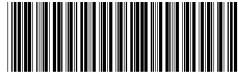


(19) 中华人民共和国国家知识产权局



(12) 发明专利申请

(10) 申请公布号 CN 102243629 A

(43) 申请公布日 2011.11.16

(21) 申请号 201010169778.7

(22) 申请日 2010.05.12

(71) 申请人 北京安华金和科技有限公司

地址 100081 北京市海淀区中关村南大街
11号2号楼201室

(72) 发明人 赵飞

(51) Int. Cl.

G06F 17/30(2006.01)

G06F 21/00(2006.01)

权利要求书 2 页 说明书 6 页

(54) 发明名称

一种基于多级视图和触发器的数据库透明加
解密方法

(57) 摘要

本发明用于对关系型数据库中的数据进行加
密保护,防止信息被非法的窃取,在关系型数据库
通用的视图和触发器的基础上,通过实现多级视
图并结合基于行标识的触发器实现对敏感数据的
自动加密和解密,达到对应用透明的目标,应用系
统无需改造;通过多级视图,实现对数据库查询
行为的精确判定,从而可以针对不同种类的查询
行为,构建专门的基于 LRU 缓存管理机制的密文
和明文数据缓存策略,构建对密文数据进行批量
预解密处理的策略,实现高效的密文查询。

1. 一种基于多级视图和触发器的数据库透明加解密方法,采用在密文表上建立三级视图的方法,包括如下步骤:

(1) 在密文表上直接构建第一级视图,除了对表上的所有字段进行检索外,增加了一个伪列 CALLTIMES,伪列 CALLTIMES 中保存一个伪列值 AI,所述伪列值 AI 为每个数据库操作会话,在每次执行本视图的时候,自动生成的一个唯一的值,用于表示一次新的查询操作的开始;

(2) 在第一级视图上构建第二级视图,除了对一级视图的所有字段进行检索外,增加一个伪列 TROWID,用于获取记录的行标识;在二级视图上构建 INSTEADOF 类型的触发器,其触发条件为 INSERT 和 UPDATE,在触发器中调用加密函数来完成对敏感数据的加密并保存到密文表相应的加密字段中,在执行 UPDATE 操作时,触发器通过行标识来对密文表数据进行更新;对于被加密的字段,在第二级视图中调用解密函数来完成解密,调用解密函数时将第一级视图中伪列 CALLTIMES 的值和密文字段的唯一标识 FIELDID (FIELDID 的具体值是在对明文数据进行预处理的过程中生成的),作为参数传递给解密函数;

(3) 在第二级视图上构建第三级视图,这个视图中包含密文表的所有字段,并且和密文表字段的顺序是一致的,由于在二级视图中完成了对加密字段数据的解密,因此这里将返回二级视图解密后的明文数据。

2. 根据权利要求 1 所述的数据库透明加解密方法,其特征在于:还包括步骤 4,将第三级视图的名称定义为用户操作的明文表的名称。

3. 一种对明文数据进行预处理的方法,包括:

(1) 改变敏感字段数据类型:其过程是先创建一个敏感字段数据的备份表,其结构是 {TROWID, F01, FN1, F02, FN2, ……F0n, FNn}, 其中 TROWID 是原表每行数据的行标识, F0i 是用于保存敏感字段明文数据, FNi 用于保存对明文数据加密后生成的密文数据。原表敏感字段的原始数据先备份到该表中 F0i 字段上。然后将原表敏感字段的数据全部 UPDATE 为 NULL,然后将该字段的类型修改为 VARCHAR2 类型,字段长度根据加密算法来确定加密后的数据最大长度来定义;

(2) 数据加密:对备份表中的全部明文数据进行加密,加密后的结果保存在对应的 FNi 字段上。全部成功后,将备份表中的密文数据按照 ROWID 更新原表敏感字段的全部记录,完成对敏感字段的数据加密,在数据加密期间对敏感字段上的 CHECK 约束禁用;

