第 42 卷 第 7 期 计 算 机 工 程 2016 年 7 月
Vol. 42 No. 7 Computer Engineering July 2016

·安全技术· 文章编号:1000-3428(2016)07-0123-06 文献标志码:A 中图分类号:TP309.2

数据库中数值型数据的加密存储与查询方法

黄保华,王添晶,贾丰玮

(广西大学 计算机与电子信息学院,南宁 530004)

摘 要: 为提高数据库中数值型数据的密文查询效率,提出一种为数据库中数值型数据建立索引的新方法。对敏感数据应用单调递增函数计算得出比较值,应用非单调函数对敏感数据计算求得混淆值,将比较值和混淆值组成索引值并随密文数据一起存入数据库中。密文数据的查询采用两阶段方法,第一阶段查询时先根据查询条件计算得出比较值,将数据库中的索引值去混淆后与其进行比较,筛选出符合条件的密文结果集,第二阶段对一阶段得到的密文结果集解密后查询得到最终所需的结果集。实验结果表明,该方法能够提高数值型数据的密文查询效率。

关键词: 加密处理;两阶段查询;数值型数据;比较值;混淆值

中文引用格式: 黄保华,王添晶,贾丰玮. 数据库中数值型数据的加密存储与查询方法[J]. 计算机工程,2016,42(7):123-128.

英文引用格式: Huang Baohua,Wang Tianjing,Jia Fengwei. Encrypting Storage and Query Method for Numeric Data in Database[J]. Computer Engineering,2016,42(7):123-128.

Encrypting Storage and Query Method for Numeric Data in Database

HUANG Baohua,WANG Tianjing,JIA Fengwei

(School of Computer,Electronics and Information,Guangxi University,Nanning 530004,China)

【Abstract】 In order to improve the query efficiency of encrypted numeric data,this paper proposes a new method to build index on the numeric data in database. The monotonically increasing function is used to compute the comparison value from sensitive data,and the non-monotonic function is used to compute the confusion value from sensitive data. Then the comparison value and confusion value are connected together to form the index value to store in database. A two-phase method is used to query the encrypted data. In the first phase,the comparison value is computed according to the query condition. Then it is compared with the index value whose confusion part is removed to select out the encrypted result set that matches the query condition. In the second phase,the encrypted result set of the first phase is decrypted and queried to gain the final result set. Experimental results show that the proposed method can greatly improve the query efficiency of encrypted numeric data in database.

【Key words】 encryption processing;two-phrase query;numeric data;comparison value;confusion value

DOI: 10.3969/j.issn.1000-3428.2016.07.021

1 概述

近年来,云计算凭借自身的优势吸引了众多的企业关注。在云模型中,海量数据被存储在云端,能够有效地节省本地资源,同时按需付费使用的方式也为用户提供了许多便利[1],但由于数据是交由第三方云服务提供商进行管理的,因此敏感数据的安全性将难以得到保证[2-3]。为了能够削弱这个阻碍云服务发展因素的影响,目前企业普遍会对外包的敏感数据进行加密处理。加密存储的模式保证了数据的安全性,但随之而来也会使得一些原本简单的问题变得复杂化,例如数据的查询问题,传统加密算法加密后密文数据不再保持原有的明文顺序性,这就给查询操作带来了极大的困难[4]。如果没有相应的处理机制来应对在密文数据上的查询,那么当通过网络对数据库进行检索时,通常需要对整个属性列或整张表进行解密,这样数据库服务器和客户端将大量的时间、空间用在了无意义的解密和通信上[5]。这使得云端仅作为一个简单的数据存储池存在,无法发挥云计算的巨大优势。因此,寻找到一种

