基于f-mOPE的数据库密文检索方案

周艺华 吉 文* 杨宇光
(北京工业大学信息学部 北京 100124)

摘 要：在云数据库环境下，为保证云存储数据的安全性，通常将数据加密存储。针对加密存储数据查询开销大，不支持密文排序，查询等缺点，该文提出一种 f-mOPE数据库密文检索方案。该方案基于可变保序编码(mOPE)，采用二叉排序树数据结构思想，生成明文一一对应的保序编码；基于AES加密方案将数据明文转化为密文存储；采用改进的部分同态加密算法提升保序加密方案的安全性。通过安全性分析及实验结果表明，该方案在保证数据隐私的基础上，不但能抵御统计型攻击，而且能够有效地降低服务器计算开销，提高数据库处理效率。

关键词：密文数据库；保序加密算法；可变保序编码
中图分类号：TP309 **文献标识码**：A **文章编号**：1009-5896(2019)08-1793-07
DOI: 10.11999/JEIT180805

Database Ciphertext Retrieval Scheme Based on f-mOPE

ZHOU Yihua JI Wen YANG Yuguang
(*Faculty of Information Technology, Beijing University of Technology, Beijing* 100124, *China*)

Abstract: In a cloud database environment, data is usually encrypted and stored to ensure the security of cloud storage data. To overcome the shortcomings of encrypting the data that the query overhead is big, the cipher text sortings and query are not support, etc, this paper puts forward a kind of f - mOPE cryptograph database retrieval scheme. Based on the mOPE sequential encryption algorithm, the idea of binary sort tree data structure is used to generate plaintext one-to-one corresponding sequential coding. Data plaintext is converted into ciphertext storage based on the AES encryption scheme. The improved partial homomorphic encryption algorithm is used to improve the security of sequential encryption scheme. The security analysis and experimental results show that this scheme can not only resist statistical attack, but also reduce effectively server computing cost and improve database processing efficiency on the basis of guaranteeing data privacy.

Key words: Ciphertext database; Sequential encryption algorithm; Mutable Order-Preserve Encoding(mOPE)

1 引言

随着“云”时代到来，越来越多个人和企业选择将数据上传到云端保存。为保证数据安全性，一般采用加密的方式，然后将加密后的密文存入云服务提供商的服务器中[1]。数据加密虽然可以成功解决云存储数据隐私安全问题，但造成的影响是用户使用效率的大幅度下降。因此，研究并提出既能保证数据安全性又能较好的提供高性能检索的云存储方案是当前云存储服务的的主要任务，具有重要的研究意义和应用价值。可搜索加密(searchable encryption)应运而生，得到了研究者广泛研究和发展[2–4]。

Agrawal等人[5]在2004年首次提出一种保序加密方案OPES(Order Preserving Encryption Scheme)，该方案把明文范围与加密后的密文范围进行映射，实现了在数据库环境下直接操作密文，该算法需要提前知道所有数据的值域，对于数据库环境下插入数据操作，仅仅是在理论上具备可行性。在OPES基础之上，为了从根本上提高OPE的效率和安全性，进行了大量的研究[6–10]。文献[6]针对保序加密提出一种新的安全性标准IND-OCPA (INDistinguishability under Ordered Chosen-Plaintext Attack)，并且基于伪随机分组密码提出一种符合IND-OCPA标准的保序加密方法。文献[11,12]提出一种非线性保序加密方案，其主要思想是对每一个明文建立一个密文索引，$y=ax+b$。但是该方案非线性函数并不具备安全性，攻击者只需要知道两对明文和密文即能够破解线性参数a和

收稿日期：2018-08-16；改回日期：2019-01-29；网络出版：2019-02-21
*通信作者： 吉文 jwnba24@163.com
基金项目：国家自然科学基金(61572053)
Foundation Item: The National Natural Science Foundation of China (61572053)

b，无法防止统计攻击，安全性不高。文献[13]提出2次噪声的概念，能够很好地保证明文顺序信息，同时能够防止选择明文攻击，其缺点是在大数据量情况下运算开销极大。2013年Popa等人[14,15]设计了一个CryptDB密文数据库，并提出一种新型保序加密方案mOPE，其实现了IND-OCPA的理想安全目标，除了明文顺序信息以外，mOPE不会暴露明文的其它信息，安全性很高。然而该方法需要$O(\lg n)$轮通信，随着n的增大，mOPE算法执行效率降低，服务器开销增大。所以在该研究基础之上，本文提出一个改进的mOPE方案 f-mOPE，降低服务器开销，提高对密文增删改查的效率。

