doi: 10.12068/j.issn.1005-3026.2018.12.004

面向多用户的多层嵌套数据库加密方案

周福才, 张鑫月, 曾 康, 秦诗悦
(东北大学 软件学院, 辽宁 沈阳 110169)

摘 要: 围绕外包数据的安全性问题与用户隐私性问题,展开对加密数据库方案的研究,提出了一个面向多用户的多层嵌套数据库加密方案. 该方案根据洋葱模型多层理论,采用多种不同类型的加密算法对用户的外包数据进行多层嵌套加密,实现了既保证数据机密性又满足多种不同SQL查询类型的数据库加密方案. 针对用户递交包含敏感信息的查询语句在一定程度上泄露用户自身的隐私这一问题,设计了基于单服务器私有信息检索(private information retrieval,PIR)技术的用户隐私保护机制,实现了用户匿名查询. 安全性分析表明,该方案满足数据机密性与用户隐私性. Sysbench基准测试实验分析表明,该方案具有良好的查询处理效率、读写吞吐量以及健壮性.

关 键 词: 外包数据;洋葱加密模型;数据库加密;PIR;隐私保护
中图分类号: TP 309.2 **文献标志码**: A **文章编号**: 1005-3026(2018)12-1691-06

Multi-layer Nested Database Encryption Scheme for Multiple Users

ZHOU Fu-cai, ZHANG Xin-yue, ZENG Kang, QIN Shi-yue
(School of Software, Northeastern University, Shenyang 110169, China. Corresponding author: ZHOU Fu-cai, E-mail: fczhou@mail.neu.edu.cn)

Abstract: Encrypted database scheme about outsourced data confidentiality and user privacy was researched, and a scheme of multi-layer nested database encryption for multiple users was proposed. A data encryption scheme was also proposed based on onion model to perform nested encryption on outsourced data by applying multi-layer theory with a variety of data encryption algorithms to guarantee data confidentiality and to work with SQL queries in different types. Meanwhile, user privacy protection scheme was proposed based on single server private information retrieval(PIR) to enable users to send SQL queries with sensitive data and protect the privacy of users when accessing the database, realizing user anonymous query. The security analysis results show that the scheme protects data confidentiality and user privacy. An evaluation result tested with a benchmark tool Sysbench demonstrates its well query processing efficiency, throughput and robustness.

Key words: outsourced data; onion encryption model; database encryption; private information retrieval(PIR); privacy protection

随着IT技术的发展,云计算作为一种新型计算模型越来越受到企业与个人的青睐. 云存储(cloud storage)是在云计算概念的基础上发展起来的一种新型存储方式[1]. 云存储主要指数据库服务提供商向用户提供数据存储以及查询等服务,也被称为DaaS(database as a service)模型[2]. 但由于用户的数据交由不可信或者半可信的第三方服务器,用户的隐私存在巨大的隐患,如2014年8月Apple公司iCloud门事件,2017年3月京东前员工涉50亿条公民信息泄露案[3]等. 云存储

收稿日期: 2017-09-20
基金项目: 国家自然科学基金资助项目(61772127,61472184); 国家科技重大专项(2013ZX03002006); 辽宁省科技攻关项目(2013217004); 中央高校基本科研业务费专项资金资助项目(N151704002); 辽宁省博士启动基金资助项目(20141012); 沈阳市科技基金资助项目(F14231108).
作者简介: 周福才(1964-),男,吉林长春人,东北大学教授,博士生导师.

数据的安全性主要包含数据的机密性与完整性. 本文主要针对机密性进行研究.

在保护数据机密性方面,目前主流解决方案是对数据进行加密. 在云存储中,数据库加密的主要问题为加密方案的安全性问题以及对数据查询处理的效率问题. 若采用对整个数据库加密、需要时下载下来、再在客户端进行解密查询的方法,将会带来巨大的传输带宽代价,耗费很长时间进行数据库的查询处理,以及低效率的数据库升级和备份. 而在用户本身隐私方面,在目前互联网环境中,很少有应用能够做到真正地保护用户隐私. 云存储用户本身隐私的主要问题体现在,用户的任何查询请求都在服务端的监控之下. 因此,在当前云存储环境下,针对用户数据的机密性以及用户本身的隐私,设计并实现一种既保证良好的安全性,又保证较高的数据库查询处理效率,同时保护用户本身隐私的数据库加密方案迫在眉睫.

