

支持邻接关系查询的图结构密文搜索方案

徐紫枫, 周福才, 李宇溪, 秦诗悦
(东北大学 软件学院, 辽宁 沈阳 110169)

摘 要: 现有的密文搜索方案不支持复杂数据结构, 因此, 提出一个针对图结构的密文搜索模型, 给出其算法的形式化定义及安全模型. 利用矩阵结构的加密索引提出一个支持邻接关系查询的图结构密文搜索方案, 给出了方案算法的具体描述, 并对安全性与效率进行分析. 方案使用伪随机函数和伪随机置换, 保证了用户的图数据和索引信息不被泄露, 并通过现实模型实验和理想模型实验的方法进行安全性证明. 对比传统密文搜索方案, 该方案支持更加灵活的查询, 并拥有更高的效率, 在大数据环境下拥有广泛的应用前景.

关 键 词: 云存储; 密文搜索; 图结构加密; 邻接矩阵; 加密索引
中图分类号: TP 309 **文献标志码:** A **文章编号:** 1005-3026(2018)08-1092-06

Searchable Encryption for Graph Data with Adjacency Query

XU Zi-feng, ZHOU Fu-cai, LI Yu-xi, QIN Shi-yue
(School of Software, Northeastern University, Shenyang 110169, China. Corresponding author: ZHOU Fu-cai, E-mail: fczhou@mail.neu.edu.cn)

Abstract: This paper proposes a searchable encryption model for graph data to address the problem that most of the current searchable encryption schemes cannot deal with complex data structures. The algorithms and security models of the scheme are described. Taking the encryption index of matrix structure into account, the detailed description of the searchable encryption scheme for graph data that supports adjacency queries is presented. The scheme utilizes pseudo-random functions and pseudo-random permutations to protect the privacy of the graph data and the encrypted index. Real models and ideal models test the security of the scheme. Compared with traditional searchable encryption schemes, the proposed scheme has more flexible searching ability and higher efficiency, which promotes broader application under big data environment.

Key words: cloud storage; searchable encryption; graph encryption; adjacency matrix; encrypted index

在云计算和大数据环境下,数据不再像传统信息系统一样集中存储在本地,而是更多地通过云服务提供商在云端进行存储,以此来节省存储和管理开销.云存储虽然为数据所有者提供了便利,但同时使其失去了对数据安全性和正确性的有效控制.为了保护数据的隐私和安全,加密成为确保云存储机密性的重要手段,但加密后的数据往往会失去原有的特征,包括顺序、大小和所属范围等,导致用户无法使用对存储数据进行搜索、浏览和在线编辑等常用服务.因此如何在保证数据机密性的前提下,确保数据的可用性成为近年来云计算和密码学领域的重要研究内容.

密文搜索是指数据以密文方式存储,并仍然可以对数据进行搜索和查询的技术.目前得到最多关注、并且应用最广泛的密文搜索技术是对称可搜索加密^[1](searchable symmetric encryption, SSE),SSE方案支持对加密文档的关键词搜索.2004年Goh^[2]指出了文献[1]中的不足,并提出了基于布隆过滤器的密文搜索方案.2006年Curtmola等^[3]提出了两个基于倒排索引的密文

搜索方案,方案的搜索操作十分高效. 2010 年 Liesdonk 等^[4]提出两个支持高效更新的密文搜索方案. 2012 年, Kamara 等^[5]对 CGK + I 方案进行扩展与改进,使其可以支持高效的更新操作(文档添加、删除和修改). 近年来,针对密文搜索存在的问题,很多拥有不同特性的方案被提出,例如支持模糊查询的方案^[6-9]、支持动态更新的方案^[10-13]、支持完整性验证的方案^[14-17]和支持布尔运算的方案等^[18-20].

目前的密文搜索方案大多集中在对文件集合进行加密,并使用关键词进行搜索;然而在大数据环境下,数据的存储结构更加复杂,并且需要支持更加灵活的查询操作. 图结构是最常用的数据结构之一,被广泛应用于很多领域. 将数据以图结构的形式进行储存,便可以利用图论的多种算法对数据进行不同的操作与查询. 因此针对图结构的密文搜索方案在云环境中拥有广泛的应用前景.