2 f-mOPE保序加密方案

f-mOPE保序加密方案参考mOPE保序加密方案的部分思想，采用平衡二叉树作为保序编码的数据结构。在此基础上，本方案提出分块生成加密索引，降低服务端计算性能，提升效率；同时，在方案中引入混淆算法以及同态加密算法，提升整个方案的安全性。

2.1 mOPE方案分析

mOPE通过引入二叉搜索树，将明文顺序转换为保序编码，与密文数据一起存入密文数据库；而对于明文数据，可直接采用对称加密(如AES等对称加密算法)对明文加密后将其存储在数据库中。经过分析该算法执行性能有待提升，其性能损耗主要体现在以下3个方面：

(1)插入和查找元素时，客户端与服务器端的交互次数为$\lg(n)$，其中n是元素的总个数；

(2)服务器端维护的平衡二叉树的重排序操作次数也随着n的增大而增大；

(3)虽然Popa提出mOPE保序加密方案能达到IND-CPA安全性，只会泄露明文的顺序关系，但是相同的元素对应的顺序编码相同，攻击者通过统计攻击能够得到明密文的对应关系，存在一定的安全隐患。

2.2 数据模糊处理

mOPE方案中，明文相同的元素其对应的保序编码一定相同，此种方案存在统计攻击。为保证安全性，提出对明文以及保序编码进行模糊化处理。针对明文索引，采用非线性函数，使同一明文得到不同保序编码结果，非线性函数表达式如式(1)

$$y = a \times x + b + \text{noise},\quad (\text{noise} \in [0, a \times \text{sens}], |x_1 - x_2| \geq \text{sens}) \tag{1}$$

在保序编码构造过程中，无法根据保序编码判断哪些元素相同，可以有效地防止统计攻击；采用AES加密算法的CBC模式对明文进行加密，使得同一明文得到的密文不同，同样混淆明文数据之间大小关系，提高安全性。本文分别针对明文数据、明文索引采用不同的混淆算法，该方案能够有效提升整个保序编码方案的安全性。

2.3 数据分块处理

mOPE方案中，通过1棵平衡二叉树来维护所有数据的保序编码，平衡二叉树重平衡操作最坏需要遍历从叶子节点至根节点，客户端与服务端交互次数最多为$\lg(n)$(n为平衡二叉树高度)。本文采用化繁为简的思想，以多棵高度固定的平衡二叉树作为维护保序编码的基础。每一次操作，只需调整对应子树节点，无需调整全部节点。保密数据分块处理模型如图1所示。

为提升算法整体安全性，子树编号生成采用改进的someWhat部分同态方案[16]，通过比较二叉树编号确定数据的密文索引。基于最大公约数问题(GCD)，选用someWhat部分同态加密方案，模型如下：