针对上述问题,本文开展对加密数据库方案的研究,提出了一个面向多用户的多层嵌套数据库加密方案.

1 相关工作

关系型数据库在现有的环境中占据了主要市场,因此将其作为本文的研究对象. 1980 年,Wood 等首次对数据库加密技术进行研究[4]. 2000 年,Song 等首次提出了带关键字可检索的加密技术[5],可直接在密文上查询处理. Agrawal 等提出了一种保持数值顺序的数据库加密方案 OPES(order preserving encryption scheme)[6]. Amanatidis 等首次提出了一种可证明安全性的数据库库外加密模型[7]. 该模型能够针对不同的对象进行加密,加密粒度也有所区别,解决了应对敏感数据的加密问题.

刘念提出了适用于 DaaS 模型数据库的加密策略和密文检索模型,并针对 DaaS 模型的密钥管理问题提出了基于哈希查询的密钥管理策略[8]. Popa 等首次提出了一种数据库洋葱加密模型,针对不同的数据类型采用不同的加密模型,较好地保证数据库的机密性[9]. Hang 等在 Popa 的基础上,设计了 ENKI 系统,提出了一种属性值粒度的新型访问控制模型,保证了加密数据库的安全性与可用性[10]. Poddar 等提出一个强安全性的加密数据库方案 Arx,其具有等同 AES 的安全性,实现功能多样性,满足实际需求[11].

目前针对关系型数据库的加密方案大都需要设置其特有的密文检索策略,这对于处于运行状态的大型数据库而言,其转换代价无疑是非常昂贵的. 同时,其方案往往只保证数据的机密性,对用户自身的隐私保护却鲜有考虑.

2 预备知识

2.1 Paillier 同态加密

Paillier 同态加密[12]是一个基于决定性组合剩余类问题假设的公钥加密方案,它实现对密文数据的代数运算,由 Gen, Enc, Dec 三个算法组成:

1) Gen:选择公钥 $\mathrm{pk}=(n,g)$,其中 $n=pq$, $\gcd(pq,(p-1)(q-1))=1$,同时选择 $g\in\mathbf{Z}_{n^2}^*$,计算 $\lambda=\mathrm{lcm}(p-1,q-1)$, $\mu=\left(\frac{g^\lambda \bmod n^2-1}{n}\right)^{-1}\times \bmod n$,得到私钥 $\mathrm{sk}=(\lambda,\mu)$.

2) Enc:对于明文 $m<n$,选择一个 $r\in\mathbf{Z}_n$,计算密文为 $c=g^m\cdot r^n \bmod n^2$.

3) Dec:对于密文 $c<n^2$,计算明文为

$$m=\frac{c^\lambda \bmod n^2-1}{n}\cdot\mu=\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)}\bmod n.$$

2.2 保序对称加密算法

保序对称加密(order-preserving symmetric encryption,OPE)的密文保持其明文数值大小顺序. 本文采用文献[13]中的保序对称加密算法,通过构建平衡二叉树的方式构建 OPE 树,将加密后的结果 c 存储于节点中. 如图 1 所示,OPE 树的每个节点包含一个确定性的密文,用 0 表示左子树、1 表示右子树的方式标识节点密文的路径编码 path,并填充 10…0 至 32 bits 或者 64 bits,使与密文的 m 位数相同. OPE 编码形式为 OPE encoding = [path]10…0,同时该编码方式仍保留明文顺序.