本文针对目前现有的密文搜索方案大多仅支持简单数据结构的问题,提出了一个支持邻接关系查询的图结构密文搜索方案. 方案使用矩阵结构构建加密索引,并使用伪随机函数和伪随机置换保证了方案的机密性与隐私性.

1 预备知识

1.1 密文搜索方案通用模型

密文搜索方案包括两方实体:客户端 C 和服务器 S. 其中客户端拥有私有数据 M ,并希望在不泄露任何信息的前提下将 M 存储在不可信的服务器上,同时能够随时对数据进行搜索操作.

密文搜索方案包含两个阶段:初始化阶段和查询阶段. 在初始化阶段中,客户端将私有数据 M 进行加密,生成密文数据 M' ,并根据需要支持的查询类型生成加密索引 δ ,然后将 M' 和 δ 发送到服务器进行保存;在查询阶段,客户端根据查询语句 q 生成查询令牌 τ ,并发送给服务器. 服务器根据查询令牌在加密索引中进行查找和计算,并返回给客户端查询结果 $m' \in M'$. 最后客户端对密文查询结果进行解密操作,得到明文查询结果 $m \in M$.

密文搜索方案有两个安全目标:索引安全和查询安全. 索引安全是指,敌手服务器根据加密索引 δ 和密文数据 M' ,无法得到任何关于私有数据 M 的信息;查询安全是指,敌手服务器根据适应性选择的查询语句 $q = (q_1, \dots, q_n)$ 和对应的查询令牌 $\tau = (\tau_1, \dots, \tau_n)$,无法得到任何关于查询语句

q 和查询结果 m 的信息.

但是在实际应用中,严格的索引安全和查询安全往往很难达到,并且在部分应用场景中并不必要. 因此在考虑密文搜索方案的安全性时,需要构造泄露函数 L_1 和 L_2 ,分别指明方案中在加密索引和查询令牌中允许泄露的信息.

1.2 伪随机函数与伪随机置换

设 D 和 R 是两个域, K 是密钥空间,其中 $D = \{0, 1\}^m, R = \{0, 1\}^n, K = \{0, 1\}^k$. 伪随机函数的定义为 $f: \{0, 1\}^m \times \{0, 1\}^k \rightarrow \{0, 1\}^n$. 一个安全的伪随机函数符合以下两个性质:

1) 给予一个输入 $x \in D$ 和一个密钥 $k \in K$,存在多项式时间算法来计算 $f: x \times k \rightarrow y$,其中 $y \in R$.

2) 任意的多项式时间敌手无法区分伪随机函数的结果与随机预言机(随机函数,对于任何输入都返回均匀分布的随机结果).

相对于伪随机函数,伪随机置换将输出映射到与输入相同的域中,它的定义为 $f: \{0, 1\}^m \times \{0, 1\}^k \rightarrow \{0, 1\}^m$,其安全定义与伪随机函数相同.

2 模型与定义

2.1 符号标志与图的表示

本节给出支持邻接关系查询的图结构密文搜索方案的形式化描述与安全定义. 为了便于描述,将采用表 1 中的符号标志.

表 1 标志及描述	
Table 1	Symbols and descriptions
标志	描述
S	服务器
C	客户端
$G = (V, E, A)$	图数据
$V = \{v_1, v_2, \dots, v_n\}$	图中点的集合
n	图中点的数量
$E = \{(v_i, v_j) \mid v_i, v_j \in V\}$	图中边的集合
$A = (a_{ij})$	图的邻接矩阵
k	安全参数
F	伪随机函数
P	伪随机置换
π	IND-CPA 的对称加密算法
K	密钥
δ	加密索引
τ	查询令牌
L_1, L_2	泄露函数

方案中的图数据为有向图,由点集合 V 、边集合 E 和邻接矩阵 A 组成: $G = (V, E, A)$. $A =$

$(a_{ij})_{n \times n}$, 其中 $a_{ij} = \{0, 1\}$, 表示点 v_i 和点 v_j 在图中的邻接关系, 若 v_i 和 v_j 在图中存在边, 则 $a_{ij} = 1$, 否则 $a_{ij} = 0$.