文中使用的符号如表1所示，其中，$\omega(\cdot)$是高阶无穷大量。

(1)KenGen(λ)：根据安全参数λ生成私钥p和大整数q, r为加密过程中随机选取的整数，p和q在密钥生成阶段产生。加密算法可表示为

$$c = m + 2^n r + pq \tag{2}$$

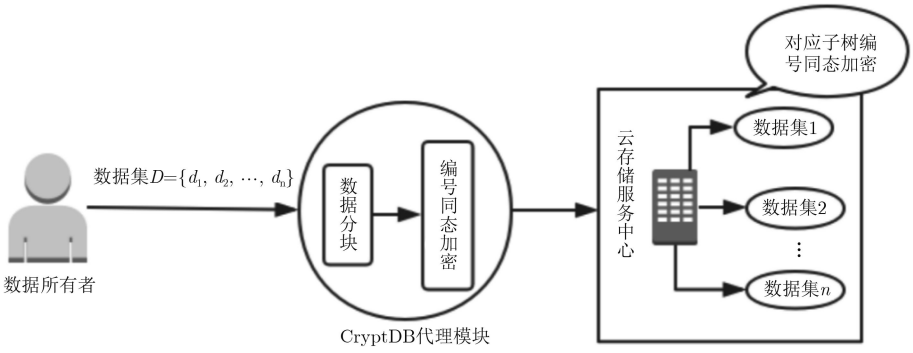

图 1 保密数据分块处理方案

表 1　公式符号说明

λ：安全参数;
ρ：噪声长度,为抵抗暴力攻击$\rho=\omega(\lg\lambda)$;
η：私钥二进制长度，η满足$\eta\geq\rho\Theta(\lambda\lg^2\lambda)$，这样才能保证压缩解密可行;
γ：公钥二进制长度，为抵抗格攻击，$\gamma=\omega(\eta^2\lg\lambda)$;
τ：公钥个数，$\tau\geq\gamma+\omega(\lg\lambda)$，文中需要的公钥个数为$2\sqrt{\tau}$;

文献[16]中将pq当做公钥，私钥p存在泄漏风险。所以，本文改为引入最大公约数问题，生成由$2\sqrt{\tau}$个0加密生成的密文：$b\in\{0,1\}$, $1\leq i\leq\sqrt{\tau}$, $x_{i,b}=pq_{i,b}+r_{i,b}$，最终公钥尺寸为 $2\sqrt{\tau}$, $\mathbf{pk}=<x_0, x_{1,0}, x_{1,1}, x_{2,0},\cdots, x_{\sqrt{\tau},1}>$。

(2)加密Encrypt($\mathbf{pk}$, m)：τ维向量$\boldsymbol{b}=<b_{i,j}>$ $(1\leq i,j\leq\sqrt{\tau}, b\in\{0,1\})$, 明文$m\in\{00,01,10,11\}$, 密文$c$为

$$c=m+2^n r+2^n\sum_{1\leq i,j\leq\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}(p\in[2^{\eta-1},2^{\eta}]) \quad (3)$$

(3)解密Decrypt(sk, c):

$$\begin{aligned}&(c \mod p) \mod 2^n\\ &=\left(\left(m+2^n r\sum_{i\leq i,j\leq\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}\right)\mod p\right)\mod 2^n\\ &=(m+2^n r) \mod 2^n=m\end{aligned} \quad (4)$$

(4)密文检索Search(sk, c)：设c_1为检索密文，c_2为待检索密文

$$\begin{aligned}R&=c_1-c_2=m_1-m_2+2^n(r_1-r_2)\\ &+2^n\left(\sum_{1\leq i,j\leq\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}-\sum_{1\leq s,j\leq\sqrt{\tau}} b_{s,\tau}x_{s,0}x_{\tau,1}\right)\end{aligned} \quad (5)$$

$$\begin{aligned}&((c_1-c_2) \mod p) \mod 2^n\\ &=\left(\left(m_1-m_2+2^n(r_1-r_2)+2^n\left(\sum_{1\leq i,j\leq\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}\right.\right.\right.\\ &\left.\left.\left.-\sum_{1\leq s,j\leq\sqrt{\tau}} b_{s,\tau}x_{s,0}x_{\tau,1}\right)\right)\mod p\right)\mod 2^n\\ &=(m_1-m_2+2^n(r_1-r_2)\mod p)\mod 2^n\\ &=m_1-m_2\end{aligned} \quad (6)$$

如果$m_1>m_2$，则$R>0$，如果$m_1=m_2$，则$R=0$，如果$m_1<m_2$，则$R<0$。根据R值，即可在加密情况下判断两个明文的大小关系，无需解密。

本文采用上述同态加密方法加密二叉子树的编号，使得攻击者无法通过分析二叉子树的编号得到所有的数据的大小关系。

2.4　保序加密方案构造

上一节介绍数据分块方案，使用多棵二叉树存储不同的区间的数据。按照mOPE保序加密方案，设i为保序编码，p为二叉树根节点至叶子节点路径，no表示二叉树编号。通用的编码公式定义为

$$i=[p]10\cdots0 \quad (7)$$

由于本文中引入分块子树的概念，所以改进保序编码公式定义为

$$i=[\text{no}][p]10\cdots0 \quad (8)$$

假设保序编码的最大长度为32位，使用22位二进制表示no，10位表示1棵树中所有元素的保序编码位数，所有子树构成1个森林，一共可以存储$2^{22}\times2\times10=2^{32}$个元素，若数据集数量大于该值，需要扩大保序编码位数。