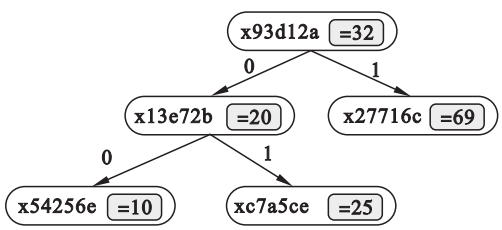

图 1 OPE 树实例

Fig. 1 OPE tree example

2.3 私有信息检索

私有信息检索(private information retrieval, PIR)是不经意传输协议(oblivious transfer protocol, OTP)的一个基本应用. 一般的私有信

息索引模型：将数据库存储的数据看作是一个 $n-$bit的字符串 $x=x_1x_2\cdots x_n$，它可由一个或多个服务器共同拥有，用户拥有查询索引 $i\in\{1,2,\cdots,n\}$，用户向数据库发送查询请求 $q(i)$，想要获取 x_i 的值，但不想让数据库知道自己对 x_i 感兴趣，又不能泄露 i 的信息；其私有性可以被定义为对任意两个索引 i 和 j，用户的查询请求 $q(i)$ 和 $q(j)$ 是不可区分的.

本文基于 Kushilevitz 等提出的计算性单服务器 PIR 协议[14]设计隐私保护查询机制.

3　模型与定义

3.1　模　型

本文提出了在一个不完全可信环境（假设数据库服务提供商为敌手）下面向多用户的多层嵌套数据库加密方案. 整体架构如图 2 所示.

该方案共有四方实体，分别是：应用客户端（application client，APP－C）、应用服务端（application server，APP－S）、代理服务端（proxy server，PS）、数据库服务端（database server，DS）.

APP－C：作为用户的应用客户端，主要扮演用户与服务端交互的角色，包括用户登录或注册、递交查询语句、进行 PIR 的客户端相关计算操作等.

APP－S：作为应用服务端以及数据的颁发方，主要承担服务端业务功能，负责与用户交互、用户身份验证、制定数据相关的访问权限、进行 PIR 的服务端相关计算操作等.

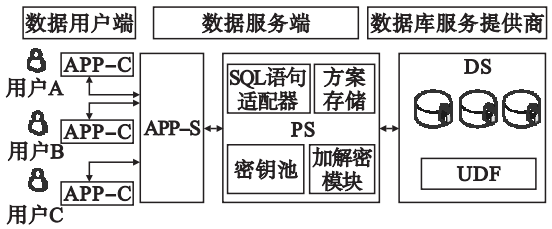

图 2　方案架构图

Fig. 2　System architecture diagram

PS：作为应用服务端与远程数据库服务端的连接器，主要起到代理转发服务器的作用，负责 SQL 语句的解析（SQL 语句适配器）、密钥生成（密钥池）、数据加解密（加解密模块）等.

DS：作为不完全可信的远程数据库服务端，主要提供数据存储服务，负责执行用户的查询语句、用户自定义函数（UDF）等.

方案执行过程主要包括初始化过程、加密存储过程、查询密态数据库过程、PIR 请求过程.

3.2　定　义

定义 1　方案主要由算法和协议组成：初始化算法、加密算法、查询协议、PIR 请求协议. 下面分别对其进行描述：

初始化算法：MK←setup(1^k) 是概率性算法，运行于客户端，用于生成主密钥 MK. 输入安全系数 1^k，输出 MK.

加密算法：ct←Enc(v,MK,t,c,o,l) 运行于代理服务端. 输入明文数据值 v、主密钥 MK、表名 t、列名 c、所属洋葱名 o 和所属层 l，输出密文 ct. 其中密文 ct 存放于数据库服务器.

查询协议：

$\mathrm{Query}(\mathrm{psd},q_i;\mathrm{EDB})=(\mathrm{Query_C}(\mathrm{psd},q_i),\mathrm{Query_S}(\mathrm{EDB}))$，由客户端与服务端交互执行. 协议中客户端输入查询语句 q_i 和用户密码 psd，服务端输入加密数据库 EDB. 协议完成时客户端返回查询结果 result，服务端返回加密的查询结果 Enc(result).