2.2 模型形式化描述

支持邻接关系查询的图结构密文搜索方案包含两方实体: 客户端 C 和服务器 S. 客户端希望将隐私的图数据 G 保存在不可信服务器上, 并且能够在不泄露信息的情况下对 G 进行邻接关系查询.

方案包含两个阶段: 初始化阶段和查询阶段. 在初始化阶段中, 客户端根据自己的图数据生成加密索引, 并将加密索引发送给服务器保存, 初始化阶段仅执行一次. 在查询阶段中, 客户端根据查询语句生成查询令牌, 并将查询令牌发送到服务器. 其中查询语句包含图中两个点的标志符 i, j , 即查询点 v_i 和点 v_j 在图中是否邻接. 服务器收到查询令牌后, 在加密索引中进行查找和计算, 并返回给客户端查询结果, 查询阶段可执行多次.

方案由 5 个多项式时间算法组成: Setup, EDS, Token, Query, Dec. 各算法的具体描述如下:

1) $K \leftarrow \text{Setup}(1^k)$: 初始化算法, 输入为安全参数 k , 输出为密钥 K .

2) $\delta \leftarrow \text{EDS}(K, G)$: 加密索引构建算法, 输入为密钥 K 和图数据 G , 输出为加密索引 δ .

3) $\tau \leftarrow \text{Token}(K, q)$: 查询令牌生成算法, 输入为密钥 K 和查询语句 q , 输出为查询令牌 τ .

4) $c \leftarrow \text{Query}(\delta, \tau)$: 查询算法, 输入为加密索引 δ 和查询令牌 τ , 输出为搜索结果 c (密文).

5) $r \leftarrow \text{Dec}(K, c)$: 解密算法, 输入为密钥 K 和密文结果 c , 输出为明文结果 r .

2.3 安全定义

设 L_1 和 L_2 为两个泄露函数, 其中 L_1 为图 G 中点的数量 n , L_2 为查询模式 (两次查询是否使用相同的查询语句).

对于敌手服务器 A, 通过构建现实模型实验和理想模型实验给出安全定义.

现实模型实验 $\text{Real}_{A,C}(k)$: 敌手 A 和客户端 C 通过方案中的真实算法进行交互. 在初始化阶段, A 首先选择两个图 G_0 和 G_1 并发送给客户端, 客户端随机选择其中一个图作为输入, 执行方案的 Setup 算法和 EDS 算法, 并将生成的加密索引发送给 A. 在查询阶段, A 选择查询语句 q 并发送给客户端, 客户端使用 Token 算法生成查询令牌, 并发送给 A. 在经过多项式次查询后, 若 A 能够以不可忽略的概率判断出客户端选择了 G_0 还是 G_1 , 则敌手获胜.

理想模型实验 $\text{Ideal}_{A,\text{Sim}}(k)$: 敌手 A 和模拟器 Sim 进行交互. 在初始化阶段, 敌手选择图 G 并发送给模拟器; 模拟器根据 L_1 生成矩阵 δ, δ 的大小为 $n \times n$, 矩阵中所有的值为随机字符串. 在查询阶段, A 选择查询语句 q 并发送给模拟器; 模拟器根据 L_2 判断以前是否使用查询语句查询过: 若查询过, 则返回 A 之前生成的查询令牌, 若没查询过, 则随机挑选邻接矩阵中未访问过的位置 x, y , 并生成查询令牌返回 A.

若敌手 A 在现实模型实验和理想模型实验中的视都是可忽略的, 则方案在泄露函数 (L_1, L_2) 下针对适应性选择查询攻击是安全的 (简称为 IND-CQA2 安全).

定理 1 支持邻接关系查询的图结构密文搜索方案在泄露函数 (L_1, L_2) 下针对适应性选择查询攻击是安全的, 即 IND-CQA2 安全.

3 方案构建

3.1 算法描述

设 $F: \{0, 1\}^k \times \{0, 1\}^* \times \{0, 1\}^* = \{0, 1\}^*$ 是一个安全的伪随机函数, $P: \{0, 1\}^k \times [n] = [n]$ 是一个安全的伪随机置换, 其中 n 是客户端的图数据中点的数量, $\pi = (\text{Gen}, \text{Enc}, \text{Dec})$ 是一个针对不可区分的选择明文攻击具有安全性 (即 IND-CPA 安全) 的对称加密算法. 在没有密钥的情况下, π 生成的密文和 F, P 的结果与随机字符串在多项式时间内是计算不可区分的.