本文采用广义二叉平衡树存储保序编码结构，平衡因子为k，以k作为二叉树是否需要重平衡的条件。当使用平衡因子k时，重平衡触发条件为$h>h_{\min}+k(k>1)$，设$h_{\min}$为最小高度。采用扩大平衡因子的方法可以有效减少重平衡次数，降低由多次重平衡操作带来的开销。当采用扩大平衡因子的方法重平衡操作得到完全平衡二叉树，可以通过数据排序的序号快速生成数据的保序编码。序号与保序编码的关系如表2所示(以15个节点为例)。通过分析序号与保序编码关系本文提出表3算法1，引入临时表参与二叉树重平衡操作。

该调整涉及子树所有节点，所以调整操作的时间复杂度为$O(n)$。

用户执行插入或检索数据算法如表4算法2所示。由于该方案检索复杂度仅与子树高度相关，时间复杂度最大为$\lg(N)$(N为子树最大高度)，由于N为固定值，所以查询及插入开销，并且不会随着数据量的增大而增大。

3　安全性分析

选择明文顺序攻击模型：假设此时攻击者有两组数据集$D1$, $D2$，两个数据集具备同样的顺序关

表 2　序号与保序编码对应关系表

序号	1	2		3		4		5		6		7		8	
保序	[000]	[00]	[001]	[0]	[010]	[01]	[011]	[1]	[100]	[10]	[1]	[1]	[110]	[11]	[1111]
编码	1=1	10=2	1=3	100=4	1=5	10=6	1=7	000=8	1=9	10=10	011=11	100=12	1=13	10=14	=15

表 3 算法1: 保序编码调整算法

符号定义：ord_num：数据的序号；index：数据十进制编码

```
//将所有的数据排序后存入临时表tmp中
insertIntoTmpTable(datas)
h = lg(n)+1;
index = 2(n-(2^h-1 -1))+1;
count = index-1;
//更新临时表中数据索引编码
if(ord_num > count):
    foreach():
      updateTmpTable(ord_num):
        index = ord_num + (ord_num-count);
    update tmp set index = index where ord_num=ord_num;
else:
    foreach():
      update tmp set index = ord_num where ord_num=ord_num;
//将临时表重平衡结果更新至数据表中，需要将临时表中index转换为二进制并加入子树标识，如式(7)描述
foreach(data):
    update OPE_Table A inner join tmp B on A.ciper = B.ciper set A.ord_code = B.index;
```

表 4 算法2: 数据插入及检索算法

```
插入元素算法：
key,IV = generateInitAttr();//初始化加密参数
//加密明文
ciphertext = encryptData(plainText);
//构建保序编码
foreach(plainTexts):
    //确定子树编码
    treeIndex = partitionTree(plainText);
    //数据模糊化处理
    fuzzyData=FuzzyData(plainText);
    //与服务端交互确定数据保序索引
    code_index=connectToServer(fuzzyData,treeIndex);
    //插入数据
    insertData(fuzzyData,code_index);
```

```
查找元素算法：
//确定子树编码
treeIndex = partitionTree(plainText);
//查询子树根节点
rootNode = searchTreeRootNode(treeIndex);
//遍历子树寻找所有符合条件密文
datalist = search(rootNode,tree,plainText):
foreach(datalist)://解密所有密文
    data = decrypt(value,key);
return datalist;
```

系，攻击者将两个数据集发送至客户端并操纵客户端对其中一个数据集执行OPE算法，如果攻击者通过加密结果能够判断客户端加密了哪一组数据，则它不满足IND-OCPA。因为f-mOPE与mOPE一样，除了明文的顺序信息，插入路径外不会暴露明文的其他信息，两组数据加密后插入路径信息，顺序信息完全一致，攻击者无法判断哪一组数据加密。所以，该方案符合IND-OCPA标准。

