基于安全比较码的云环境隐私保护排序方法

任 晖 戴 华 杨 庚

(南京邮电大学计算机学院 南京 210023) (江苏省大数据安全与智能处理重点实验室 南京 210023)

摘 要 基于云计算的外包服务模式因节省计算、存储等资源配置和维护成本而被越来越多的公司和个人所使用。然而,资源外包模式也使得数据拥有者失去对其数据的直接控制,敏感数据的隐私保护问题日益凸显。排序是计算机中常用的一种操作,数据加密是云环境中常用的隐私保护策略。如何在不泄露明文信息的前提下实现基于密文的隐私保护排序,是一个难点问题。文中提出面向云环境的基于安全比较码的隐私保护排序方法。通过引入0-1编码和HMAC来构造安全比较码机制;数据所有者对其敏感数据进行加密和编码预处理,将生成的密文和安全比较码外包存储至云服务端;此时云服务器即可利用安全比较码实现无需明文数值参与的密文数据排序,从而实现针对数据拥有者外包数据的隐私保护排序。实验结果表明,隐私保护排序方法在时间和空间上均优于现有同类方法。

关键词 云计算,数据外包,隐私保护,排序

中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.05.023

Secure Comparator Based Privacy-preserving Sorting Algorithms for Clouds

REN Hui DAI Hua YANG Geng

(College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)

(Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, Nanjing 210023, China)

Abstract Data outsourcing is accepted by more and more companies and individuals due to its profits on low costs of resource configuration and maintenance. However, it makes owners lose control of their owned data, which cause privacy-preserving problems of sensitive data. Sorting is a common operation in computer applications. It is a challenge to implement privacy-preserving sorting over encrypted data without leaking plaintext. This paper proposed secure comparator based privacy-preserving sorting algorithms. Secure comparator is constructed by 0-1 coding and HMAC techniques, which can be used to compare two data items without knowing their real values. Data owners firstly encrypt their data and then outsource the generated corresponding secure comparators into clouds. Cloud servers can sort the outsourced encrypted data according to their corresponding secure comparators by using the proposed privacy-preserving sorting algorithms. Experiment results show that the proposed privacy-preserving sorting algorithms have better performance on time and space metrics than other algorithms.

Keywords Cloud computing, Data outsourcing, Privacy preservation, Sorting

1 引言

随着云计算技术的成熟与广泛应用,以Amazon EC2和Google App Engine等为代表的"云"服务模式得到快速的发展和普及[1]。在云环境中,云服务提供商通过建立云服务器(Cloud Server,CS)来提供存储、计算等服务;数据拥有者(Data Owner,DO)为节省软硬件的配置和维护成本,将其拥有的数据外包存储至CS端。但是,数据的远程外包存储使得DO失去了对数据的直接管理和控制权,从而势必存在被CS窥探敏感数据的威胁,这也在一定程度上阻碍了云计算的进一步推广和应用。当前,如何在提供可靠、高效云计算服务的同时保护用户数据的隐私,已成为研究和应用领域的一个热点问题[2-4]。

数据加密是保护云端数据隐私的常用技术手段,而排序是各种实际应用中常用的基本方法之一。如何在不泄露明文信息的前提下实现云端基于密文的排序,是一个难点问题。目前,在云环境下的基于保序加密的隐私保护排序方法[5-6]存在安全性问题,而基于全同态加密的隐私保护排序方法[7-8]则

到稿日期:2017-03-27 返修日期:2017-12-11 本文受国家自然科学基金项目(61300240,61402014,61572263,61472193,61373138),江苏省自然科学项目(BK20151511,BK20161516),中国博士后科学基金(2015M581794),江苏省高校自然科学研究项目(15KJB520027),安徽省自然科学基金项目(1608085MF127),江苏省博士后科研资助计划(1501023C),南京邮电大学校级科研基金(NY214127)资助。