(3) 将原表改名,为表中的每个敏感字段分配一个唯一的值,这个值称为 FIELDID,并且将该 FIELDID 值保存在表中供使用。

4. 一种对建立三级视图和触发器的密文表的查询优化方法,包括判定执行计划、执行全表扫描优化处理、执行跳跃查询优化处理,其特征在于:所述判定执行计划为判定执行计划为全表扫描或为跳跃查询,包括如下步骤:

(1) 根据第二级视图中调用解密函数传入的 CALLTIMES 参数的值,确定是否是一次新查询的开始;如果不是,则进入步骤 2;如果是,按照数据库全表扫描方式读取数据的顺序,从表中密文字段读取前 N 条密文数据, $N \leq 200$, 并一次性批量的对密文数据进行解密处理,将明文和对应的密文数据保存在数组中;

(2) 对于每一条解密函数传入的待解密的密文数据,与步骤 1 中生成的数组中的密文进行比对,如果找到,则将命中统计值加 1,并将数组中对应的明文数据返回;如果没有找

到,则对密文数据解密;如果前M次操作累计的命中率超过K% (其中M<=N,K>=80),则判定为全表扫描并开始执行全表扫描优化处理;否则判定为跳跃查询并执行跳跃查询优化处理。

5. 根据权利要求4所述的查询优化方法,其特征在于:所述全表扫描优化处理包括:继续按照数据库全表扫描查询方式读取数据的顺序,从表中密文字段批量的读取前N1条密文数据,N1<=2000,并批量的进行解密处理,将明文和对应的密文数据保存在数组中,供比对;对于每一次解密函数传入的待解密的密文数据,直接对数组中的密文进行比对,并返回对应的明文数据;当在数组中没有找到符合的条目时,则继续顺序的批量读取和解密下一批的密文数据,依次重复执行,直到查询执行完毕。

6. 根据权利要求4所述的查询优化方法,其特征在于:所述跳跃查询优化处理包括:

- (1) 判断是否已建立热数据缓存,如果没有建立热数据缓存,则建立热数据缓存;
- (2) 在热数据缓存中查询是否存在对应的密文数据,如果没有找到,进入步骤3;如果有则直接返回对应的明文结果,并对该记录的“热度值”加1;
- (3) 对密文数据解密后,将明文和密文成对加入到热数据缓存中,并按照LRU换入换出算法对缓存中的数据进行换入换出的处理。

一种基于多级视图和触发器的数据库透明加解密方法

技术领域

[0001] 本发明涉及计算机数据安全领域,特别是涉及一种对关系型数据库中的数据进行加解密的方法。

背景技术

[0002] 随着计算机技术的飞速发展,数据库的应用已经十分广泛,深入到了各个领域。政府组织、商业机构和金融机构都是利用数据库服务器保存其重要的人事信息、贸易记录、市场决策性信息等各种敏感数据。这些数据的重要性不容置疑,它关系到国家的安全、企业的兴衰。因此,如何有效地保证数据库系统的安全,实现数据的保密性、完整性、有效性和可用性,已经成为业界人士研究的重要课题。目前,国内使用的主流商业数据库主要都是从国外进口,由于法律的限制,安全数据库系统基本不对中国出口,因此对现有主流商业数据库系统的数据加密、保护技术和密文数据的高效检索技术的需求非常强烈。

[0003] 现有技术中,一般采取基于数据库前置代理的加密保护方法,这种方法的缺点一是应用必须使用加密前置代理提供的 API,因此需要对现有程序进行改造,无法实现应用的透明;二是造成大量数据库产品的特性无法正常使用,并且这种“加解密前置”的方法对于在数据库内部执行的存储过程、函数都是无效的。

[0004] 近年来有采用基于数据库的视图和触发器实现透明加解密方法,这种方法目前采用的是单级视图,无法在没有主键的表上实现加密处理,无法做到真正的应用透明,并且对于存在复合主键的表,其数据更新性能将受到影响;同时不能准确的判断数据库的优化器如何处理任意一个查询操作请求的数据检索方式,造成无法进行有效的数据缓存和批量预解密处理,只能逐条的处理,极大的影响了查询的性能。