基金项目: 国家自然科学基金资助项目(61262072)。

作者简介: 黄保华(1973-),男,副教授、博士,主研方向为信息安全;王添晶、贾丰玮,硕士研究生。

收稿日期: 2015-07-13 **修回日期:** 2015-08-11 **E-mail:** bhhuang66@gxu.edu.cn

能够兼顾安全和效率的密文检索方法是数据库安全领域必需要研究的重点问题。

为解决以上问题,本文提出一个能够安全高效地在数值型数据的密文上进行检索的方案,以实现对数值型数据的各种查询请求,并且通过实验对其相关性能进行验证。

2 相关研究

目前,在数据库数值型数据的密文检索方面,国内外学者也做了一些相关研究。如文献[6]提出了一种桶划分的方法,把敏感属性的值域划分成范围值连续的几个区域,并为每个区域分配一个称为桶号的唯一标识符。当要对敏感属性进行查询时,则将查询语句转化为对属性值隶属的桶区间的查询。该种方法能够在第一阶段查询后过滤掉一部分不符合查询条件的记录,从而减少需要解密的数据量,提高对密文数据库的查询效率。这种索引方式会将符合条件的桶区间内的所有记录都返回给用户,用户再从返回的记录中进行进一步搜索[7],因此,当桶区间内的记录数目过多时这种方法的搜索效率就比较低。增加划分的桶数目可以提高搜索性能,但是有可能造成信息的泄露并且在多敏感属性的情况下会造成数据的冗余。为了优化桶划分模型下的密文查询问题,文献[8-9]对原始的查询语句进行了逻辑重组,重组后再按不同属性进行分桶[7],从而避免多敏感属性条件下应用桶划分造成的数据冗余过大的问题。文献[10]提出一种能够提高密文搜索效率的最佳桶划分算法,其基本思路是在按照安全需求确定好要划分的桶数目后,把求解原始桶划分的最优策略问题等价地转化为递归求解其 2 个子划分的最优策略问题,并记录其子划分的最小代价和划分路径,最后将子问题的最优结果合并,得到原始桶划分的最优策略[11]。同时文献[10]证明利用桶划分为多个敏感属性建立索引时,划分次数和划分的桶数目越多,查询的准确性就越高,但同时安全性也会越低,不过文中并没有就提高索引的安全性给出具体的解决方法。文献[12]提出一种保序加密(Order Preserving Encryption,OPES)算法,这种加密算法能够让密文保持对应明文的顺序大小,即若有 $P_i < P_j$,则 $C_i < C_j$(P_i,P_j 为明文,C_i,C_j 是其对应的密文),因此,可以支持对密文数据的最大值、最小值、范围查询,同时可以对密文数据执行 GROUP BY,ORDER BY 操作。但是由于密文数据的有序性,因此在明文空间较小的情况下,攻击者可以通过预先知道的一些明文-密文对根据密文的大小关系对明文数据进行猜测,因此这种保序加密算法只能适合一些对数据安全性要求不高的应用场景。由于同态加密算法能够直接对密文执行计算操作,因此近些年来也有许多学者致力于对它的研究,但同态加密算法目前的算法效率始终没有办法达到实际应用的要求[13-15]。

3 数值型数据的加密存储与查询

3.1 系统结构

本文选择在应用程序和数据库管理系统之间增加一层加/解密层来实现对数据的加密处理,这种方式具有很好的灵活性和平台适应性,它既无需修改内核结构复杂的数据库管理系统,也无需修改应用程序,因此,能够更好地根据用户需求来实现对数据的加密。数值型数据的加密存储与查询的系统结构如图 1 所示,其中,元数据中存储的是一些加密和转化 SQL 语句所需要的参数。当进行数据存储时,需要用这些参数来加密数据和生成对应的索引值。当进行数据查询时,需要应用这些参数将原始的 SQL 语句转化成为能够在密文数据上查询的 SQL 语句。临时表中存放的是第一阶段查询后得到的结果集解密后的结果。