任 晖(1993-),男,硕士,主要研究方向为面向云环境的安全数据管理,E-mail:haypo@live.cn;戴 华(1982-),男,博士,副教授,主要研究方向为分布式数据管理与安全、数据库安全技术,E-mail:daihua@njupt.edu.cn(通信作者);杨 庚(1961-),男,博士,教授,主要研究方向为网络与信息安全、分布与并行计算、大数据隐私保护。

存在排序效率低和差错率高的问题。因此,研究安全、高效的面向云环境的隐私保护排序方法具有重要的实际意义。

本文假设云服务提供商遵循“半诚实模型”[9]提供服务,即 CS 能够严格遵守既定协议和算法执行,但有窥探 DO 数据隐私的企图。基于该安全模型,本文提出了一类面向云环境的基于安全比较码的隐私保护排序方法(Secure Comparator based Privacy-preserving Sorting)。数据拥有者首先对其敏感数据进行预处理,即利用对称加密方法对敏感数据进行加密处理,并结合 0-1 编码和 HMAC 生成对应的安全比较码;然后再将密文数据和相应的编码外包存储至云服务端。我们利用安全比较码无需明文参与的数值比较特性,结合基于明文的经典排序方法,设计并给出了基于安全比较码的隐私保护排序算法,以实现针对密文数据的排序。实验结果表明,本文提出的隐私保护排序方法在时间和空间效率上均显著优于现有的同类方法。

2 相关工作

现有的能够实现云环境隐私保护排序的主要方法有两种:1)基于保序加密的隐私保护排序;2)基于同态加密的隐私保护排序。

保序加密(Order Preserving Encryption)[10]是一种明文偏序关系与对应密文的偏序关系保持一致的数据加密方法。黄汝维等提出了一种基于随机树的保序加密算法 OPEART(Order-preserving Encryption Algorithm based on Random Tree)[5],该算法支持数据库的索引创建和快速检索。周雄等对 OPEART 算法进行改进,提出一种基于随机间隔的保序加密算法(OPERI)[6]。显然,数据加密的保序特性能够应用到云环境中,以实现数据的隐私保护排序。然而,保序加密的安全性相对较低,特别是在密文数据样本较多时,其对于唯密文攻击的防范能力将显著降低[11]。

Gentry 等[12]提出的全同态加密(Fully Homomorphic Encryption,FHE)是一种特殊的加密方法,满足:对密文进行特定运算(加法、乘法等)得到的结果,与对该密文对应的明文执行同样的运算得到的数据再进行加密所得到的结果完全一致。Melchor 等[13]提出将 FHE 用于实现针对整型数据隐私保护排序的思想;Chatterjee 等[14]在此基础上提出了基于 FHE 的密文排序算法(FHES),该方法通过减少重加密次数来获得更高的排序效率,并将全同态密文排序算法应用到云环境中[7-8]。FHES 方法由于在排序时需要对密文数据进行重加密,时空效率较低,并不适用于存储和管理大规模数据的云计算场景;此外,该方法还存在一定的排序错误率。

3 预备知识

0-1 编码是 Lin 等[15]提出的能够解决“姚氏百万富翁问题”[16]的安全多方计算方法。设包含 w 个二进制位的数值 $x=b_1b_2\cdots b_{w-1}b_w$($b_i\in\{0,1\}$且 $1\leqslant i\leqslant w$)。根据二进制编码中“0”和“1”的位置特点,分别对 x 进行 0 编码和 1 编码,得到的 0 编码集合记为 $E_0(x)=\{b_1b_2\cdots b_{i-1}1\mid b_i=0\wedge i\in\{1,2,\cdots,w\}\}$,1 编码集合记为 $E_1(x)=\{b_1b_2\cdots b_i\mid b_i=1\wedge i\in\{1,2,\cdots,w\}\}$。其中,$b_1b_2\cdots b_{w-1}b_w$ 是数值 x 的二进制编码表示,b_1 为最高位,b_w 为最低位。由文献[15]可知,对于包含 w 个二进制位的数值 x 和 y,当且仅当 $E_1(x)\cap E_0(y)\neq\emptyset$ 时 $x>y$ 成立。该结论表明,对于任意两个数据而言,可以利用其相应的 0-1 编码进行大小比较,而无需其明文数值的参与。此外,对于任意数据 x 而言,有 $|E_0(x)|+|E_1(x)|=w$ 成立,且满足 $1\leqslant|E_0(x)|\leqslant w\wedge 1\leqslant|E_1(x)|\leqslant w$。