统计攻击：攻击者已知明文的大小以及各个明文的数量，将这些明文加密后存入数据库，攻击者能够知道所有明文的保序编码，统计保序编码相同的数量，与明文相同的数量对比，可以知道明文与编码的对应关系。统计攻击发生在明文与保序编码，密文一一对应，且不会改变的情况下，而f-mOPE方案中，明文加密采用的是AES CBC模式，一个明文可能对应多个密文，保序编码可能随着二叉树的重平衡而改变。因此，能够有效地防止统计攻击。

改进someWhat部分同态加密方案安全性及性能分析：本方案所依靠的数学难题是整数近似 GCD 难题(approximate-GCD problem)以及稀疏子集和难题。

本文基于近似最大公约数问题，构造了一个部分同态加密算法加密二叉树编号，若攻击者可以破解文的方案，则攻击者可以由$x_{i,b}=pq_{i,b}+r_{i,b}(1\le i\le\sqrt{\tau})$求出私钥$p$，从而分析出各个子树存放数据大小关系。而实际上最大公约数问题到目前为止是不能被解决的，所以，本文方案是安全的。改进的部分同态加密方案与DGHV部分同态加密方案对比如表5所示。

表 5　DGHV部分同态加密方案与本文改进方案对比

	DGHV部分同态加密方案	本文改进部分同态加密方案
加密效率	1次加密1 bit明文	1次加密$n$bit明文
安全性	q是对外开放的，那么如果 pq作为公钥，很容易计算出私钥p的值。	加入一些明文为0加密得到的密文$\{x_i: x_i=2^n r_i+pq_i\}$，将这些密文组成一个集合$s$，以$\sum_{1\le i,j\le\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}$作为公钥，任意选取集合元素$x_i\in S$加入运算，因为其明文都是0，不会改变加密结果，并且能够提高算法安全性。在运算过程中只需要将2^n上传到服务器即可，把$2^n\sum_{1\le i,j\le\sqrt{\tau}} b_{i,j}x_{i,0}x_{j,1}$作为公钥，即使获取$2^n$也无法获取密钥$p$
复杂度	该方案公钥尺寸约为$O(\lambda^{10})$	该方案中，参数取$\rho=\lambda,\eta=O(\lambda^2),\gamma=O(\lambda^5),\tau=O(\lambda^3)$，所以该部分同态加密公钥尺寸为$r+\tau(\lambda+\eta)=O(\lambda^5)$

4　实验结果与分析

mOPE保序加密方案的资源开销主要体现在以下方面：(1)插入、查询数据时客户端与服务器端的多次交互带来的通信开销；(2)服务器端维护的平衡二叉查找树重平衡操作的开销。本节主要从以上两个方面进行实验。

实验环境为：Windows操作系统，8GB运行内存，Inter Core i5; java开发环境为JDK1.8，使用Mysql数据库；使用IntelliJ IDEA作为编码环境。

实验1　数据检索时间对比。

以查找服务器端维护的平衡二叉查找树的叶子节点为例(查找叶子节点为最极端情况，客户端与服务器端交互次数最多)，分析在不同数据量情况下两种方案查询的通信开销如表6所示。

本实验数据集为从0开始的递增数列，所有数据插入完成后，服务器端维护的平衡二叉查找树的最左叶子节点应该为0，所以查询0开销最大。mOPE方案中，客户端与服务器端交互次数为$\lg(n)$，与数据总数量n有关，呈对数增长；f-mOPE由于采用的是划分子树的方式，子树节点数固定，交互次数不会随n的变化而变化。

表 6　mOPE与f-mOPE查询开销对比

数据个数	mOPE查询开销	f-mOPE查询开销
500	8	10
5000	11	10
10000	12	10
20000	13	10
50000	15	10

实验2　比较mOPE与f-mOPE方案中总数据量对平衡二叉查询树重平衡次数的影响，对比重平衡操作的时间消耗。

图2本文采用mOPE方案分析平衡因子k对重平衡次数影响，可以看出，当重排因子增大的时候，服务器端维护的二叉平衡查找树重平衡操作发生次数显著减少；图3是mOPE方案和f-mOPE方案插入元素的时间消耗对比，可以看出f-mOPE方案能够显著降低数据插入时间。采用平衡因子$k=2$对比