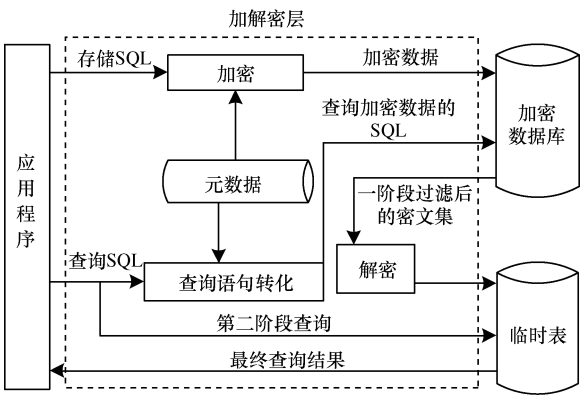

图 1 存储与查询结构

3.2 数值型数据的加密存储

3.2.1 存储模式

假设原关系模式为 $R(X_1,X_2,\cdots,X_r,\cdots,X_n)$,其中,$X_r$ 为需要进行加密的敏感数值型属性,X_1,X_2,$\cdots$,X_{r-1},X_{r+1},$\cdots$,X_n 为非敏感数值型属性,将敏感属性加密后,原关系模式将以 $R(X_1,X_2,\cdots,X_r^E,\cdots,X_n,X_r^s)$ 的形式存储到数据库中。X_r^E 是属性 X_r 经过加密处理后生成的密文字符串,即 $X_r^E=E(X_r)$,E 为使用的加密算法,X_r^s 是根据 X_r 构造的新属性,用来存储 X_r 的索引值,通过 X_r^s 来完成对加密属性 X_r^E 的查询。下面将对生成索引值的方法和存储过程展开详细论述。

3.2.2　加密存储算法

算法 1　比较值的生成

输入　敏感数值 *num*

输出　比较值

(1)确定敏感属性列的值域 G_{Xr},记值域的上限值为 *Max*。

(2)确定在敏感属性值域上单调递增的函数 $F(x)$,并使得 $F(Max)$ 的值不超过数据库中的数值数据类型的表示范围。

(3)应用确定好的单调递增函数 $F(x)$ 计算敏感数值 *num*,得到比较值的结果 $F(num)$。

算法 2　混淆值的生成

输入　敏感数值 *num*

输出　混淆值

(1)确定一个混淆函数 $Q(x)$,使得 $Q(x)$ 在敏感属性列的值域范围内是非单调的。

(2)应用混淆函数 $Q(x)$ 计算敏感数值 *num*,得到混淆值的结果 $Q(num)$。

算法 3　索引值的生成及存储

输入　比较值结果 $F(num)$,混淆值结果 $Q(num)$

输出　索引值、密文值

(1)比较值的结果 $F(num)$ 转化成 String 类型。

(2)混淆值的结果 $Q(num)$ 转化成 String 类型。

(3)由用户确定一个秘密的连接符,连接符必须只能由数字构成。

(4)将混淆值中和连接符相同的部分使用另一数字字符替代。

(5)在混淆值的最后加上连接字符,得到新的混淆值字符串 $Q'(x)$。

(6)将新的混淆值字符串 $Q'(x)$ 和比较值字符串 $F(x)$ 进行连接,得到索引字符串 $S = Q'(x) + F(x)$。

(7)将索引字符串存入数据库中。

(8)应用加密算法将敏感数值 *num* 加密,并将密文值存入数据库中。

3.3　加密数值型数据的查询

对于在数值型数据上的查询,主要有范围查询、等值查询、最大值查询、最小值查询,下面将分别论述如何通过上述生成的索引值来完成对数值型密文数据的范围查询、等值查询、最大值查询、最小值查询以及以范围查询和等值查询为基础的复合查询。

3.3.1　第一阶段查询

第一阶段查询具体如下:

(1)范围查询

常见的范围查询可以表示为 $column < num$,$column > num$(其中,*column* 为敏感属性的列名;*num* 为查询结果的界限值),下面将具体论述范围查询的具体过程。

进行范围查询的基本思路是将查询的界限值按照上面提及的算法 1 生成比较值 $F(num)$,然后将索引值列中的字符串数据去除掉混淆值后,转化成对应的比较值数据与 $F(num)$ 进行比较,从而得出结果。假设索引值存储在 X_r^s 列中,连接符记为 *c*,比较值的表示范围为 $numeric(n,m)$。