对于任意数据的 0-1 编码而言,由于原始编码存在可逆性,因此在利用 0-1 编码实现数据的秘密比较时,通常会使用单向 Hash(如 HMAC 等)进行安全化处理。

4 面向云环境的隐私保护排序算法

4.1 数据预处理

为保护外包存储至 CS 端的数据的私密性,并支持数据在 CS 端的隐私保护排序,DO 首先对需要外包存储的数据进行加密和编码等预处理,其中加密采用对称加密(如 DES,AES 等)来保护数据隐私,而编码则采用 0-1 编码和 Hash 身份认证编码(HMAC)来生成支持密文排序功能的安全比较码;然后,将对应的密文和安全比较码发送至 CS 端。

定义 1(安全比较码,Secure Comparator) 对于任意数据 x,利用 0-1 编码和 HMAC 生成编码数据 $H_g(E_0(x))$ 和 $H_g(E_1(x))$,其中前者称为 0 型安全比较码,后者称为 1 型安全比较码,g 为 HMAC 密钥。

由于 HMAC 具有单向不可逆和低碰撞率特性[17],根据 0-1 编码的数值比较方法可知,当且仅当 $H_g(E_1(x))\cap H_g(E_0(y))\neq\emptyset$ 时 $x>y$ 成立,即安全比较码能够实现无需数据明文参与的秘密比较。

设需外包存储的数据集为 $D=\{d_1,d_2,\cdots,d_n\}$,则 DO 针对该数据集的预处理过程如算法 1 所示。其中,k 和 g 分别为 DO 私有的加密密钥和 HMAC 密钥。

算法 1 数据预处理算法

1. 对$\{d_1,d_2,\cdots,d_n\}$分别进行加密处理,从而生成密文集$\{(d_1)_k,(d_2)_k,\cdots,(d_n)_k\}$;
2. 对$\{d_1,d2,\cdots,d_n\}$分别进行 0-1 编码处理,生成 0-1 编码集合$\{E_0(d_1),E_1(d_1),E_0(d_2),E_1(d_2),\cdots,E_0(d_n),E_1(d_n)\}$,然后再分别进行 HMAC 处理,生成安全比较码集合$\{H_g(E_0(d_1)),H_g(E_1(d_1)),H_g(E_0(d_2)),H_g(E_1(d_2)),\cdots,H_g(E_0(d_n)),Hg(E_1(d_n))\}$;
3. 将步骤 1 和步骤 2 生成的密文以及相应的安全比较码所构成的三元组集合上传并存储至 CS,如下式所示:

$$DO\Rightarrow CS:\left\{\begin{array}{c}\langle(d_1)_k,H_g(E_0(d_1)),H_g(E_1(d_1))\rangle,\\ \langle(d_2)_k,H_g(E_0(d_2)),H_g(E_1(d_2))\rangle,\\ \cdots\\ \langle(d_n)_k,H_g(E_0(d_n)),H_g(E_1(d_n))\rangle\end{array}\right\}$$

4.2 基于安全比较码的隐私保护排序算法

CS 接收到来自 DO 的外包数据后,即可针对该外包数据进行排序。显然,传统的基于明文的数据排序算法(如归并排序、快速排序、堆排序等)无法解决针对密文的排序问题,但通过引入安全比较码机制,则能够实现基于密文的无需明文参与的隐私保护排序。