PIR 请求协议：

$\mathrm{PIR}(p(x_i),N;\mathrm{EDB})=(\mathrm{PIR_C}(p(x_i),N),\mathrm{PIR_S}(\mathrm{EDB}))$，由客户端与服务器交互执行. 协议中客户端输入 $p(x_i)$ 来请求第 i 个数据，以及困难集元素 N，服务端输入加密数据库 EDB. 协议完成时客户端返回查询结果 x_i，而服务端无法知道 x_i.

4　方案构建

4.1　初始化算法

setup(1^k)：用于初始化方案中主密钥. APP－C 客户端随机生成 k bit 私有的主密钥 MK，其由用户密码 psd 使用 AES 加密后，存储于 PS 端，对应用服务器和数据库服务器不可见.

4.2　加密算法

Enc(v,MK,t,c,o,l) 是基于洋葱加密模型的加密算法. 其中，洋葱加密模型对数据由内到外逐层加密，最内层为用户外包数据本身，次外层根据不同的 SQL 查询操作选择相应的保性加密（property preserving encryption，PPE）方案，最外层采用具有密文随机性（RND）的方案，防止基于频率的密码学攻击. 针对 SQL 查询语句种类的多样性，本文设计了 4 类洋葱加密模型，分别负责不同的用户查询操作：1）EQ 洋葱模型负责密文数据的等值查询，PPE 层采用 AES－ECB 与 ECJ 加密方案，RND 层采用 AES 的 CBC 工作模式；2）

ORD 洋葱模型负责范围查询,PPE 层采用 OPE 与 ECJ 加密方案,RND 层同采用 AES 的 CBC 模式;3)SRH 洋葱模型负责模糊查询,无 RND 层,PPE 层采用可搜索加密算法;4)FUN 洋葱模型针对数值型数据进行加密,无 RND 层,负责对密文数据的聚合查询,PPE 层采用 Paillier 同态加密算法.

加密存储架构如图 3 所示. 加密存储流程:

1) PS 端接收来自用户的数据表,以表的列为单位,对表进行属性级粒度的洋葱加密.

2) PS 端判断该表以及该列是否是新增的,若为新增则将该表名及其列名匿名化,并将该模式记录于本地记录表中. 记录模式为"用户: 数据表名散列值: 列名散列值: 洋葱名: 洋葱层数".

3) PS 端根据用户的主密钥 MK 生成其对应的加解密密钥 K,密钥 K 存放在主机的内存当中,随着用户的退出而从内存中删除,从而保证用户密钥的安全性. $K = \mathrm{AES}_{\mathrm{MK}}(t\_n, f\_n, o\_n, o\_s\_n)$,对(表名,属性名,洋葱模型名,洋葱层数)作 AES 加密,结果即为密钥 K.

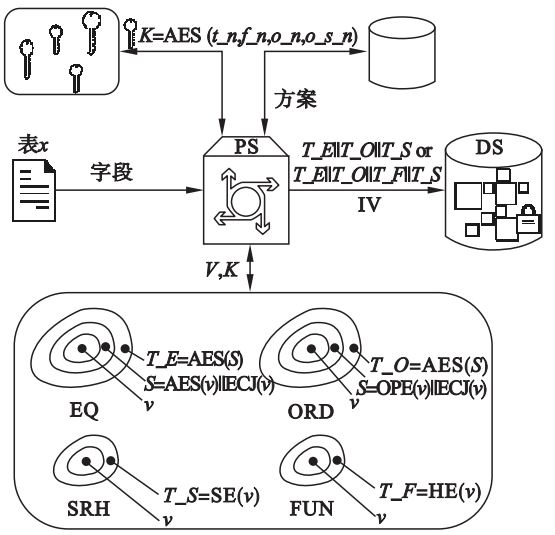

图 3 加密存储架构图

Fig. 3 Architecture diagram of encrypting and storing

4) PS 端根据 K 对数据 v 进行基于洋葱模型的加密. 首先判断 v 的类型:若为数值型,则进行 EQ/ORD/SRH/FUN 洋葱加密;若为字符型,则进行 EQ/ORD/SRH 洋葱加密. EQ 模型的第 1 层为 v;第 2 层为 $S = \mathrm{AES}(v) \parallel \mathrm{ECJ}(v)$;第 3 层为 $\mathrm{AES}(S)$. ORD,SRH 与 FUN 洋葱模型的加密模式与 EQ 模型一致,不再赘述.