则当查询条件是 $column > num$ 时,可以将其转换成对索引列 X_r^s 的查询。

```
Where
cast(substring(X_r^s,(charindex(c,X_r^s,1)+len(c)),len(X_r^s))
asnumeric(n,m)) > = F(num)
```

当查询条件是 $column < num$ 时的处理方法也类似。

(2)等值查询

等值查询的查询语句可以表示为:

```
where
column = num
```

其中,*column* 为敏感属性的列名;*num* 为查询结果应满足的值。进行等值查询的基本思路是将 *num* 按照算法 1 生成计算值 $F(num)$,然后将索引值列中的字符串数据去掉混淆值后,转化成数值型数据后与 $F(num)$ 进行比较,筛选出与 $F(num)$ 相等的记录。设索引值存储在 X_r^s 列中,连接符记为 *c*,计算值的表示范围为 $numeric(n,m)$。

则当查询条件是 $column = num$ 时,将其转换成对索引列 X_r^s 的查询,语句如下:

```
Where
cast(substring(X_r^s,(charindex(c,X_r^s,1)+len(c)),len(X_r^s))
asnumeric(n,m)) = F(num)
```

(3)最大最小值查询

最大最小值查询的查询条件可以表示为:

```
where
column in(select max(column)from table),where column
in(select min(column)from table)
```

其中,*column* 为敏感属性的列名;*table* 为存储敏感属性的数据库表。进行最大最小值查询的基本思路是将索引值去混淆后,找出比较值最大和最小的索引项,然后将该索引项对应的数据记录完整输出。假设索引值存储在 *table*1 表的 X_r^s 列中,连接符记为 *c*,比较值的表示范围为 $numeric(n,m)$。

则当查询语句是:

```
where
column in(select max(column)from table)
```

可以将其转换成对索引列 X_r^s 的查询。

```
Select * from table1
Where
(cast(substring(X_r^s,charindex(c,X_r^s,1)+len(c),lenX_r^s))
as numeric(n,m)))in
(select
```

max(cast(substring(X_r^s,charindex(c,X_r^s,1)+len(c),
len(X_r^s)) as numeric(n,m))) from table1)

当查询语句是:

where
column in(select min(column)from table)

时也按照同样的方法对查询语句进行转化。

(4)复合查询

若查询条件为 $column \geqslant num$,那么可以将查询条件转变成为 $column > num$ or $column = num$,然后按照前面所述分别将范围查询 $column > num$ 和等值查询 $column = num$ 转化成针对索引值上的查询语句,若查询条件为 $column \leqslant num$ 也按照同样的方法进行转化。

若查询条件为 $num1 \leqslant column < num2$,那么可以将查询条件转变成为 $column < num2$ and $column \geqslant num1$,然后再按照上述方法将其转化成对索引值上的查询语句。

若查询条件为 $num1 \leqslant column \leqslant num2$ 或 $num1 < column \leqslant num2$ 或 $num1 < column < num2$ 的处理方法也类似。

3.3.2 第二阶段查询

将第一阶段查询得到的结果集进行解密,然后在结果集上执行原始的 SQL 查询语句,此时得到的结果集将是最终所需要的结果集。

4 算法分析

4.1 安全性分析

在本文方法中,索引值是由比较值和混淆值构成的,通过应用单调递增函数计算敏感数据得到比较值,应用非单调函数计算敏感数据得到混淆值,再应用数值连接符将比较值和混淆值合并成最终的索引值存入数据库表中。使用这种方法来建立的索引安全性较高,即使攻击者知道了索引的生成方法,并能够从数据库中得到索引值以及发现一些明文值和其所对应的索引值,但在不知道连接符以及所使用的单调函数的情况下也没办法从索引值中识别出比较值部分并由比较值反推出敏感数据的数值。并且由于敏感数值经过混淆函数的处理后的长度大小是不同的,因此连接符所处的位置也是不确定的,所以,攻击者在连接符数值和位置不确定的情况下,想要猜测出正确的连接符概率相当小,假设索引值的整数部分长度是 L,连接字符的长度上限是 K,那么想要正确猜测出连接符的概率可以表示为 $1/C_{L-2}^1 C_K^1$,退一步来说即使能够幸运地猜测出正确的连接符,也只能够通过选择明文攻击得出数值大小的分布情况,在没有知道具体的单调递增函数情况下,由比较值直接反推出敏感数据的数值基本上非常困难。