本文以归并排序为例,通过引入安全比较码机制,给出支

持 CS 端密文排序功能的隐私保护归并排序算法。

为描述简便,将 CS 接收到的密文及对应的安全比较码构成的三元组集合记为$\{c_1, c_2, \cdots, c_n\}$。算法 2 中,$c_i.e_0$ 和 $c_i.e_1$ 分别表示密文$(d_i)_k$ 及其相应的 0 型和 1 型安全比较码。

算法 2　隐私保护归并排序算法 PMS

```
PMS({c1,c2,…,cn},l,h)
Begin
m=(l+h)/2;
IF l<h THEN
    PMS({c1,c2,…,cn},l,m);
    PMS({c1,c2,…,cn},m+1,h);
    Merge({c1,c2,…,cn},l,m,h);
END IF
END

Merge({c1,c2,…,cn},l,m,h)
Begin
  i=l,j=m+1,k=0;
Create({c1′,c2′,…,ch-l+1′});
WHILE i≤m∧j≤h DO
    IF ci.e1∩cj.e0≠Ø THEN ck′=cj,k++,j++;
    ELSE ck′=ci,k++,i++;
    END IF
    WHILE i≤m DO
      ck′=ci,k++,i++;
      END WHILE
    WHILE j≤h DO
      ck′=cj,k++,j++;
    END WHILE
k=0;
    WHILE k<h-l+1 DO
cl+k=ck′,k++;
    END WHILE
  END WHILE
END
```

算法 2 中,$Create(\{c_1', c_2', \cdots, c_{h-l+1}'\})$表示创建包含 $h-l+1$个由密文及对应的安全比较码构成的三元组集合变量$\{c_1', c_2', \cdots, c_{h-l+1}'\}$。其他诸如快速排序、堆排序等经典的排序算法也可以借鉴算法 2 通过引入安全比较码来实现相应的隐私保护排序,具体算法过程不再一一赘述。本文将在实验部分给出类似的隐私保护排序算法的性能测试。

由于排序过程中需要比较关键字的大小,因此比较次数决定了排序算法的执行效率。基于安全比较码的数值比较需要验证编码集合之间交集的存在性,因此与相应的明文排序算法相比,其在实现密文排序时还需要额外的比较运算。根据算法 2 可知,PMS 的排序时间复杂度为 $O(n \times \log_2 n \times w^2)$,其中 w 为排序数据明文数值的二进制位数。

4.3　安全性分析

对于 CS 而言,DO 外包存储至 CS 端的数据包含两类:1)由 DO 对其私有数据进行加密之后生成的密文数据;2)支持密文排序的由 HMAC 数据构成的安全比较码。对于密文数据而言,由于加密密钥仅为 DO 私有,因此 CS 利用其存储的密文获取对应的明文数据的难度与破解加密算法的难度[18]相同;对于由 HMAC 数据构成的安全比较码而言,HMAC 密钥也仅为 DO 私有,HMAC 的雪崩效应以及单向不可逆性[17]使得 HMAC 数据的逆向推测具有计算复杂的特点,从而使得 CS 无法利用安全比较码反向推测其对应的明文数据。因此,对于 DO 外包存储至 CS 端的数据而言,本文提出的隐私保护排序方法不仅能够保证数据的私密性,而且实现了针对密文数据的排序。

5　实验与分析

类似于隐私保护归并排序 PMS,本文实验同样利用安全比较码机制实现了隐私保护堆排序 PHS 和隐私保护快速排序 PQS,并分别从数据预处理时间消耗、排序时间消耗和数据空间占用这 3 个方面将所提方法与 Chatterjee 等提出的 FHES 方法进行了性能对比。

本文实现 PMS,PHS,PQS 和 FHES 的软件环境为 Windows10 操作系统和 Netbeans8;硬件环境为 Core i5 5200U 内核,8 GB 内存。实验数据集由随机数发生器生成,数据加密采用 AES-128,HMAC 采用 128 位的 HMAC-MD5。