支持邻接关系查询的图结构密文搜索方案算法的详细构建方法如下.

1) $K_1, K_2, K_3 \leftarrow \text{Setup}(1^k)$: 初始化算法, 由客户端执行. 首先根据安全参数 k , 随机生成两个 k 比特长度的密钥 K_1 和 K_2 ; 然后选择 IND-CPA 安全的对称加密算法 π , 并生成 $K_3 \leftarrow \pi. \text{Gen}(1^k)$. 算法的伪代码如下:

Algorithm: $K_1, K_2, K_3 \leftarrow \text{Setup}(1^k)$

Input: security parameter k

Output: secret keys K_1, K_2 and K_3

1 Random Sample $K_1 \xleftarrow{\$} \{0, 1\}^k$

2 Random Sample $K_2 \xleftarrow{\$} \{0, 1\}^k$

3 Choose a IND-CPA symmetric encryption scheme π

4 Run $K_3 \leftarrow \pi. \text{Gen}(1^k)$

5 Return K_1, K_2 and K_3

2) $\delta \leftarrow \text{EDS}(K_1, K_2, K_3, G)$: 加密索引构建算法, 由客户端执行. 设图 G 中点的数量为 n , 首先构建新的矩阵 δ , δ 和图 G 的邻接矩阵 A 的大小相同, 即 $n \times n$; 对于所有的 $1 \leq i, j \leq n$, 在邻接矩阵中查找 a_{ij} , 并使用对称加密算法 π 对其进行加密, 得到 $t = \pi. \text{Enc}_{K_3}(a_{ij})$; 之后使用伪随机置换, 根据 i 和 j 计算新的位置坐标, $i' = P_{K_2}(i)$, $j' = P_{K_2}(j)$; 最后, 使用异或操作计算 $t \oplus F_{K_1}(i, j)$, 并将结果保存到 $\delta[i', j']$ 中. 矩阵 δ 即为图 G 的加密索引. 算法的伪代码如下:

Algorithm: $\delta \leftarrow \text{EDS}(K_1, K_2, K_3, G)$

Input: secret keys K_1, K_2, K_3 and graph $G = \{V, E, A\}$

Output: encrypted index δ

```

1 Initialize Matrix  $\delta$  of size  $n \times n$ 
2 for each  $1 \leq i, j \leq n$ 
3   Compute  $t = \pi. \text{Enc}_{K_3}(a_{i,j})$ 
4   Compute  $i' = P_{K_2}(i)$ 
5   Compute  $j' = P_{K_2}(j)$ 
6   Assign  $\delta[i', j'] = t \oplus F_{K_1}(i, j)$ 
7 end for each
8 Return  $\delta$ 

```

3) $\tau \leftarrow \text{Token}(K_1, K_2, q)$: 查询令牌生成算法, 由客户端执行. 一个邻接关系查询 q 中包含图中两个点的标志符, $q = (x, y)$. 在生成查询令牌时, 首先使用伪随机置换, 计算 $x' = P_{K_2}(x)$ 和 $y' = P_{K_2}(y)$; 然后使用伪随机函数, 计算 $s = F_{K_1}(x, y)$. 查询令牌 τ 包含以上三个部分, 即 $\tau = (s, x', y')$. 算法的伪代码如下:

Algorithm: $\tau \leftarrow \text{Token}(K_1, K_2, q)$

Input: secret keys K_1, K_2 , and query $q = (x, y)$

Output: query token τ

```

1 Compute  $x' = P_{K_2}(x)$ 
2 Compute  $y' = P_{K_2}(y)$ 
3 Compute  $s = F_{K_1}(x, y)$ 
4 Return  $\tau = (s, x', y')$ 

```

4) $c \leftarrow \text{Query}(\delta, \tau)$: 查询算法, 由服务器执行. 在收到查询令牌 τ 后, 服务器首先在加密索引中查找 (x', y') 位置的值, 即 $\delta[x', y']$; 之后计算 $c = \delta[x', y'] \oplus s$, c 是查询结果的密文, 即点 v_x 和 v_y 在图 G 中的邻接关系的密文. 算法的伪代码如下:

Algorithm: $c \leftarrow \text{Query}(\delta, \tau)$

Input: encrypted index δ and query token τ

Output: query result c

```

1 Parse  $\tau$  as  $(s, x', y')$ 
2 Lookup  $\delta[x', y']$ 
3 Compute  $c = \delta[x', y'] \oplus s$ 
4 Return  $c$ 

```

5) $r \leftarrow \text{Dec}(K_3, c)$: 解密算法, 由客户端执行. 在收到查询结果的密文 c 后, 用户使用 π 和密钥 K_3 进行解密, 得到 $r = \pi. \text{Dec}_{K_3}(c)$. r 是查询 q 的最终结果, 即点 v_x 和 v_y 在图 G 中的邻接关系. 算法的伪代码如下:

Algorithm: $r \leftarrow \text{Dec}(K_3, c)$

Input: secret key K_3 and ciphertext c

Output: plaintext r

```

1 Run  $r = \pi. \text{Dec}_{K_3}(c)$ 
2 Return  $r$ 

```

3.2 方案交互过程

方案交互过程分为初始化阶段和查询阶段.

在初始化阶段, 客户端首先执行初始化算法 Setup, 生成密钥; 然后执行加密索引构建算法 EDS, 使用密钥将隐私图数据进行加密, 并生成加密索引; 最后客户端将加密索引发送给服务器. 初始化阶段仅执行一次.

在查询阶段, 客户端根据查询语句, 首先执行查询令牌生成算法 Token, 使用密钥和查询语句生成查询令牌, 并发送给服务器; 服务器在收到查询令牌后, 执行查询算法 Query, 在加密索引中查找与计算, 并返回客户端查询结果; 客户端收到查询结果后, 执行解密算法 Dec, 利用密钥将密文的查询结果解密, 得到最终的明文查询结果. 查询阶段可多次执行. 方案的交互图如图 1 所示.

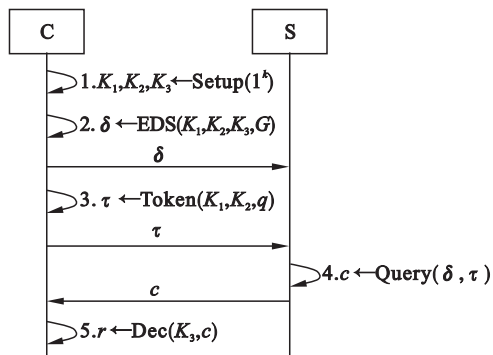


图 1 方案交互图

Fig. 1 Interaction diagram of the scheme

4 安全性证明和效率分析

4.1 定理 1 的安全性证明

根据方案的安全性定义,通过构建现实模型实验和理想模型实验证明方案的安全性,即证明,对于敌手服务器 A,加密索引 δ 不泄露任何关于图数据 G 的信息,并且适应性选择的查询语句 q 不泄露任何关于查询的信息,除了预定义的泄露函数 L_1 和 L_2 中的信息.

首先构建现实模型实验,在现实模型实验中,敌手服务器 A 与真实的用户使用方案中的算法进行交互. 具体实验如下:

$\text{EXP}_A^{\text{Real}}(1^k):$
 $(G_0, G_1) \leftarrow A$
 $b \xleftarrow{\$} \{0, 1\}$
 $K_1, K_2, K_3 \leftarrow \text{Setup}(1^k)$
 $\delta \leftarrow \text{EDS}(K_1, K_2, K_3, G_b)$
 $q \leftarrow A$
 $\tau \leftarrow \text{Token}(K_1, K_2, q)$
 $\hat{b} \leftarrow A(\delta, \tau)$
 if $\hat{b} = b$, output 1
 otherwise, output 0

在以上实验中, A 是一个概率多项式时间敌手服务器. 在初始化阶段, A 选取两个相同大小(点的数量相同)的图, G_0 和 G_1 , 并发送给客户端 C. 在收到两个图后, 客户端随机选取一个比特 $b = \{0, 1\}$, 并选择图 G_b 作为自己的图数据. 之后客户运行方案中的 Setup 算法和 EDS 算法, 得到加密索引 δ , 并发送给敌手服务器. 在查询阶段, A 选择查询语句 q 并发送给客户端. 客户端使用方案中的 Token 算法生成查询令牌并返回给 A. 在进行多项式次查询后, A 根据学到的知识(δ 和 τ)猜测一个比特 $\hat{b} = \{0, 1\}$. 如果 $b = \hat{b}$, 则实验输出 1, 否则输出 0.