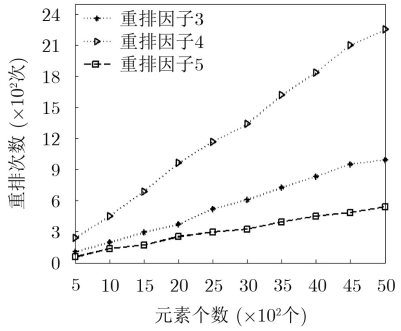

图 2 平衡因子k对重平衡次数影响图

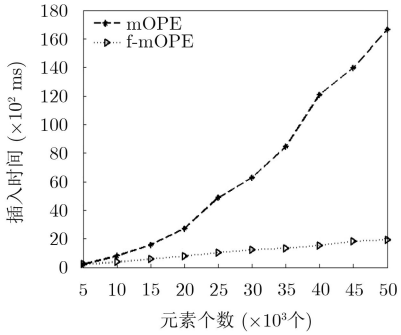

图 3 mOPE方案和f-mOPE方案插入元素时间消耗对比

mOPE与f-mOPE两种方案中的重平衡次数，如图4所示，随机一次性插入数据个数区间为$[5\times10^4, 50\times10^4]$, f-mOPE方案中触发重平衡操作频率约为0.23%，而mOPE方案触发重平衡操作频率约为29.4%，实验结果表明该方案能够有效地降低重平衡操作，从而提升插入元素效率。

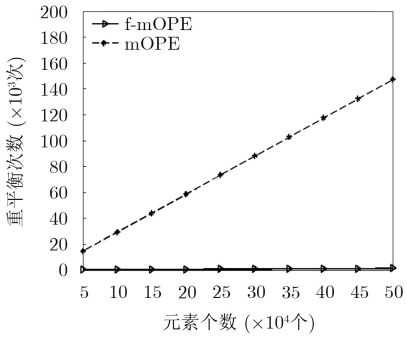

图 4 元素个数与重平衡次数关系

实验3 检索效率分析。

本文引入同态加密算法提高整个保序加密方案的安全性。因此本实验对比引入同态加密算法后的方案的检索效率与未引入同态加密算法的方案的检索效率，如图5所示。

从总体上分析，使用同态加密方法后，整体的检索执行时间大约增加20%，并不会影响整体的检索效率。

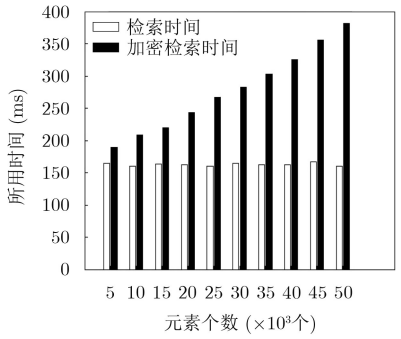

图 5 检索执行时间对比

实验4 时间复杂度分析。

表7为mOPE与f-mOPE时间复杂度比较，其中需要注意的是mOPE需要关注的是全体数据集，而f-mOPE仅需要关注某个区间段的数据集即可，数据规模相差巨大。

5 结论

本文在数据库保序加密场景下，提出f-mOPE密文检索方案。本文方案引入分块治理的思想，将服务器端维护的二叉查找树从1棵变为多棵子树，每棵子树维护相应数据段数据的索引大小关系，在降低整个保序加密方案的通信开销下，减少服务器端的性能损耗。本文提出的部分同态加密算法，提高整个保序加密检索模型的安全性。实验结果表明，本文方案在保证安全性基础上能够有效提高密文检索性能。

表 7 mOPE与f-mOPE时间复杂度比较

时间复杂度	mOPE	f-mOPE
计算OPE编码	$O(\lg n)$	$O(\lg n)$
调整编码	最好情况$O(1)$最坏情况$O(n)$	$O(1)$
检查是否平衡/需要调整	最好情况$O(1)$最坏情况$O(n)$	最好情况$O(n)$最坏情况$O(n)$

参 考 文 献

[1] GABEL M and MECHLER J. Secure database outsourcing to the cloud: Side-channels, counter-measures and trusted execution[C]. The 2017 IEEE 30th International Symposium on Computer-Based Medical Systems, Thessaloniki, Greece, 2017: 799–804.