5.1　数据预处理耗时的对比实验

DO 在外包数据之前,须对数据进行相应的预处理。文献[7]中 FHES 的预处理过程主要是对明文数据进行全同态加密;而本文提出的方法的预处理过程主要是对数据进行对称加密,并生成相应的安全比较码。在给定 4 组不同规模的数据集上,对这两类方法的数据预处理过程的时间性能进行评测,实验结果如表 1 所列。

表 1　FHES 预处理和本文方法预处理耗时的对比

Table 1　Comparison of time cost of preprocessing in FHES and the proposed methods

(单位:ms)

数据量	FHES 的预处理	本文方法的预处理
1000	7332.6	141.02
3000	22966.2	395.75
5000	37452.1	657.1
7000	52339.2	866.15

由表 1 可知,随着测试数据集规模的增大,两种预处理方法的耗时都逐渐增加,但本文的数据预处理耗时显著少于 FHES 方法。其主要原因是:在 FHES 中,数据经由时间复杂度为 $O(\lambda^{10})$的按位加密生成规模较大的密文数据[19],其中 λ 为整型安全系数,其值的大小与安全性正相关;而本文预处理主要采用对称加密的方式生成密文数据,同时利用 HMAC 生成安全比较码,数据处理的复杂性远低于 FHES 中的全同态加密过程。因此,本文的数据预处理过程的耗时明显少于 FHES 方法的数据预处理耗时。

5.2　排序耗时的对比实验

为了更好地与 FHES 方法[7]进行排序耗时的对比,在文献[7]中相同规模的数据集(分别为 10,20,30 和 40)上执行 FHES 排序方法和本文提出的排序方法,两种方法的耗时结果如表 2 所列。

表 2 FHES 方法与本文方法耗时的对比

Table 2 Comparison of time cost of sorting between FHES and the proposed methods

(单位:ms)

数据量	FHES	PMS	PHS	PQS
10	992235	0.12	0.125	0.111
20	1182491	0.174	0.265	0.187
30	1305123	0.443	0.481	0.266
40	1412327	0.606	0.703	0.417

由表 2 的实验结果可知,FHES 排序的耗时显著多于本文提出的隐私保护排序算法的耗时。其主要原因是:1)在 FHES 中,明文数据经由按位操作的全同态加密过程生成的密文数据占用的空间较大,而本文提出的方法生成的密文数据和编码数据占用的空间远小于 FHES(见实验 5.3),降低排序数据的对象空间占用可提升排序效率;2)FHES 在进行比较运算时需要进行大量的按位大数运算,且包含重加密过程,以将加密时产生的噪声控制在一定的阈值范围内,但这也造成了较大的时间消耗。

为了对本文提出的隐私保护排序方法 PMS,PHS 和 PQS 做进一步分析,以随机生成的规模分别为 1000,3000,5000,10000,20000 的数据集为测试对象,对本文提出的隐私保护排序方法进行排序耗时的性能测试,结果如图 1 所示。

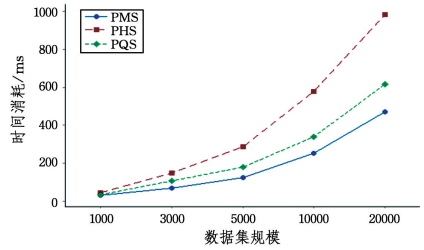

图 1 本文提出的隐私保护排序算法的耗时比较

Fig. 1 Comparison of time cost of sorting in PMS,PHS and PQS

由图 1 可知,随着数据集规模的增大,本文提出的隐私保护排序算法的排序耗时均同步增长,且增长趋势一致。具体而言,PMS 的时间消耗最少,PQS 的耗时略多于 PMS,PHS 耗时最多。

5.3 数据占空的对比实验

同样以随机生成的规模分别为 1000,3000,5000,10000 和 20000 的数据集作为排序对象,对 FHES 和本文提出的隐私保护排序方法所产生的编码数据的空间占用进行评测实验,结果如表 3 所列。

