be used to data object lineage inspection and data continuity audit.

Keywords: data object data provenance assurance of traceability data continuity

1 引言

可溯源性是数据质量评价的关键要素之一^[1]。在数据连续性研究中,数据的可溯源性是数据质量在时间维度上的重要表现形式,数据可溯源性保障是降低数据的“失信”风险的主要手段^[2]。在大数据环境下,数据失信风险主要源自数据本身的多源、异构、海量、动态特征,因此对数据质量审计时需要重视数据源、数据

类型、数据整体的生命周期及应用领域等方面^[3]。目前,失信风险的凸显不仅对数据的可信度审计提出了更高要求,还对数据的有效利用带来了新挑战。数据的可溯源性已成为数据管理领域重点研究的新课题,这一课题的研究将有助于数据质量评价、数据审计、权属关系确认、数据恢复等领域的进步^[4]。

通常,数据溯源性保障主要采用数据溯源的方法和技术实现。目前,数据溯源方法和技术已广泛应用

*本文系国家自然科学基金项目“数据连续性的实现方法与保障机制研究”(项目编号:15BTQ054)的研究成果。

于多个领域,如科学数据流、云计算、医疗健康、地理信息学等。但是,现有研究所提出的溯源方法仅适用于各自的特定领域,并且所溯源对象大多为所在领域内的单一数据类型,如文本、XML、图像等。数据溯源性保障研究的主要局限体现在两个方面:一是跨领域应用能力差,相关方法可以指导类似领域相似数据的可溯源性的保障工作,但是无法直接推广应用到其他领域中;二是跨数据类型的应用能力差。多数方法的应用仅限于某一特定的已知数据类型,无法满足大数据时代海量、异构、动态数据的可溯源性保障的实际工作需要。

本文主体内容的组织方式如下:首先,梳理数据对象的相关研究,尤其是数据对象的版本号计算方法和数据溯源理论的研究进展;其次,探讨提出新的版本号计算方式、数据对象的DNA生成和校验方式及其在IT实现及应用,最后,总结本方法优势及应用价值,讨论下一步工作方向。

2 数据对象的相关研究

数据对象的内涵经历了从“特指”到“泛指”的过渡。起初,数据对象被认为是存储系统中的最基本元素^[5]。通常,在数据存储管理时,需要存储包括数据本身及其附带的元数据信息,二者结合称为数据对象。数据对象具有逻辑或物理意义上的独立性,可以被识别^[6]。数据对象中的元数据具有对数据进行描述、搜索、管理和回溯的功能^[7]。随着数据管理工作的深入,数据对象的外延不断扩大,数据对象的覆盖范围拓展至文件、图像、视频、音频等多种数据类型,数据对象的结构和类型呈现出多样化趋势^[8]。例如,在非图存储类NoSQL数据库中,数据以Key-Value/Document/Column形式存储^[9],即对所有数据对象分配一个Key(键),将数据内容的当作一个整体的存取单位进行处理。因此,在大数据时代,数据对象的概念不再是“特指”数据存取的最基本单元,而是“泛指”一切的数据存取单位,其粒度可大可小,不再要求必须为最小的组成要素。

在实际业务系统中,为了存取、传播和利用目的,通常将“数据”封装成“对象”,并以“数据对象”为单位进行操作和管理。例如,在数据的长久保存中,通常将数据封装成OAIS对象,进行统一格式化和长期安全保存^[10];在数据传输时,需要进行加密或压缩处理后封装成数据对象^[11];在数据传播时,通常对数据封装成网页或文件对象^[12];在数据共享时,数据往往封装成XML和JSON数据对象,对外提供访问接口^[13]。

在大数据环境下,一般数据应用系统所处理的数据对象并非为单一或者特定类型,需要将不同类型的数据对象进行统一处理。因此,本文所指的“数据对象”为在实际业务系统中的数据存取单位,不要求在结构或语义上的原子性,而强调的是数据的存储和获取中具体采用的粒度,其具体表现形式可以为文件夹、文件或从文件中抽取的片段数据。可见,本文中的数据对象具有更强的通用性,符合大数据环境下的数据类型更多、体量更大、处理难度高等新应用场景。

2.1 版本号的计算方法

目前数据对象版本计算方式尚未有统一的标准,基本以简单易识读为主要原则。版本号的计算方式是由版本控制所决定的,版本控制又称为修订控制,是对数据、文档、计算机程序以及其他信息集合的多个版本进行管理^[14],版本控制系统可以帮助回溯文件更改的历史演变关系^[15]。常用的有语义版本控制、日期版本控制等诸多版本生成方式。其中在软件领域中常用的版本号计算方式是语义版本控制,由major、minor、patch三部分组成^[16];有部分项目采用的是日期控制,在Wine项目使用的就是基于ISO 8601模式的日期生成版本号,Microsoft Office也会采用日期的方式编码版本,但通常是出于营销的目的^[17];在数据管理中,如MongoDB这样的NoSQL数据库中存在着特定的版本管理模块——Vermongo,其中文件等数据对象的版本是用简单数字累加的方式来计算^[18],如V1、V2等,版本记录仅针对数据对象中的异构数据本身;还有一些会采用特殊数字来对版本进行控制的软件或系统,如Knuth编写的排版系统TeX的版本号接近于 π ^[19]。这些版本号均仅从人类识读的角度出发,计算机难以理解和处理,因此本文提出一种新的版本号计算方式。

2.2 数据溯源理论

描述数据对象间的相关关系有两种理论——关联数据理论(Linked Data)和数据溯源理论(Data Provenance),前者描述的是数据对象间横向的关系,强调的是数据对象的语义标注和多个数据对象间关联关系^[20],后者描述的是同一数据对象历史变化的纵向关系^[21],因此数据溯源理论是研究数据对象的可溯源性的基础。