发明内容

[0005] 本发明的目的是实现加密解密操作的应用透明性,应用透明的范围主要包括:应用系统原先使用的各种开发接口 API 不用进行任何改变;原有的 SQL 语句和事务处理(ACID、读一致性等事务特性)设计不需要进行任何改变等。

[0006] 本发明基于数据库通用的视图和触发器机制实现对应用透明的支持,为此要解决的技术问题是:1) 实现对数据的加密操作,不需要依赖于原表的结构,从而实现不依赖于表结构限制的透明性。2) 能够让加解密程序模块准确的判断出数据库解析器和优化器对当前查询请求的处理方式,这些处理方式包括:全表扫描、密文索引扫描、跳跃扫描(跳跃扫描指的是根据非加密字段进行查询,查询的返回结果集包含了密文字段)等,从而能够在解密函数中,根据具体的处理方式进行专门的密文查询优化处理,提高查询的效率。

[0007] 本发明采用的技术方案是:一种基于多级视图和触发器的数据库透明加解密方法,采用在密文表上建立三级视图的方法,包括如下步骤:

[0008] 1) 在密文表上直接构建第一级视图,除了对表上的所有字段进行检索外,增加了一个伪列 CALLTIMES,伪列 CALLTIMES 中保存一个伪列值 AI,所述伪列值 AI 为每个数据库

操作会话,在每次执行本视图的时候,自动生成的一个唯一的值,用于表示一次新的查询操作的开始;

[0009] 2) 在第一级视图上构建第二级视图,除了对一级视图的所有字段进行检索外,增加一个伪列 TROWID,用于获取记录的行标识;在二级视图上构建 INSTEAD OF 类型的触发器,其触发条件为 INSERT 和 UPDATE,在触发器中调用加密函数来完成对敏感数据的加密并保存到密文表相应的加密字段中,在执行 UPDATE 操作时,触发器通过行标识来对密文表数据进行更新;对于被加密的字段,在第二级视图中调用解密函数来完成解密,调用解密函数时将第一级视图中伪列 CALLTIMES 的值和密文字段的唯一标识 FIELDID(FIELDID 的具体值是在对明文数据进行预处理的过程中生成的),作为参数传递给解密函数;

[0010] 3) 在第二级视图上构建第三级视图,这个视图中包含密文表的所有字段,并且和密文表字段的顺序是一致的,由于在二级视图中完成了对加密字段数据的解密,因此这里将返回二级视图解密后的明文数据。

[0011] 进一步,还包括步骤 4,将第三级视图的名称定义为用户操作的明文表的名称。

[0012] 进一步,为了对存在明文数据的数据表实现透明加密和解密功能,本发明提供一种对明文数据进行预处理的方法,包括:

[0013] 1) 改变敏感字段数据类型:其过程是先创建一个敏感字段数据的备份表,其结构是 {TROWID, F01, FN1, F02, FN2, F0n, FNn}, 其中 TROWID 是原表每行数据的行标识, F0i 是用于保存敏感字段明文数据, FNi 用于保存对明文数据加密后生成的密文数据。原表敏感字段的原始数据先备份到该表中 F0i 字段上。然后将原表敏感字段的数据全部 UPDATE 为 NULL, 然后将该字段的类型修改为 VARCHAR2 类型, 字段长度根据加密算法来确定加密后的数据最大长度来定义;

[0014] 2) 数据加密:对备份表中的全部明文数据进行加密,加密后的结果保存在对应的 FNi 字段上。全部成功后,将备份表中的密文数据按照 ROWID 更新原表敏感字段的全部记录,完成对敏感字段的数据加密,在数据加密期间对敏感字段上的 CHECK 约束禁用;

[0015] 3) 将原表改名,为表中的每个敏感字段分配一个唯一的值,这个值称为 FIELDID,并且将该 FIELDID 值保存在表中供使用。

[0016] 进一步,为了提高对已建立三级视图和触发器的密文表的查询效率,本发明提供一种查询优化方法,包括判定执行计划、执行全表扫描优化处理、执行跳跃查询优化处理,其特征在于:所述判定执行计划为判定执行计划为全表扫描或为跳跃查询,包括如下步骤:

[0017] 1) 根据第二级视图中调用解密函数传入的 CALLTIMES 参数的值,确定是否是一次新查询的开始;如果不是,则进入步骤 2;如果是,按照数据库全表扫描方式读取数据的顺序,从表中密文字段读取前 N 条密文数据,N <= 200,并一次性批量的对密文数据进行解密处理,将明文和对应的密文数据保存在数组中;

[0018] 2) 对于每一条解密函数传入的待解密的密文数据,与步骤 1 中生成的数组中的密文进行比对,如果找到,则将命中统计值加 1,并将数组中对应的明文数据返回;如果没有找到,则对密文数据解密;如果前 M 次操作累计的命中率超过 K% (其中 M <= N, K >= 80),则判定为全表扫描并开始执行全表扫描优化处理;否则判定为跳跃查询并执行跳跃查询优化处理。

[0019] 进一步,所述全表扫描优化处理,包括:继续按照数据库全表扫描查询方式读取数据的顺序,从表中密文字段批量的读取前N1条密文数据,N1<=2000,并批量的进行解密处理,将明文和对应的密文数据保存在数组中,供比对;对于每一次解密函数传入的待解密的密文数据,直接对数组中的密文进行比对,并返回对应的明文数据;当在数组中没有找到符合的条目时,则继续顺序的批量读取和解密下一批的密文数据,依次重复执行,直到查询执行完毕。

[0020] 进一步,所述跳跃查询优化处理,包括:

[0021] 1) 判断是否已建立热数据缓存,如果没有建立热数据缓存,则建立热数据缓存;

[0022] 2) 在热数据缓存中查询是否存在对应的密文数据,如果没有找到,进入步骤3;如果有则直接返回对应的明文结果,并对该记录的“热度值”加1。

[0023] 3) 对密文数据解密后,将明文和密文成对加入到热数据缓存中,并按照LRU换入换出算法对缓存中的数据进行换入换出的处理。

[0024] 本发明是在关系型数据库通用的视图和触发器的基础上,通过实现多级视图并结合基于行标识的触发器实现对敏感数据的自动加密和解密,达到对应用透明的目标,应用系统无需改造;通过多级视图,实现对数据库查询行为的精确判定,从而可以针对不同种类的查询行为,构建专门的基于LRU缓存管理机制的密文和明文数据缓存策略,构建对密文数据进行批量预解密处理的策略,实现高效的密文查询。本发明有益效果是:

[0025] 1、透明加密和解密

[0026] 通过采用多级视图,并在第二级视图中增加了行标识伪列,然后将视图触发器构建在第二级视图上,可以使触发器实现基于行标识对记录进行更新,而不再依赖于表中必须创建主键字段或唯一性约束字段,具有了更好的应用透明性。同时,在第三级视图只包含原表中的所有字段,则保证了查询解密的透明性。

[0027] 2、密文查询优化

[0028] 通过采用多级视图,并在第一级视图上增加了一个递增的、无重复的序列值字段,或者是时间戳字段,然后在第二级视图上将该字段的数据作为参数传递给解密函数,可以使解密函数能够判断何时开始了一次新的查询,然后可以分析出查询操作的类型是全表扫描还是跳跃查询,并相应的实现了面向全表扫描的批量预解密处理优化和面向跳跃查询的“热数据”缓存优化,有效减少解密处理的次数,极大的提升了在没有使用密文索引的情况下密文查询的性能。

具体实施方式

[0029] 本发明是在关系型数据库通用的视图和触发器的基础上,通过实现多级视图并结合基于行标识的触发器实现对敏感数据的自动加密和解密,达到对应用透明的目标,应用系统无需改造;通过多级视图,实现对数据库查询行为的精确判定,从而可以针对不同种类的查询行为,构建专门的基于LRU缓存管理机制的密文和明文数据缓存策略,构建对密文数据进行批量预解密处理的策略,实现高效的密文查询。