表 3 各方法的数据空间占用的对比

Table 3 Comparison of space cost of different methods

(单位:MB)

数据量	1000	3000	5000	10000	20000
FHES	83.98	251.95	419.92	839.84	1679.69
PMS	0.96	2.88	4.81	9.61	19.23
PHS	0.96	2.88	4.81	9.61	19.23
PQS	0.96	2.88	4.81	9.61	19.23

由表 3 可知,随着测试数据规模的增大,FHES 以及本文提出的隐私保护方法的编码数据的占空均增大,但本文方法显著低于 FHES;且对于相同规模的数据集,本文方法的编码数据占用的空间都相同。其主要原因在于:FHES 需要按位生成空间占用较大的大数和大浮点数数组,而重加密过程则进一步增加了编码数据的空间占用开销,使得 FHES 的空间开销显著高于本文提出的方法;此外,由于采用了基于安全比较码的数据比较策略,当排序数据规模不变时,本文提出的方法所生成的编码数据规模也不变。

结束语 在数据外包存储的云环境中,数据的外包托管特性导致数据的私密性容易受到威胁。排序是数据预处理方法中的一种常用手段,如何在外包云环境中实现针对敏感数据的隐私保护排序,是一个亟待解决的关键问题。本文通过引入 0-1 编码和 HMAC,提出无需明文数值参与的安全比较码机制,并基于此提出面向云环境的隐私保护排序方法。实验结果表明,本文提出的方法在时间和空间上的性能都显著优于现有方法。在后续工作中,我们将考虑如何对编码方法进行优化,以进一步降低隐私保护排序方法的时空开销。

参 考 文 献

[1] DING Y,WANG H M,SHI P C,et al. Trusted Cloud Service [J]. Chinese Journal of Computers,2015,38(1):133-149. (in Chinese)
丁滟,王怀民,史佩昌,等. 可信云服务[J]. 计算机学报,2015,38(1):133-149.

[2] ZHANG M,HONG C,CHEN C. Server Transparent Query Authentication of Outsourced Database [J]. Journal of Computer Research and Development,2010,47(1):182-190. (in Chinese)
张敏,洪澄,陈驰. 一种服务器透明的外包数据库查询验证方法[J]. 计算机研究与发展,2010,47(1):182-190.

[3] ARORA R,PARASHAR A. Secure User Data in Cloud Computing Using Encryption Algorithms [J]. International Journal of Engineering Research and Applications,2013,3(4):1922-1926.

[4] WANG Y D,YANG J H,XU C,et al. Survey on access control technologies for cloud computing[J]. Journal of Software,2015,26(5):1129-1150. (in Chinese)
王于丁,杨家海,徐聪,等. 云计算访问控制技术研究综述[J]. 软件学报,2015,26(5):1129-1150.

[5] HUANG R W,GUI X L,CHEN N J,et al. Encryption algorithm supporting relational calculations in cloud computing [J]. Journal of Software,2015,26(5):1181-1195. (in Chinese)
黄汝维,桂小林,陈宁江,等. 云计算环境中支持关系运算的加密算法[J]. 软件学报,2015,26(5):1181-1195.

[6] ZHOU X,LI T S,HUANG R W. Order-preserving Encryption Algorithm based on Random Interval in Cloud Environments [J]. Journal of Taiyuan University of Technology,2015,46(6):741-748. (in Chinese)
周雄,李陶深,黄汝维. 云环境下基于随机间隔的保序加密算法[J]. 太原理工大学学报,2015,46(6):741-748.

[7] CHATTERJEE A,SENGUPTA I. Searching and Sorting of Fully Homomorphic Encrypted Data on Cloud [C]// IACR Cryptology ePrint Archive. 2015:981.

[8] CHATTERJEE A,SENGUPTA I. Translating Algorithms to handle Fully Homomorphic Encrypted Data on the Cloud [C]// IEEE Transactions on Cloud Computing. 2015.

(下转第 167 页)