数据溯源理论的概念、模型和方法对数据对象的可溯源性研究具有指导和启发作用,溯源模型可以帮助确定可溯源性研究的工作思路与流程,溯源技术可以帮助可溯源性研究提供技术保障。

国内外研究者们提出了众多理论模型,其中开放

数据溯源模型(OPM)、Provenir模型被认为是最经典的数据溯源模型,2013年W3C发布数据溯源PROV标准,并提供一种推荐的数据溯源模型——PROV数据溯源模型^[22],PROV模型主要描述的是数据或者文件实体、加工处理活动、代理人或者组织间的相互关系,所使用的PROV-O为基于不同应用和不同领域的起源信息建模^[23]。标准化的模型对于数据对象可溯源性的保障方法具有指导作用,明确数据对象可溯源性保障中需要注意处理的对象与处理的方式。

同时也出现了如DNA双螺旋模型、数据起源安全模型等一系列独创性的数据溯源模型,DNA双螺旋模型借助生物学中DNA链独特的排列方式,提出一个双链式的方式记录操作序列和数据序列的溯源方式;数据溯源安全模型在改进前人的研究成果后,提出加入时间戳来保护溯源数据本身的安全性新思路,这两个模型由于提出较早,不能完全适应当前大数据时代的数据特点,但给可溯源性保障提供了启发作用,在数据对象可溯源性保障工作中可以将生物学中DNA的多样性、特异性、由父代遗传等特点融入数据对象溯源中,不但可以高效进行溯源工作,而且还可以保护溯源数据的安全性。

数据溯源追踪的经典方法有标注追踪法、逆向查询追踪法、双向指针法、专用查询语言等,其中标注追踪法简单且易用,在数据存储时额外标注它的历史演变信息等内容。大部分数据溯源模型都采用标注法,如DBnotes、Mondrian等,但由于标注法需要额外增加标注数据的存储空间,不适用于目前海量的数据体系;而逆向查询追踪法是逆向计算,需要构造特定的逆推函数,由结果逆推出原结果,计算相对复杂,但仅在需要时计算,是一种惰性计算方式,而且储存空间需求低^[24];双向指针法仅适用科学工作流等特定数据类型,不具有普适性。考虑到大数据时代数据对象的特征,本文借鉴了逆向查询追踪法的思路以减少大量额外的储存开销。

3 数据对象版本号计算方法

由于目前常用的版本计算方法主要是面向人类用户的使用,而非面向计算机的理解与处理;数字的变化仅代表内容有所改变,并未记录变化的原因,不能很好地应用在数据对象的自动处理中。所以本文根据数据对象的生成方式提出一种新的版本计算方法,其

主要面向的是计算机的自动化处理,且兼顾人类用户的使用需求。本方法得到的版本号不仅记录了版本变化次数及变化量,而且还记录了版本变化的原因,如对上一版本数据进行“插入”(Insert)操作后得到当前版本。

3.1 数据对象的封装

本文所提出的数据对象可以看作由两个部分组成:被封装的内容及其元数据。被封装的内容可以为任何类型的数据,如文本、数值、图片、视频、富媒体及某种组合,系统将数据对象的内容作为一个整体来处理,计算其版本号。同时,将数据内容对应的所有元数据放在一个文件之中,并作为一个整体进行版本号计算,如图1所示。

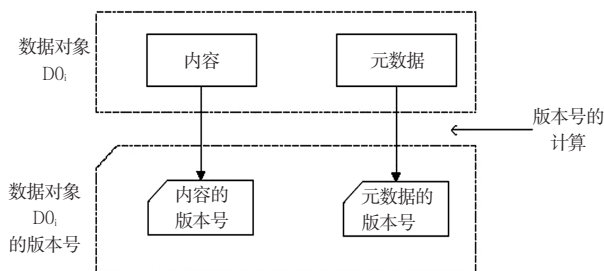


图1 数据对象的两种版本号

本文提出的版本号计算方法不仅与变化频数有关,而且还与数据对象本身的生成方式有关。数据对象(DO_j)的生成方式有两种:直接产生一个全新的数据对象和基于已有的数据对象(DO_j)产生定义新的数据对象,如图2和图3所示。

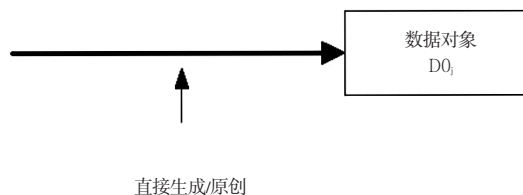


图2 数据对象的生成方式(直接生成)

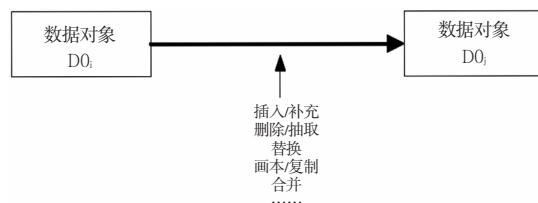


图3 数据对象的生成方式(基于其他数据对象生成)

3.2 数据对象的溯源类元数据

溯源类元数据主要用于保障数据对象的可溯源

性,记录的是数据对象的世系信息,采用的是数据溯源技术,尤其是本文提出的数据对象的版本号计算方法和数据对象的DNA计算方法。与其他类型的元数据不同的是,溯源类元数据具有凭证作用,如关联类元数据本身无法证明是否可信时需要借助溯源类元数据进行进一步验证。本文提出数据对象连续性保障中溯源类元数据至少包括以下五种:

(1)URI:数据对象的URI(Universal Resource Identifier,通用资源标识符),用于唯一标识数据对象,URI的具体实现可以采用URN(Universal Resource Name,统一资源名),即资源的逻辑名称和URL(Universal Resource Locator,通用资源定位器),即资源的定位和访问地址。本文对URI技术的实现方法不做限制,可以采用目前广泛应用的任何URI技术,如CoolURI、哈希URI(Hash URI)和303URI等。

(2)CVersion和MVersion:数据对象的内容版本号(Content Version)和元数据版本号(Metadata Version)。与传统版本号计算方法不同的是,本文采用的是两套制版本号,即内容的版本号和元数据的版本号。

(3)DNA:数据对象的DNA是本文首次提出的概念,与人类DNA类似,用于记录数据对象的世系关系,其主要影响因素为父类数据对象的DNA及当前数据对象的生成方法。

(4)Hash:数据对象的Hash值记录的是数据对象的Hash值,用于校验数据对象的完整性,即是否对数据对象进行了非法篡改。

(5)CopyrightInfo:数据对象的版权信息,记录的是数据对象的版权信息,用于溯源数据对象的版权信息的填写、维护和查看。该元数据项体现了数据空间中的主体相关性特点。

溯源类元数据的审计主要分为两个阶段:(1)检查是否有相应元数据;(2)检查元数据值是否有效。

3.3 数据对象的版本号计算方法

本文提出的数据对象的版本计算方法的基本思路和主要步骤如下:

(1)我们将数据对象的版本分为两种,即内容版本号和元数据版本号,如图4所示。

(2)版本号为变长的多位数值,其中每一个位代表一次版本变化,如数据对象DO_i的版本号为0125代表的是该数据对象(DO_i)的第四版本,即相对于原始版本已经历了三次版本变化,如图5所示。

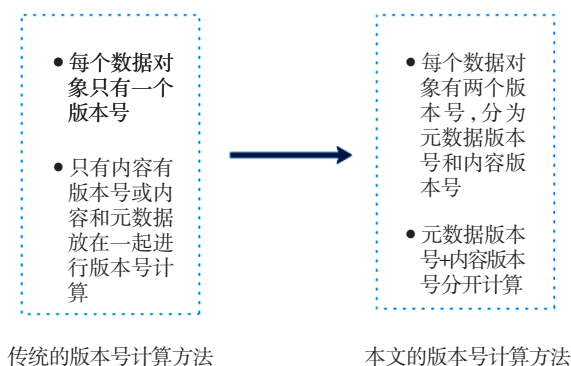


图4 本文的版本号与传统版本号的区别之一

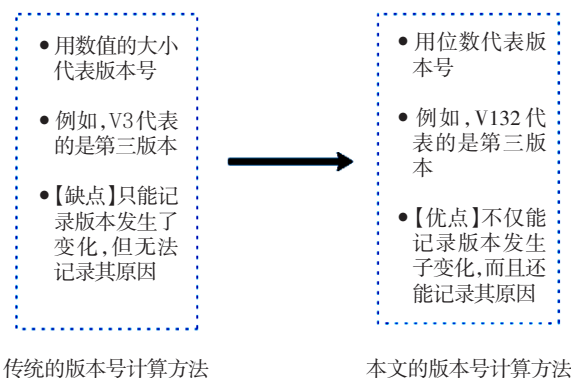


图5 本文的版本号与传统版本号的区别之二

(3)版本号的计算由版本号原子操作和算子组成。原子操作的编号有四种,具体如下:

- * 0: Create,即代表“新生成”操作
- * 1: Insert,即代表“插入”操作
- * 3: Delete,即代表“删除”或“抽取”操作
- * 5: Replace,即代表“替换”操作

(4)在本方法中,“替换(5)”操作和“插入(1)+删除(3)”是有区别的,前者代表的是在同一个内容或位置上先进行删除,后进行插入操作;后者代表的是插入操作和删除操作发生的位置并不一样。

(5)算子为加号(+)计算,代表的是几种操作同时出现,如1+3等于4,即4代表的是该版本为上一版本数据对象的基础上,进行“插入”和“删除”操作获得。以下数值有且仅有一种计算方法得到:

- * 4=1+3,即代表操作“插入(Insert)+删除(Delete)”
- * 6=5+1,即代表操作“插入(Insert)+替换(Replace)”
- * 8=3+5,即代表操作“删除(Delete)+替换(Re-

place)”

* 9=1+3+5,即代表操作“插入(Insert) + 删除(Delete) + 替换(Replace)”

(6)代号“新生成(0)”不参加任何计算,即代表的是数据对象的“新生成(Create)”代表的是新生成操作的结果,已经包含生成过程中插入(Insert) + 删除(Delete) + 替换(Replace)操作。

(7)代号2和7无法通过加号算子(+)和原子代号计算得到。因此,本文为代号2和7分别定义了以下特殊含义:

* 2:代表的是副本(Duplicate)操作,可以代表数据对象在复制操作或网络传输,充分体现数据对象的共享性。

* 7:代表的是集成/合并(Integrate)操作,说明该版本为对已有几个数据对象进行合并/集成处理后得到的。

本文提出的版本代号表示及含义如表1所示。

表1 版本代号表

代号	是否为原子代号	是否可以运算	计算方法	英文名称	含义
0	否	否	无	Create	该版本不存在父版本,属于原始数据对象,其内容或元数据的新生成
1	是	是	无	Insert	当前版本的数据对象是通过对上一版本进行了“插入/增加”新内容/元数据的方式生成
2	否	否	无	Duplicate	当前版本的数据对象是通过对上一版本进行了“副本/复制”操作的方式生成
3	是	是	无	Delete	当前版本的数据对象是通过对上一版本进行了“抽取/删除”内容/元数据的方式生成
4	否	否	1+3	Delete & Insert	当前版本的数据对象是通过对上一版本进行了“插入/增加”和“删除/抽取”的方式生成,但二者发生的位置并不在一起,否则视为Replace(替换)操作
5	是	是	无	Replace	当前版本的数据对象是通过对上一版本进行了“替换”操作的方式生成
6	否	否	5+1	Insert & Replace	当前版本的数据对象是通过对上一版本进行了“替换”和“插入”操作的方式生成,在此不限制二者的先后顺序
7	否	否	无	Merge	当前版本的数据对象是通过对其他数据对象进行“合并/集成”操作的方式生成
8	否	否	5+3	Delete & Replace	当前版本的数据对象是通过对上一版本进行了“替换”和“删除”操作的方式生成,在此不限制二者的先后顺序
9	否	否	1+5+3	Insert & Delete & Replace	当前版本的数据对象是通过对上一版本进行了“替换”、“插入”和“删除”操作的方式生成,在此不限制三种操作的先后顺序

(注:只有原子代号参加加法(+)计算)

4 数据对象DNA的生成和校验

数据对象的DNA计算方法是本文提出的创新点,

名称源自生物DNA,在结构上,数据对象中内容和元数据一一对应,相互连接,构成双链形式;在特性上,数据对象DNA具有同生物DNA相同的多样性、特异性、稳定性等特点。数据对象的DNA主要用于保障数据对象的可溯源性,解决的是有争议数据的溯源及世系检验问题。从数据计算和应用视角看,数据对象的DNA的生成必须符合以下四个基本要求:

(1)抗碰撞性:从本质上看,数据对象的DNA计算是一种映射技术,将数据对象空间映射到DNA空间。因此,如何保证数据DNA的唯一性,即较好地避免不同数据对象的DNA相互碰撞的可能性是一个关键问题。

(2)快速计算:数据对象的DNA计算必须符合快速计算的要求,计算过程不宜过于复杂。因此,如何设计一种简单有效的DNA生成方法是另一个关键问题,据研究显示Hash技术计算效率很高^[25]。为此,本文采取了Hash技术,确保数据计算的速度。

(3)单向计算:数据连续性保障工作与数据保密工作之间往往存在交叉关系,甚至可能出现相互冲突^[26]。因此,DNA计算和检验过程不得破坏数据的保密性。也就是说,数据对象的DNA计算需要遵循单向计算原则,即可以从数据对象计算出其DNA,但不能从DNA推导或还原出数据对象。单向计算的目的是保证数据的保密性需求。

(4)防伪造和篡改:数据对象的溯源具有完整性和保密性的要求^[27],因此数据对象的DNA必须满足防伪造的特点,避免其他用户伪造DNA数据,并对伪造或篡改的DNA有较强的识别能力。

根据以上四个要求本文提出了数据对象DNA的计算方法。

4.1 数据对象的Hash值

考虑到因数据对象的粒度大小(Size或Volume)不同导致的计算复杂性,本文首先将不同大小的数据对象映射成相同长度的消息摘要空间。从实现方法看,可以采取两种不同策略,即消息认证码(MAC)和Hash函数^[28],都可以将一个可变长度的消息输出为固定长度的值。MAC通过将消息加密后附在消息后,供接收方验证,MAC不要求可逆性;Hash函数是用映射的方式产生定长结果,作为验证符。

由于消息认证码的生成过程具有时间消耗长、硬件开销大等局限^[29],本文采用基于Hash函数的映射方法。基于Hash函数的映射方法与基于消息认证码(MAC)的消息映射方法的主要区别在于前者不需要加

密处理,计算速度更快。

4.2 数据对象DNA的生成

根据数据溯源安全模型的思路,模型应包括世系关系、起源记录、安全组件三部分内容,起源链可以溯源数据对象的生成及演变过程,起源记录是对数据对象的当前修改和内容的记录,安全组件防止数据对象和溯源数据遭到破坏和修改^[27]。本文提出的数据对象的DNA的实现原理如图6所示,数据对象的DNA由三个因素决定:父版本的DNA、当前版本的数据对象本身以及当前版本的产生方式决定。考虑到计算复杂度和用户体验,本文提出的数据对象的DNA并不是基于数据对象的内容及其元数据生成,而是根据数据对象的Hash值进行计算而成。其中,父版本DNA记录了数据对象的世系关系,版本号体现了当前数据对象的生成方式,Hash值代表了数据对象的当前版本。

具体过程是对数据对象的每个版本进行Hash计算的基础上,将Hash值和上一版本的DNA进行异或操作,并以版本号为密钥对异或操作的结果进行对称加密,获得当前版本的DNA。数据对象的DNA是一个固定长的鉴别码,其计算公式:

$$DNA_i = C(DO_i, K)$$

式中, DO_i 为Hash值和上一版本的DNA进行异或操作的结果; K 为双方共享的密钥; C 为MAC函数, $C(DO_i, K)$ 为MAC函数的返回值(固定长度)。

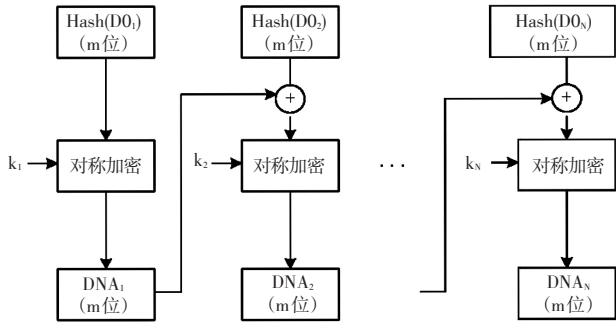


图6 数据DNA生成过程

数据对象的DNA生成详细步骤如下:

(1)计算Hash值。将计算数据对象的每个版本 DO_1, DO_2, \dots, DO_N 的Hash值,即 $Hash(DO_1), Hash(DO_2), \dots, Hash(DO_N)$ 。每个版本的Hash值的长度相同,即 m 位(固定值)。

(2)生成密钥。生成每个版本对应的密钥(Key),如版本 DO_2 对应的密钥为 k_2 。每个版本的Key为该版本的合成版本号,即对内容版本号和元数据版本号进

行二进制值异或运算的结果。版本号的生成方法见本文第二部分。

(3)生成DNA。从第一个版本(DO_1)的Hash值($Hash(DO_1)$)开始逐个计算所对应的版本的DNA值,计算公式如下:

$$DNA_1 = Ek(Hash(DO_1))$$

$$DNA_i = Ek(Hash(DO_i) \oplus DNA_{i-1}) \quad (1 < i < N)$$

从以上公式可以看出,从第二个版本(如 $Hash(DO_2)$)开始,每个版本的DNA值取决于3个因素:该版本的Hash值 $Hash(DO_i)$ 、前一个版本的DNA值(DNA_{i-1})和密钥 K_i 。也就是说,当前版本 DO_i 的DNA值(DNA_i)是对该版本的Hash值($Hash(DO_i)$)与前一个版本的DNA值(DNA_{i-1})进行异或操作($Hash(DO_i) \oplus DNA_{i-1}$)之后,采用当前版本的密钥 k 进行加密处理($Ek(M_i \oplus O_{i-1})$)后得到的值。

4.3 数据对象DNA的回溯

从数据对象的DNA生成方法可以很容易推导出其溯源方法,我们采用的Mac算法是对称加密的,易回溯推出父版本的DNA。如图7所示。以数据对象的当前版本 $DO_2 \dots N$ 的溯源为例,我们可以通过以下步骤进行溯源至其之前版本。

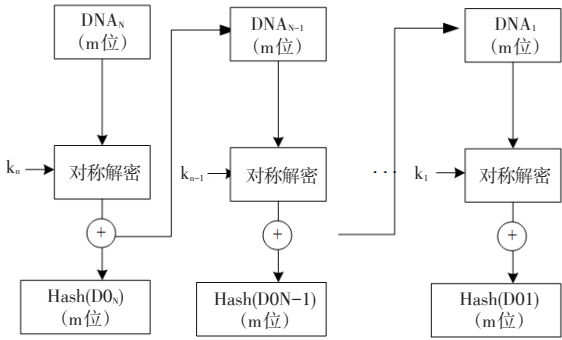


图7 数据DNA溯源过程

从第 N 个版本(DO_N)的DNA值及Hash值($Hash(DO_N)$)开始逐个计算之前版本的DNA值,其计算公式如下:

$$DNA_{N-1} = Ek(Hash(DO_N) \oplus DNA_N)$$

$$DNA_{i-1} = Ek(Hash(DO_i) \oplus DNA_i) \quad (1 < i < N)$$

如果,计算后得到的 DNA_{N-1} 与上一版本的元数据中存储的DNA一致,说明当前版本确实系上一版本的世系,以此类推。

5 数据对象DNA的IT实现及应用

本文用Python编写桌面应用,可以应用在数据对

象DNA的生成,及数据对象溯源和数据连续性审计等方面。编写的应用界面草图如图8所示。



图8 数据对象DNA生成及应用软件界面图

针对数据连续性工作的要求,本文在数据对象元数据中加入了关联类、溯源类、语义类和审计类元数据,数据对象的元数据XML部分截图如图9所示。本文采用关联类元数据记录数据对象演变过程,基于第3部分所定义的数据对象生成的10种方式记录关联类元数据,溯源类元数据记录版本号、DNA、消息摘要空间(Hash值)及版权信息,语义类元数据记录默认语义元素,审计类元数据是为数据连续性审计记录做准备。

5.1 数据对象DNA的IT实现

数据对象DNA生成的核心部分在于版本号、消息摘要空间(Hash值)及DNA的生成,版本号是依据第4部分阐明的计算方式进行处理。消息摘要空间要计算数据文件的Hash值以及版本号的Hash值,数据对象初代版本DNA只有版本号和自身的Hash值参与运算,之后版本的DNA由版本号、自身的Hash值及父代DNA运算而来。

数据文件Hash值生成函数如下所示,版本号Hash值计算方式相似:

```
def FormFileHash(path):
    if os.path.isfile(path):
        fp=open(path,'rb')
        contents=fp.read()
        fp.close()
        filehash=hashlib.md5(contents).hexdigest()
#用Hash函数映射为定长
    return filehash
    else:
        print('File doesn't exist')
```

```
- <metadata>
- <correlation>
    <!--关联类元数据-->
    <InsertOn/>
    <DuplicateFrom/>
    <DeleteOn/>
    <InsertAndDeleteOn/>
    <ReplaceOn/>
    <InsertAndReplaceOn/>
    <MergeFrom/>
    <DeleteAndReplaceOn/>
    <InsertDeleteAndReplaceOn/>
    <isFirstVersion/>
</correlation>
- <headward>
    <!--溯源类元数据-->
    <URI/>
    <CVersion/>
    <MVersion/>
    <DNA/>
    <Hash/>
    <CopyrightInfo/>
</headward>
- <semantic>
    <!--语义类元数据-->
    <what/>
    <when/>
    <where/>
    <who/>
    <how/>
    <why/>
</semantic>
- <audit>
    <!--审计类元数据-->
    <Auditor/>
    <AuditHash/>
    <AuditScore/>
```