[2] 陆海宁. 可隐藏搜索模式的对称可搜索加密方案[J]. 信息网络安全, 2017(1): 38–42. doi: 10.3969/j.issn.1671-1122.2017.01.006.

LU Haining. Searchable symmetric encryption with hidden search pattern[J]. *Netinfo Security*, 2017(1): 38–42. doi: 10.3969/j.issn.1671-1122.2017.01.006.

[3] DEMERTZIS I and PAPAMANTHOU C. Fast searchable encryption with tunable locality[C]. 2017 ACM International Conference on Management of Data, Chicago, Illinois, USA, 2017: 1053–1067.

[4] PENG Tianyue, LIN Yaping, YAO Xin, *et al*. An efficient ranked multi-keyword search for multiple data owners over encrypted cloud data[J]. *IEEE Access*, 2018, 6: 21924–21933. doi: 10.1109/ACCESS.2018.2828404.

[5] AGRAWAL R, KIERNAN J, SRIKANT R, *et al*. Order preserving encryption for numeric data[C]. 2004 ACM SIGMOD International Conference on Management of Data, Paris, France, 2004: 563–574.

[6] BOLDYREVA A, CHENETTE N, LEE Y, *et al*. Order-preserving symmetric encryption[C]. The 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, 2009: 224–241.

[7] LIU Zheli, CHEN Xiaofeng, YANG Jun, *et al*. New order preserving encryption model for outsourced databases in

cloud environments[J]. *Journal of Network and Computer Applications*, 2016, 59: 198–207. doi: 10.1016/j.jnca.2014.07.001.

[8] TERANISHI I, YUNG M, and MALKIN T. Order-preserving encryption secure beyond one-wayness[C]. The 20th International Conference on the Theory and Application of Cryptology and Information Security, Taiwan, China, 2014: 42–61. doi: 10.1007/978-3-662-45608-8_3.

[9] MAVROFORAKIS C, CHENETTE N, O'NEILL A, *et al*. Modular order-preserving encryption, revisited[C]. 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Australia, 2015: 763–777.

[10] ZHANG Huanguo, HAN Wenbao, LAI Xuejia, *et al*. Survey on cyberspace security[J]. *Science China Information Science*, 2015, 58(11): 1–43. doi: 10.1007/s11432-015-5433-4.

[11] LIU Dongxi and WANG Shenlu. Programmable order-preserving secure index for encrypted database query[C]. The 2012 IEEE 5th International Conference on Cloud Computing, Honolulu, USA, 2012: 502–509.

[12] LIU Dongxi and WANG Shenlu. Nonlinear order preserving index for encrypted database query in service cloud environments[J]. *Concurrency and Computation: Practice and Experience*, 2013, 25(13): 1967–1984. doi: 10.1002/cpe.2992.

[13] 张成果. CryptDB密文数据库系统研究[D]. [硕士论文], 南京邮电大学, 2017.
ZHANG Chengguo. The research of cryptDB encrypted database system[D]. [Master dissertation], Nanjing University of Posts and Telecommunications, 2017.

[14] POPA R A, REDFIELD C M S, ZELDOVICH N, *et al*. processing queries on an encrypted database[J]. *Communications of the ACM*, 2012, 55(9): 103–111. doi: 10.1145/2330667.2330691.

[15] POPA R A, LI F H, and ZELDOVICH N. An ideal-security protocol for order-preserving encoding[C]. 2013 IEEE Symposium on Security and Privacy, Berkeley, USA, 2013: 463–477. doi: 10.1109/SP.2013.38.

[16] VAN DIJK M, GENTRY C, HALEVI S, *et al*. Fully homomorphic encryption over the integers[C]. The 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, 2010: 24–43.

周艺华：男，1969年生，副教授，研究方向为网络与信息安全.
吉　文：男，1993年生，硕士，研究方向为信息安全.
杨宇光：女，1976年生，教授，研究方向为信息安全及信息安全与其他学科的交叉学科.