4.2 过滤效率分析

在本文所论述的方案中,对密文数据的过滤主要依靠于索引值中的比较值部分,而比较值部分是通过应用单调递增函数对敏感数据进行计算得到的,因此,比较值部分具有保序性,能够保持明文顺序的大小,这也是能够在索引上进行范围查询的基础。假设要查询大于某个数值的记录,那么先计算出该数值的比较值,然后将其与索引值中的比较值依次进行比较,找出大于等于该数值的比较值的记录。这是由于一般来说比较值的计算结果会涉及到小数位,而若截取的小数位数不够精确,那么只是仅仅找出大于该数值比较值的结果可能会漏掉一些原本符合条件的结果,但是这也会造成一定的误检率,也就是会把符合等于条件的记录集也返回给了用户。但是在实际情况中,一般来说这部分的记录所占的数量并不会很多,因此,误检率是非常低的。

4.3 存储空间分析

存储空间的大小取决于 2 个方面:(1)敏感数据的大小;(2)所采用的计算函数。假定产生索引值的计算函数已经确定,那么存储空间只取决于记录的数值。当记录中的数值大小不相同时,经过计算函数的运算后,产生的结果位数也不尽相同,因此,没办法确切地给出每条记录索引值所占用的存储空间,只能给出索引值所占用存储空间的上限值。假设敏感数据列中最大值为 a,a 经过单调递增函数的计算后得出的比较值位数为 m,在敏感数据列中某一值为 b,b 经过非单调函数计算后得到的混淆值最大,混淆值的位数为 n,令连接字符的位数为 k,那么索引值长度的上限为 $m+n+k$。则在存储了 N 条记录的数据库表中,存储索引值所使用空间大小的上限值为 $(m+n+k) \times N$。虽然添加索引字段会在一定程度上造成空间开销,但是却节省了解密大量数据的时间,随着现代扩容技术的不断进步,人们是能够接受以一定的空间开销来换取时间效率上的提高的。

5 仿真实验

进行实验的目的主要有 2 个:(1)验证本文提出的方法能够在密文的状态下,通过索引完成精确度较高的范围查询、等值查询;(2)与传统的全解密方法对比,观察运用本文方法后在时间性能上的提升效果。

依据 TPC-H 标准,实验使用 dbgen 工具自动生成了数据库,取比例因子为 0.1,即数据库大小为 100 MB。将其中的 orders 表作为实验数据库表,o_custkey 作为需要保密的属性,对其采用 AES 算法进行加密。实验系统采用 Java 语言进行编写,运行在 Windows7 环境

下,采用的数据库管理系统是 MSSQL2005。计算机的配置为 Intel CPU 2.40 GHz,内存 4.0 GB。

5.1　过滤性能测试

这部分的测试主要针对数值型数据上常用的范围查询和等值查询来进行,通过过滤效率来评估算法的查询性能,过滤效率越高,则该方法的性能越优越。计算过滤效率的公式如下,其中,n_1 是过滤后得到的记录数;n_2 是最终所需要的记录数;N 是数据库表中的总记录数。

$$glxl = \frac{N - n_1}{N - n_2}$$

实验表中共有 150 000 条记录,作为实验源的 o_custkey属性的数据类型为 int,值域为[1,14 999],计算比较值的函数为$\frac{x^2}{90}$,保留两位小数,计算混淆值的函数为$\frac{(x - 6\ 500)^2}{85}$,同时为了更方便地验证结果的正确性,在索引表和实验表 orders 中均添加了一列名为 ID 的新属性,使得同一 ID 值对应的分别是某条记录和某条记录生成的索引值。随机抽取几个数值进行范围查询的实验结果如表 1 所示,进行等值查询的实验结果如表 2 所示。