图9 数据对象元数据部分截图

数据对象初代版本DNA计算方式如下:

```
def FormFirstDNA(path):
    #版本号Hash值计算并生成加密算法的key
    VersionHash=FormVersionHash(FormVersion
    (path))
    key = binascii.unhexlify(VersionHash)
    cipher = AES.new(key, AES.MODE_ECB)
    #数据文件Hash值
    text=binascii.unhexlify(FormFileHash(path))
    #初代DNA生成
    DNA =cipher.encrypt(text)
    return binascii.hexlify(DNA)
```

数据对象后代版本DNA计算方式如下

```
def FormSonDNA(path,FatherDNA):
    #父代DNA和数据文件异或操作
    XOR=hex(int(FormFileHash(path), 16)^(int(Fat
    herDNA, 16))).replace("0x","")
    VersionHash=FormVersionHash(FormVersion
```


(path))

```
key = binascii.unhexlify(VersionHash)
cipher = AES.new(key, AES.MODE_ECB)
DNA = cipher.encrypt(XOR)
return binascii.hexlify(DNA)
```

将版本号、DNA、Hash 值、语义信息等元数据信息写入数据对象元数据文件之后,就得到完整的数据对象,方便我们对于数据对象的进一步应用和管理。

5.2 数据对象的世系检验

根据本文第4部分提出的数据对象生成和回溯方式,可以用逆向追踪检验任意两个数据对象的是否存在世系关系,判断两个数据对象是否同源的原理如图10所示。以两个数据对象的DNA和版本号作为判断依据,因为有版本号作为基础判断依据,可以比逆向追踪法更快的排除无关联关系,更加快速准确地判断其二者间的世系关系,充分体现了数据对象的可溯源性。

详细数据对象世系检验方法原理步骤如下:

(1)读取数据对象DO_i和DO_j中所储存的DNA、消息摘要空间(Hash值)和版本号;

(2)数据对象DO_i版本号为 m 位,数据对象DO_j版本号为 n 位($m < n$),判断DO_i版本号是否和DO_j版本号前 m 位完全相同,如果完全相同说明两者有可能是经由同世系变化而来,若版本号前 m 位存在差异则直接证明两者不可能为同世系;

(3)若版本号前 m 位完全相同,使用本文3.4部分中的计算方法回溯数据对象DO_j($n-m$)次,可以得到一个新的DNA _{$j-(n-m)$} ;

(4)判断DNA _{$j-(n-m)$} 与DNA _{i} 是否相等,若其相等可以证明数据对象DO_i就是DO_j的祖先版本,反之,则

二者无关。

5.3 数据对象的自动化审计

数据对象DNA可以帮助检验两个数据对象间的世系关系,因此可以借助数据对象的DNA来审计其可溯源性,进而分析数据连续性。数据连续性的实现可以借助人工或自动化审计等方法来实现。通过数据连续性审计工作,不仅可以对数据对象的可连续性特征进行量化分析,而且可以记录审计过程中发现的问题,进而为数据对象的开发和利用奠定基础。本文提出的自动化审计流程如图11所示。

通过审计数据对象的元数据中记录的Hash值、版本号、DNA等信息,审计数据对象是否可信,是否具有可关联性、可溯源性和可理解性是数据连续性审计的关键,主要的审计思路如下:

(1)检查数据对象是否带有效Hash值。元数据项中Hash值的功能是判断数据对象的完整性是否已被破坏,以便识别数据对象的元数据生成之后是否在数据内容上发生了新的改动,或者说数据内容的改动是否体现在数据对象的元数据之中。判断方法为重新计算Hash值,并判断重新计算的Hash值与数据对象自带的元数据中的Hash值是否相同。

(2)可关联性审计要读取数据对象的关联类元数据,依次遍历已存在元数据(M_i),并判断数据对象的改变是否与数据对象的版本号记录一致,这两者可以互为补充,建立良好的审计机制。

(3)可溯源性审计主要是重新计算DNA值,分析DNA、版本号及关联数据之间的对应关系,并判断新计算出的DNA值是否与数据对象的自描述信息中所包含的DNA值一致及判断是否带有效版权信息。数据可溯源性校验需要查阅是否有版权信息,版权信息的存在

代表着数据对象可以回溯的具体主体,判断方法为数据对象的版权信息是否与所对应的版权协议一致。

(4)可理解性校验的主要依据为领域本体及元数据的模式(Schema)信息。校验过程涉及两个问题:一是判断对应元数据项是否已记录;二是判断已记录的元数据项是否有效。