[0030] (一):敏感字段加密

[0031] 本发明方法中,对于每个需要对敏感字段进行加密的表:

[0032] 步骤1:改变敏感字段数据类型

[0033] 需要将表中敏感字段的数据类型修改为 VARCHAR2 类型, 字段长度根据加密算法来确定加密后的数据最大长度来定义。其过程是先创建一个敏感字段数据的备份表, 其结构是 {TROWID, F01, FN1, F02, FN2, F0n, FNn} , 其中 TROWID 是原表每行数据的行标识, F0i 是用于保存敏感字段明文数据, FNi 用于保存对明文数据加密后生成的密文数据。原表敏感字段的原始数据先备份到该表中 F0i 字段上。然后将原表敏感字段的数据全部 UPDATE 为 NULL, 然后将该字段的类型进行更改。

[0034] 步骤 2 :数据加密

[0035] 在上一步骤的基础上, 对备份表中的全部明文数据进行加密, 加密后的结果保存在对应的 FNi 字段上。全部成功后, 将备份表中的密文数据按照 ROWID 更新原表敏感字段的全部记录, 完成对敏感字段的数据加密。在数据加密期间, 需要对字段上的 CHECK 约束禁用。

[0036] 步骤 3 :将原表改名, 为表中的每个敏感字段分配一个唯一的值, 这个值称为 FIELDID, 并且将该 FIELDID 值保存在表中供使用。

[0037] 步骤 4 :创建多级视图

[0038] 视图是构建在数据库表之上的具有检索用途的虚拟表, 用于向请求者返回查询结果数据。在本发明中通过创建多级视图和视图触发器来实现透明加、解密。这里介绍的是采用了三级视图来实现的方法, 步骤如下 :

[0039] 1) :第一级视图是直接构建在密文表上的, 除了对表上的所有字段进行检索外, 增加了一个名为 CALLTIMES 的伪列 (虚拟列), 形式为 {ODC_FUNC_GETSEC() AS CALLTIMES, C1, C2, Cn-1, Cn} , 其中 C1 ~ Cn 是原表的字段。CALLTIMES 伪列可以是一个递增的、无重复的序列值, 或者是高精度的时间戳等, 其作用是对于每个数据库操作会话, 在每次执行本视图的时候, 会自动生成一个唯一的值, 用于表示一次新的查询操作的开始。

[0040] 2) :第二级视图是构建在第一级视图上的, 增加了一个行标识伪列, 用于获取记录的行标识 (例如 ORACLE 数据库的 ROWID), 此外, 对于被加密的字段, 在这个视图中将调用解密函数来完成解密, 并且将第一级视图中的 CALLTIMES 伪列的值作为参数传递到解密函数中, 形式为 :{ROWID AS TROWID, C1, C2, ... DecryptNoContext(Ci, FIELDID, CALLTIMES) AS Ci... Cn-1, Cn} ; 其中 DecryptNoContex 为解密函数, Ci 为敏感字段, FIELDID 为密文字段的唯一标识, CALLTIMES 为第一级视图中的伪列。

[0041] 3) :第三级视图是构建在第二级视图上的, 视图的名称和原表同名, 所有对加密前表的操作请求将被自动的施加到这个视图上。在这个视图中包含了原表的所有字段, 并且和原表字段的顺序是一致的, 形式为 {C1, C2, Cn-1, Cn} ; 由于在二级视图中完成了对加密字段数据的解密, 因此这里将返回二级视图解密后的明文数据, 从而实现了数据的透明解密。

[0042] 步骤 5 :创建基于行标识的触发器

[0043] 触发器是构建在前面的第二级视图上的 INSTEAD OF 类型的触发器, 其触发条件为 INSERT 和 UPDATE。由于在第二级视图上增加了行标识伪列, 因此对于 UPDATE 操作, 触发器可以通过该行标识来对表数据进行更新, 而不需要依赖于表上必须创建主键, 增强了透明性。