5) PS 端将 v 的加密内容 $T\_E \parallel T\_O \parallel T\_S$(字符串型)或 $T\_E \parallel T\_O \parallel T\_S \parallel T\_F$(数值型)以及向量 IV 存储到 DS 模块,完成整个加密存储流程.

4.3 查询协议

$\mathrm{Query}(\mathrm{psd}, q_i; \mathrm{EDB}) = (\mathrm{Query}_{\mathrm{C}}(\mathrm{psd}, q_i), \mathrm{Query}_{\mathrm{S}}(\mathrm{EDB}))$是加密数据库的查询协议. 查询由用户发起,经由 PS 端重写 SQL 查询语句,提交给 DS 端进行基于密文数据的查询处理. 以 select 操作为例详细介绍协议内容:

1) 用户在 APP - C 发起查询 q_i,提交给 APP - S.

2) APP - S 将 q_i 和用户密码 psd 发送给 PS.

3) PS 接收 q_i,使用 psd 解密出 MK,并将 q_i 重写为 q_i':PS 对表名、查询字段作匿名化处理. PS 检查是否需要脱层处理:若需要,调用 UDF 模块进行脱层;若不需要,直接重写查询语句. 根据该列的运算类型,将列名重写为相应洋葱列名,对常量作相应洋葱加密. 对某些运算符进行转换,如 SUM 聚合、列加法运算需转换为同态加法. PS 将 q_i'递交给 DS.

4) DS 对密文数据作 q_i'查询处理,将密文结果返回给 PS.

5) PS 端使用密钥对查询结果进行解密得到最终的明文查询结果 result,并将结果发送给 APP - S.

6) APP - S 返回查询结果给 APP - C.

insert 操作与常规操作无异,逐层加密至对应洋葱的当前最高层后存储即可. delete 直接删除对应项即可,无需多余的操作. update 则分两种情况:第 1 种,设置列的值为常量值时,比如:ID = 3,过程与 insert 一致;第 2 种,根据现有列的值进行更新,比如:ID = ID + 1,则使用 HOM(ID + 1) = HOM(ID) × HOM(1)的同态加密,然而此时 OPE 与 DET 层均已失效,所以要更新该条目的所有洋葱,即先使用 select 语句得到旧值,由 PS 计算新值并加密,再用 update 语句提交新值.

4.4 PIR 请求协议

$\mathrm{PIR}(p(x_i), N; \mathrm{EDB}) = (\mathrm{PIR}_{\mathrm{C}}(p(x_i), N), \mathrm{PIR}_{\mathrm{S}}(\mathrm{DB}))$是 PIR 请求协议. 该协议中,应用服务器不知道用户具体的查询,故可保护用户隐私. 不失一般性,假设 APP - S 拥有数据库 x 并用矩阵 $\boldsymbol{M}_{s\times t}$表示,用户在 APP - C 上检索 x_i 数据并用 $\boldsymbol{M}[a,b]$表示. 协议如下:

1) APP - C 初始化生成 $k/2$ 位的素数 p_1 和 p_2,由 $N = p_1 \cdot p_2$ 得出 N 的值. 然后计算 PIR 客户端所需的 y 值:随机选取长度为 k 位的 t 个数,分别为 $y_1, \cdots, y_b, \cdots, y_t \in \mathbf{Z}_N^{+1}$,其中 y_b 为模 N 的二次

非剩余(QNR),其余的 $t-1$ 个数 $y_{j(j\neq b)}$ 为模 N 的二次剩余(QR),即 $Q_N(y_b)=1$, $Q_N(y_{j(j\neq b)})=0$. APP－C 再将 N 与 y 发送给 APP－S.