(5)最后,将上述步骤中的结果及问题列表进行汇总,形成数据对象的连续性评分及问题列表,作为数据连续性审计分析总报告的内容之一。

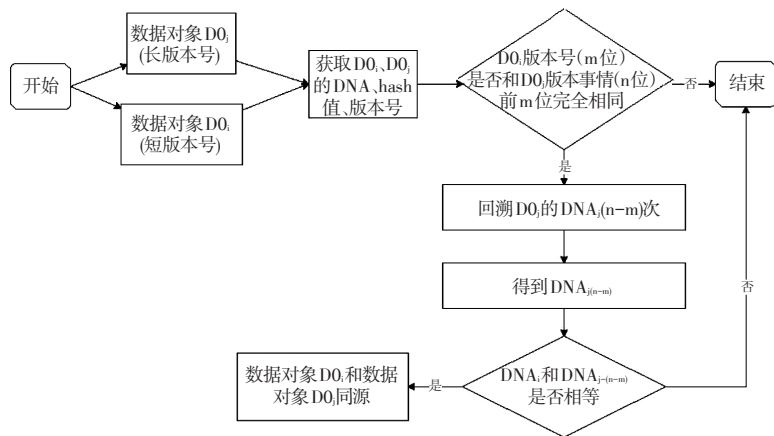


图10 数据对象世系检验方法原理

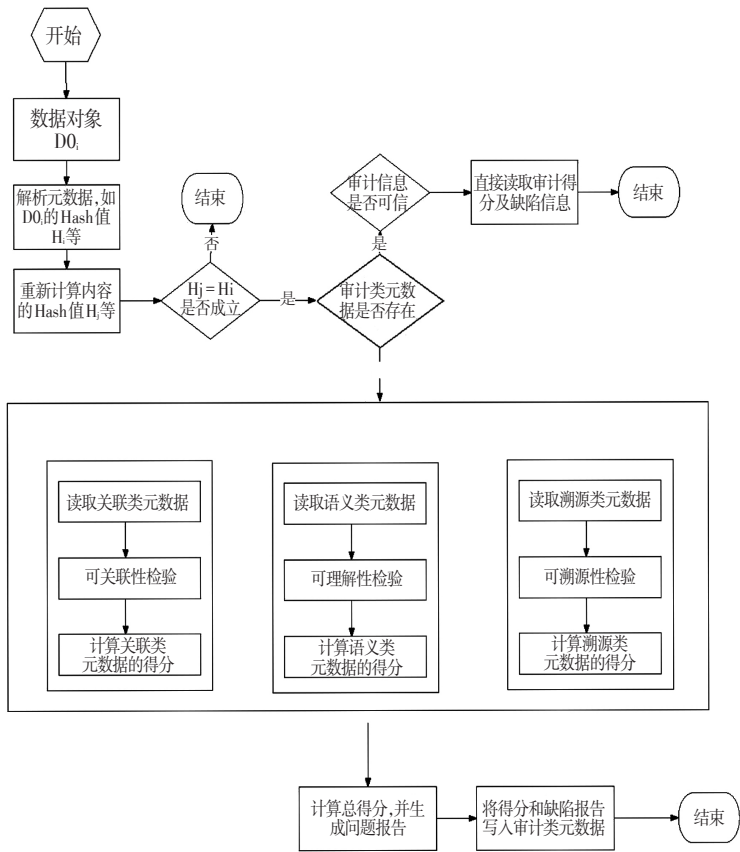


图 11 数据连续性审计流程

6 结论

本文借用数据溯源理论的技术和方法,创造性地提出一种新的数据对象版本号计算方式和数据对象 DNA 的概念,并在此基础上开发出具有数据对象 DNA 生成和数据连续性审计功能的应用,切实保障数据对象的可溯源性。本文提出的数据对象可溯源性保障方法具有以下优势:

(1)易于生成和快速计算。本文提出的数据连续性保障方法的核心是数据对象版本号计算方式与数据对象 DNA 的生成。其中,数据对象的版本号计算方式简单灵活,用 10 个数字简单快捷地记录数据对象的变更及生成过程,易于计算机计算和处理;数据对象 DNA 的生成需要借助 Hash 函数生成消息映射空间, Hash 函数计算简单快速,且用固定长度的消息映射空间代表原数据对象的数据,消除了数据对象粒度大小不同带来的计算复杂性;在数据对象的可溯源性审计时,仅需重新计算 Hash 值与 DNA,就可以完成对数据对象完整性和可溯源性的审计。

(2)支持跨领域多数据类型。通过有限长度的 DNA 字段不但记录了本代数据的特点,还记录了与父代数据对象的世系关系,比数据溯源理论中常用的为数据对象添加注释的方法更为灵活,更适用于当前大数据环境下多源、海量、异构的数据对象。通过逆向查询来保障数据对象的可溯源性是一种节约计算内存的方式,逆向查询是一种惰性计算方法,仅在需要时才进行计算,减小了对于系统运行内存的消耗。

(3)抗碰撞性防篡改性强。数据对象的 DNA 由父代版本的 DNA、自身的消息映射空间和版本号共同决定,其计算依据加密映射技术,在技术层面上就具有很强的抗碰撞性和保密性。消息映射空间和父代版本的 DNA 异或操作不但使得数据对象自身的安全性和完整性得到保护,同时也保护了其世系关系。因此本文所提出的方法也可以满足对数据溯源理论中对数据安全性及溯源数据安全性的要求。

(4)支持数据质量的自动化审计。本文提出的数据对象版本号计算方式突破了元数据中原有版本记录由简单数字累加的形式,将更改内容记录在版本号中。这不仅满足面向人类的识读,还有益于计算机识别和自动处理有限长度的数据对象 DNA 包含数据对象的内容、演变过程等信息,更加便于处理和计算,为数据对象的管理和数据连续性的自动审计提供基础,有效减轻人力维护成本。

(5)具有更强的普世价值。因为具有上述特性与优势,本文提出的方法比原有数据溯源领域的方法有更强的普世价值,不受数据类型和数据体量的限制,可以应用于不同的数据对象和不同的行业领域,尤其适用于多源海量数据且对其连续性和可靠性有要求的具体应用领域,具有更广阔的应用前景。

本文所提出的数据对象可溯源性保障方法——数据对象 DNA,将有助于审计数据对象的可溯源性,有助于解决数据管理中有争议数据的溯源问题并有效解决数据失信问题,为数据溯源工作提供一种新思路,对数据对象自动化管理和审计将有重要意义,可适用于大数据环境下多源海量数据及各个行业领域,也便于进行自动化处理,减轻计算成本和维护成本。下一步,我