表 1　范围查询结果

o_custkey 属性取值	在索引表上查询所得结果的行数	在 orders 表上查询所得结果的行数	过滤效率/%
大于 10 000	49 944	49 932	99.98
大于 2 000	13 010	13 009	99.99
小于 5 000	50 161	50 148	99.98
小于 7 000	70 151	70 131	99.97

表 2　等值查询结果

o_custkey 属性取值	在索引表上查询所得结果的行数	在 orders 表上查询所得结果的行数	过滤效率/%
280	13	13	100
560	12	12	100
1 000	16	16	100
6 700	24	24	100
14 999	16	16	100

由表 1 可以看出,在进行范围查询时,在索引表上查询出的记录数目与在实验表 orders 表中直接查询得到的记录数目几乎一致,并且经过比对,在索引表上查询出来的结果集的 ID 值是直接从 orders 表查询得到结果集 ID 值的一个超集,由此可以知道,索引表中查询得出的结果集解密后几乎就是所需要的查询记录。

由表 2 可以看出,在进行等值查询时,在索引表上查询出的记录数目与在实验表 orders 表中直接查询得到的记录数目是一致的,并且经过比对,在索引表上查询出来的结果集的 ID 值与直接从 orders 表查询得到的结果集的 ID 值是一致的,由此可知,索引表中查询得出的结果集解密后正是所需要的查询记录。

综上所述,通过使用索引值查询的方法能够过滤掉大量不符合条件的记录,完成对数值型数据精确度较高的范围查询、等值查询,验证了实验的第一个目的。

5.2　时间性能测试

实验所用到的密文表中共存放 150 000 条记录,均采用 AES 算法进行加密。在明文上进行查询的时间相对于解密数据所需要的时间来说是非常小的,基本可以忽略不计,因此,查询时间主要由第一阶段查询后得到的密文记录数目决定,得到的密文记录越少,需要解密的数据量越少,则所需要的查询时间就越少。表 3 将使用索引后所花费的时间代价和全部解密该数据集后再进行查询所花费的时间代价进行对比,由上面所进行的查询性能测试可知,使用索引后第一阶段查询得到的密文集误检率达到 0,也就是说使用索引后的时间代价其实就是解密最终所需查询结果的时间,从表 3 可以看出,相较于传统的全解密再查询方法,使用索引后所需要的时间开销大大减少。

表 3　时间性能对比

o_custkey 属性取值	使用索引后的时间代价/ms	全解密的时间代价/ms	提高百分比/%
大于 10 000	371 084	1 114 903	66.7
大于 2 000	966 658	1 114 903	13.2
小于 5 000	372 696	1 114 903	66.5
小于 7 000	521 222	1 114 903	53.2
等于 280	97	1 114 903	99.9
等于 560	89	1 114 903	99.9
等于 1 000	119	1 114 903	99.9
等于 6 700	178	1 114 903	99.9
等于 14 999	119	1 114 903	99.9

6　结束语

针对数值型密文数据,本文提出一种新的索引方法。使用新方法建立的索引能够支持范围查询、精确查询、最大最小值查询,并且精确度较高,比现有的索引方法大大地减少了需要解密的数据量,节省了时间。并且通过向索引中添加混淆值,打乱了索引的保序性,攻击者即使在知道了某些明文密文对应关系的情况下,也无法推测出相应的明文信息

或数值分布信息。相比现有的索引方法而言,本文针对数值型数据提出的索引方法能够更好地兼顾数据的安全性和查询效率。本文主要针对数值型数据加密与查询问题进行研究,下一步将该方法扩展应用于数据库中,研究字符型等其他非数值型数据的加密与查询。

参考文献

[1] 林 闯,苏文博,孟 坤,等. 云计算安全:架构、机制与模型评价[J]. 计算机学报,2013,36(9):1765-1784.