[0044] 步骤 6 :将步骤 1 中创建的备份表删除。

[0045] (二) :密文表查询和优化

[0046] 来自请求者的对密文表的查询将从第三级视图(与原表同名的视图)开始,并被依次的作用在前面创建的第二级和第一级视图上,在查询时会自动的执行视图中使用的解密函数对密文数据进行解密。在本发明方法中,解密的处理将根据数据库优化器和执行器根据不同的查询操作确定的查询方式(执行路径),来进行专门的处理;首先,数据库的优化器将根据对查询语句的解析结果和数据统计信息构建最优的执行计划。通常会产生三种类型的执行计划:

[0047] 第1种:通过定义的密文索引执行查询。

[0048] 第2种:通过表上的其他非敏感字段的索引执行查询(跳跃查询)。

[0049] 第3种:通过全表扫描执行查询

[0050] 对于第1种类型的查询,由于不属于本发明方法的范围,这里不做专门的说明。

[0051] 本发明方法涉及对第2、3种查询执行计划的处理,具体步骤如下:

[0052] 步骤1:判定执行计划

[0053] 在本发明方法中,解密函数首先需要精确的判断出查询的执行是按照前面的第2和第3种查询的哪一种来执行的,然后才能针对性的进行解密处理的优化。在本发明方法中判定执行计划的过程如下:

[0054] 1):根据前面创建的第二级视图中调用解密函数传入的CALLTYPES参数的值,确定是否是一次新查询的开始。如果不是,则进入2)。如果是,按照数据库全表扫描方式(FULL SCAN)读取数据的顺序,从表中密文字段(根据解密函数的FIELDID参数可以确定密文字段)读取前N条(N<=200)密文数据,并一次性批量的对密文数据进行解密处理,将明文和对应的密文数据保存在数组中。

[0055] 2):如果已经判定了执行计划为全表扫描,则进入步骤2;如果已经判定了执行计划为跳跃查询,则进入步骤3;如果还没有判定,则对于每一条解密函数传入的待解密的密文数据,与1)中生成的数组中的密文进行比对,如果找到,则将命中统计值加1,并将数组中对应的明文数据返回;如果没有找到,则对密文数据解密;如果前M(M<=N)次操作累计的命中率超过K%(K>=80),则判定为全表扫描并开始执行全表扫描优化处理;否则判定为跳跃查询并执行跳跃查询优化处理。

[0056] 步骤2:全表扫描优化处理

[0057] 这里采用的优化方法是一种批量预解密的方法,目标是减少解密处理的次数和代价,提高查询效率。

[0058] 具体方法是:继续按照数据库全表扫描查询方式读取数据的顺序,从表中密文字段批量的读取前N1条(N1<=2000)密文数据,并批量的进行解密处理,将明文和对应的密文数据保存在数组中,供比对。对于每一次解密函数传入的待解密的密文数据,直接对数组中的密文进行比对,并返回对应的明文数据。当在数组中没有找到符合的条目时,则继续顺序的批量读取和解密下一批的密文数据,依次重复执行,直到查询执行完毕。

[0059] 步骤4:跳跃查询优化处理

[0060] 对密文字段进行跳跃式的查询无法象全扫描查询那样进行批量的预处理,只能逐条的对传入的密文值进行处理。对于这种查询,可以认为被查询的记录中存在一定量的“热区数据”,也就是经常被使用的数据。为此,设计了一种“热区数据”缓存策略:按照LRU换

入换出算法,将常用的密文数据和对应的明文数据成对的缓存在数组中,形成热数据缓存,供查询使用,以减少逐条解密操作的次数,提高查询效率。具体过程如下:

[0061] 1) :首先在热数据缓存中查询是否存在对应的密文数据,如果没有找到,进入 2);如果有则直接返回对应的明文结果,并对该记录的“热度值”加 1。

[0062] 2) 对密文数据解密后,将明文和密文成对加入到热数据缓存中,并按照 LRU 换入换出算法对缓存中的数据进行换入换出的处理。