们将进一步拓展本文的研究范围,尤其需要优先开展两种特殊方法的专题研究:一种是面向特定领域的数据可溯源性验证方法,另一种是面向特定领域的数据可溯源性修复方法。其中,数据可溯源性验证方法的研究应结合不同领域的特殊需求,提出特定的验证方法,从而支持碎片信息的可信度评价;数据可溯源性修复方法研究应建立在数据可溯源性验证方法的基础上,从数据的可关联性、可溯源性和可解释性三个方面修复数据,进而提高数据碎片的连续性。

参考文献

- [1] 陈红玉,翟 军,袁长峰,等.开放政府数据的溯源元数据研究及应用[J].情报杂志,2017,36(6):148-155.
- [2] 朝乐门.数据连续性:未来跨学科研究的重要课题[J].情报学报,2016,35(3):227-236.
- [3] 刘 冰,庞 琳.国内外大数据质量研究述评[J].情报学报,2019,38(2):217-226.
- [4] 徐 扬,王申罡.数据起源研究进展[J].情报理论与实践,2016,39(7):136-140,135.
- [5] 张 翔,张福炎.数字图书馆中数据对象的描述和检索[J].计算机工程与应用,2001(8):114-117.
- [6] 胡勇其.基于语义的数据对象访问和存储管理研究[D].北京:中国科学院研究生院(计算技术研究所),2006.
- [7] Greenberg J. Understanding metadata and metadata schemes [J]. Cataloging & Classification Quarterly, 2005, 40(3-4): 17-36.
- [8] Márquez, Fausto Pedro García, Lev B. Big Data Management [M]. Cham: Springer Nature, 2016:92-93.
- [9] Tamura K, Matsutani H. An In-Kernel NOSQL cache for range queries using FPGA NIC[C]. International Conference on Fpga Reconfiguration for General-purpose Computing. IEEE, 2016.
- [10] 罗文武.电子文件封装策略比较研究[D].杭州:浙江大学,2013.
- [11] Shajeemohan B S, Govindan V K. Compression scheme for faster and secure data transmission over internet[J]. Computer Science, 2006(1):1-4.
- [12] Kwon J W, Moon S M. Web application migration with closure reconstruction[C]. ACM Press the 26th International Conference, 2017:133-142.
- [13] 雷映喜,习淑婷,彭俊峰,等.XML与JSON在WEB中对数据封装解析的对比[J].价值工程,2013(9):210-211.
- [14] O'Sullivan B, Bryan O. Mercurial: The Definitive Guide[M]. Boston: O'Reilly Media, 2009.
- [15] Soules C A N, Goodson G R, Strunk J D, et al. Metadata efficiency in versioning file systems[C]. Usenix Conference on File & Storage Technologies. 2003.
- [16] Preston-Werner T. Semantic Versioning 2.0.0[EB/OL]. [2019-03-01].<https://semver.org/#semantic-versioning-200>.
- [17] Wikipedia. Software Versioning [EB/OL]. [2019-04-01].https://en.wikipedia.org/wiki/Software_versioning.
- [18] Thilo Planz. Vermongo: Simple Document Versioning with MongoDB[EB/OL]. [2019-03-18].<https://github.com/thiloplanz/v7files/wiki/Vermongo>.
- [19] Knuth D E. The Future of TEX and METAFONT [EB/OL]. [2019-04-01]. <http://www.ntg.nl/maps/05/34.pdf>.
- [20] Tang H L, Pan M, Jiang S P, et al. Linked data: evolving the web into a global data space[J]. Molecular Ecology, 2011, 11(2):670-684.
- [21] Ikeda R, Widom J. Panda: a system for provenance and data [J]. Bulletin of the Technical Committee on Data Engineering, 2010, 33(3):42-49.
- [22] W3C Recommendation. PROV-DM: The PROV Data Model [EB/OL]. [2019-03-18].<https://www.w3.org/TR/2013/REC-prov-dm-20130430/Overview.html>.
- [23] 倪 静,孟宪学.关联数据环境下数据溯源描述语言的比较研究[J].现代图书情报技术,2013(2):18-23.
- [24] 明 华,张 勇,符小辉.数据溯源技术综述[J].小型微型计算机系统,2012,33(9):1917-1923.
- [25] 祖克珂,郑 宇,何大可.嵌入式环境下HASH算法的效率分析[J].计算机应用,2008,28(b06):312-314.
- [26] Davidson S B, Roy S. Provenance: Privacy and Security[M]// Liu L, Özsu M T. Encyclopedia of Database Systems. New York: Springer, 2018: 2927-2932.
- [27] Hasan R, Sion R, Winslett M. The case of the fake Picasso: preventing history forgery with secure provenance[C]. Proceedings of the Conference on File & Storage Technologies. USE-NIX Association, 2009.
- [28] William Stallings. 密码编码学与网络安全[M]. 刘玉珍,王丽娜,傅建明,等译.北京:电子工业出版社,2004.
- [29] Rehman S U, Bilal M, Ahmad B, et al. Comparison based analysis of different cryptographic and encryption techniques using Message Authentication Code (MAC) in Wireless Sensor Networks (WSN)[J]. International Journal of Computer Science Issues, 2012, 9(1):96-101.

[作者简介]朝乐门,男,1979年生,中国人民大学副教授,博士生导师。
李昊璟,男,1999年生,中国人民大学信息资源管理学院本科生。
冀佳钰,女,1996年生,中国人民大学信息资源管理学院硕士研究生。
收稿日期:2019-09-08