2) APP－S 向 PS 发起 PIR 条目预查询请求.

3) PS 将 PIR 检索的 SQL 查询语句发送给 DS.

4) DS 执行 SQL 语句查询,返回查询结果给 PS.

5) PS 对密文结果解密,恢复出明文信息,并发送给 APP－S.

6) APP－S 根据预查询结果计算出 PIR 服务端的 z 值:APP－S 对矩阵 $\boldsymbol{M}_{s\times t}$ 每一行 $r(1\leqslant r\leqslant s)$ 计算

$$w_{r,j}=\begin{cases}y_j^2, & \boldsymbol{M}[r,j]=0;\\ y_j, & \boldsymbol{M}[r,j]=1.\end{cases}\quad 1\leqslant r\leqslant s,1\leqslant j\leqslant t.$$

由二次剩余假设[14]可知,当 $j\neq b$ 时, $Q_N(w_{r,j})$ 的值恒等于 0;而当 $j=b$ 时,当且仅当 $\boldsymbol{M}[r,j]=0$, $Q_N(w_{r,j})$ 的值才为 0,其他情况下 $Q_N(w_{r,j})$ 的值必为 1. 因此可得

$$\boldsymbol{M}[r,b]=0\Leftrightarrow Q_N(w_{r,b})=0\ . \tag{1}$$

根据 $w_{r,j}$,计算

$$z_r=\prod_{j=1}^{t}w_{r,j}\ . \tag{2}$$

根据式(1)和(2),可以推出:

$$\boldsymbol{M}[r,b]=0\Leftrightarrow Q_N(z_r)=0\ . \tag{3}$$

7) APP－S 将 $z_1,\cdots,z_r,\cdots,z_s$ 返回给 APP－C,共计 $s\cdot k$ 比特位.

8) APP－C 获取用户的索引输入 $p(x_i)$,并根据 z 值计算出该索引对应的查询结果:APP－C 判断 z_a 是否是模 N 的二次剩余,由式(3)可知,若 $Q_N(z_a)=0$,则一定会有 $\boldsymbol{M}[a,b]=0$;反之,则必有 $\boldsymbol{M}[a,b]=1$.

至此,用户从 APP－C 端获得了 x_i 的值,且 APP－S 无法得知用户检索条目 i 的值.

5　安全性分析

本方案的加密算法是基于已证明安全加密算法的综合应用,满足数据的机密性;使用 PIR 协议,保护了用户的查询隐私. 由于篇幅限制,具体加密算法与 PIR 协议的安全性证明过程可参考文献[14－15],不再赘述.

6　性能分析

性能分析工具为 Sysbench,它是一个模块化、跨平台、多线程基准性能分析工具,主要用于评估各种不同系统参数下的数据库负载情况以及查询处理能力.

图 4 为 Sysbench 对原明文数据库以及本方案的性能测试结果对比. 由图可知,在单线程环境下,本方案的效率是平均每秒执行 13.59 次事务,即共约处理了 122.31 次 SQL 语句;同时,平均每秒的读写吞吐量达 244.66 次,表明对于常见的小型数据库表,本方案足够满足其日常的查询需求. 在处理异常方面,单线程下平均每秒仅发生 0.2 次,双线程下并未出现任何异常现象,表明本方案具有良好的健壮性. 通过性能分析,可以判断在应对小型数据库上,其查询处理效率、读写吞吐量以及健壮性等均达到了较为理想的效果.