[2] 冯登国,张 敏,张 妍,等. 云计算安全研究[J]. 软件学报,2011,22(1):71-83.

[3] Subaashini S,Kavitha V. A Survey on Security Issues in Service Delivery Models of Cloud Computing[J]. Journal of Network and Computer Applications,2011,34(1):1-11.

[4] 祝 钢. 数据库中密文检索优化模型仿真与研究[J]. 计算机仿真,2014,31(11):336-339.

[5] 马勺布,胡 磊,徐德启. 一种动态安全的密文数据库检索方法[J]. 计算机工程,2005,31(6):132-133.

[6] Hacifumus H,Lyer B,Li C. Executing SQL over Encrypted Data in the Database-server-provider Model[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York,USA:ACM Press,2002:216-227.

[7] 项 菲,刘川意,方滨兴,等. 云计算环境下密文搜索算法的研究[J]. 通信学报,2013,34(7):143-153.

[8] Hacigumus H,Lyer B,Sharad M. Efficient Execution of Aggregation Queries over Encrypted Relational Database[C]//Proceedings of DASFAA'04. Berlin,Germany:Springer-Verlag,2004:125-136.

[9] Hacigumus H,Lyer B,Sharad M. Query Optimization in Encrypted Database Systems[C]//Proceedings of Conference on Database Systems for Advanced Applications. Berlin,Germany:Springer-Verlag,2005:43-55.

[10] Hore B,Mehrotra S,Tsudik G. A Privacy-preserving Index for Range Queries[C]//Proceedings of the 30th VLDB Conference. New York,USA:ACM Press,2004:720-731.

[11] 王 迪,刘国华,于醒兵. 基于最佳桶划分策略的密文索引技术[J]. 小型微型计算机系统,2008,29(4):649-652.

[12] Agrawal R,Kiernan J,Srikant R. Order Preserving Encryption for Numeric Data[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York,USA:ACM Press,2004:563-574.

[13] Gentry C,Halevi S,Vaikuntanathan V. A Simple BGN-type Cryptosystem from LWE[C]//Proceedings of the 29th International Conference on Theory and Application of Cryptographic Techniques. Berlin,Germany:Springer-Verlag,2010:506-522.

[14] Gentry C,Halevi S,Smart N. Better Bootstrapping in Fully Homomorphic Encryption[C]//Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography. Berlin,Germany:Springer-Verlag,2012:1-16.

[15] Brakerski Z,Vaikuntanathan V. Efficient Fully Homomorphic Encryption from (Standard) LWE[C]//Proceedings of the 52nd Annual Symposium on Foundations of Computer Science. Washington D.C.,USA:IEEE Computer Society,2011:97-106.

编辑 刘 冰

(上接第 122 页)

[15] Wang Haodong,Li Qun. Distributed User Access Control in Sensor Networks[C]//Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems. San Franciso,USA:IEEE Press,2006:305-320.

[16] He Daojing,Chen Chun,Bu Jiajun,et al. Analysis and Improvement of a Secure and Efficient Handover Authentication for Wireless Networks[J]. IEEE Communications,2012,16(8):1270-1273.

[17] Rahman M,Khatib K E. Secure Time Synchronization for Wireless Sensor Networks Based on Bilinear Pairing Functions[J]. IEEE Transactions on Parallel and Distributed Systems,2010(19):14-25.

[18] Sun Kun,Ning Peng,Wang Chen,et al. TinySeRSync:Secure and Resilient Time Synchronization in Wireless Sensor Networks[C]//Proceedings of ACM Conference on Computer and Communications Security. New Yrok,USA:ACM Press,2006:264-277.

[19] 冯承天. 从求解多项式方程到阿贝尔不可能性定理:细说五次方程无求根公式[M]. 1 版. 上海:华东师范大学出版社,2014.

[20] Heinzelman W B,Chandrakasana A P,Balakrishnan H. An Application-specific Protocol Architecture for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless Communications,2002,1(4):660-670.

编辑 索书志