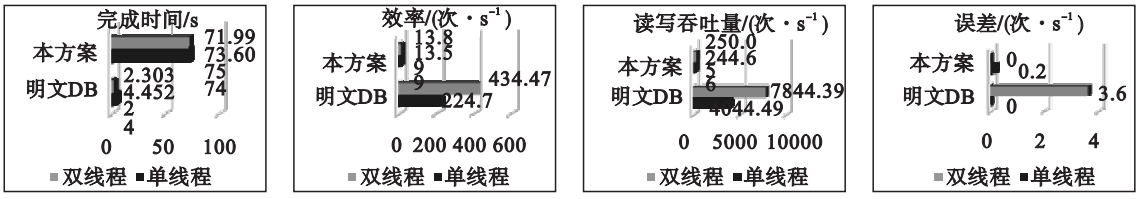

图 4　本方案与明文数据库性能对比

Fig. 4　Performance comparison between our scheme and original database

7　结　　论

针对常规的数据加密方案无法在安全性、效率以及功能满足上达成平衡,本文提出面向多用户的多层嵌套数据库加密方案. 首先,根据洋葱模型多层理论采用多种不同的加密算法对数据进行多层嵌套加密,保证了数据的机密性;其次,结合单服务器 PIR 计算方案,设计用户隐私保护机制,使用户得到其查询结果且 DBMS 无法得知具体查询内容,从而实现对用户隐私的细粒度保护;最后,利用 Sysbench 基准测试工具对方案进行了性能测试. 通过性能分析,验证了方案的正确性、可行性与健壮性,具有理论与实际应用价值.

参考文献:

[1] Zhang Y, Sun Y. Cloud storage management technology [C]// Proceedings of the 2nd International Conference on Information and Computing Science. Washington, DC, 2009:309 - 311.

[2] Hacigümüs H, Iyer B, Mehrotra S. Providing database as a service [C]// Proceedings of the 18th International Conference on Data Engineering. Washington, DC, 2002: 29 - 38.

[3] 新京报. 50 亿条公民信息泄露,京东前员工牵涉其中[EB/OL]. [2017 - 03 - 11]. http://www. chinanews. com/sh/2017/03 - 11/8171115. shtml.
(The Beijing News. 5 billion citizens' information disclosure, former staff of Jingdong involved in [EB/OL]. [2017 - 03 - 11]. http://www. chinanews. com/sh/2017/03 - 11/8171115. shtml.)

[4] Wood C, Fernandez E B, Summers R C. Database security: requirements, policies, and models [J]. *IBM Systems Journal*, 1980, 19(2):229 - 252.

[5] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]// Proceedings of the 2000 IEEE Symposium on Security and Privacy. Washington, DC, 2000:44 - 55.

[6] Agrawal R, Kiernan J, Srikant R, et al. Order preserving encryption for numeric data[C]// Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. New York, 2004:563 - 574.

[7] Amanatidis G, Boldyreva A, O'Neill A. Provably-secure schemes for basic query support in outsourced databases [C]// Proceedings of the 21st Annual IFIP WG 11. 3 Working Conference on Data and Applications Security. Redondo Beach, 2007:14 - 30.

[8] 刘念. DAS 模型中的数据库加密与密文检索研究[D]. 北京:北京邮电大学, 2010.
(Liu Nian. The study of database encryption and cipher text query in DAS[D]. Beijing: Beijing University of Posts and Telecommunications, 2010.)

[9] Popa R A, Redfield C, Zeldovich N, et al. CryptDB: protecting confidentiality with encrypted query processing [C]// Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. New York, 2011:85 - 100.

[10] Hang I, Kerschbaum F, Damiani E. ENKI: access control for encrypted query processing [C]// Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. Melbourne, 2015:183 - 196.

[11] Poddar R, Boeltert T, Popa R A. Arx: a strongly encrypted database system [EB/OL]. [2016 - 06 - 07]. https://eprint. iacr. org/2016/591. pdf.

[12] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]// Advances in Cryptology—EUROCRYPT'99. Prague, 1999:223 - 238.

[13] Popa R A, Li F H, Zeldovich N. An ideal-security protocol for order-preserving encoding[C]// Proceedings of the 2013 IEEE Symposium on Security and Privacy. Washington, DC, 2013:463 - 477.

[14] Kushilevitz E, Ostrovsky R. Replication is not needed: single database, computationally-private information retrieval[C]// Proceedings of the 38th Annual Symposium on Foundations of Computer Science. Washington, DC, 1997:364 - 373.

[15] Popa R A. Building practical systems that compute on encrypted data[D]. Cambridge, MA: MIT, 2014.