敌手 A 想要赢得以上实验,即以不可忽略的概率使实验输出 1 或 0,则需要判断出收到的 δ 是根据 G_0 还是 G_1 生成的. 因为 G_0 与 G_1 拥有相同的大小,因此生成的 δ 的大小也相同. 根据方案中的算法 EDS, $\delta[x', y']$ 中存储的数据为 $\pi. \text{Enc}_{K_3}(a_{xy}) \oplus F_{K_1}(x, y)$, 其中 a_{xy} 是图 G_b 的邻接矩阵 A 的元素, $x' = P_{K_2}(x)$, $y' = P_{K_2}(y)$. 因此若 π 是一个 IND-CPA 的对称加密算法,且 F 和 P 是安全的伪随机函数和伪随机置换,则 δ 中的所有数据计算不可区分.

在查询时,客户端根据查询语句生成查询令

牌. 根据方案中的算法 Token,对于查询语句 $q = (x, y)$, 查询令牌的生成方法为 $\tau = (F_{K_1}(x, y), P_{K_2}(x), P_{K_2}(y))$. 因此若 F 和 P 是安全的伪随机函数和伪随机置换,则查询令牌与随机字符串计算不可区分.

综上所述,敌手 A 在该实验中获得优势的概率是可忽略的,即 $\text{Adv}_A = |\Pr[\text{EXP}_A^{\text{Real}}] - \Pr[\text{EXP}_A^{\text{Ideal}}]| = \varepsilon$, 即 A 在现实模型中的视是可忽略的,即 $\text{View}_A^{\text{Real}}[C(G_A), A] = \varepsilon$.

然后构建理想模型实验,在理想模型实验中,敌手服务器 A 与模拟器 Sim 进行交互. 在初始化阶段, Sim 根据泄露函数 L_1 , 发送一个矩阵 δ 给 A, δ 的大小与 A 选择的图 G 的邻接矩阵的大小相同,且 δ 中保存的数据为随机字符串. 在查询阶段, A 选择查询语句 q , 并发送给 Sim. Sim 根据泄露函数 L_2 判断该查询语句之前是否出现过. 若出现过,则返回 A 之前生成的查询令牌;若没出现过,则随机挑选未访问过的位置 x, y , 生成查询令牌,并返回给 A.

若 π 是 IND-CPA 安全的对称加密算法,则算法生成的密文与随机字符串不可区分,并且若 F 和 P 是安全的伪随机函数和伪随机置换,则 δ 中的所有数据与随机字符串计算不可区分. 综上所述,敌手 A 在理想模型实验中的视是可忽略的,即 $\text{View}_A^{\text{Ideal}}[\text{Sim}(L_1, L_2), A] = \varepsilon$.

根据以上两个实验,可以得到 $\text{View}_A^{\text{Real}}[C(G_A), A] \approx \text{View}_A^{\text{Ideal}}[\text{Sim}(L_1, L_2), A] = \varepsilon$, 即敌手在现实模型下和理想模型下的视都是可忽略的. 因此对于敌手服务器 A,加密索引 δ 不泄露任何关于图数据 G 的信息,并且适应性选择的查询令牌 τ 不泄露任何关于查询语句的信息,除了预定义的泄露函数 L_1 和 L_2 中的信息.

综上所述,如果 π 是一个 IND-CPA 的对称加密算法,且 F 和 P 是安全的伪随机函数和伪随机置换,则支持邻接关系查询的图结构密文搜索方案在泄露函数(L_1, L_2)下针对适应性选择查询攻击是安全的.

4.2 效率分析

在初始化阶段,客户端根据图数据 $G = (V, E, A)$, 使用算法 EDS 生成加密索引 δ . 设图中点的数量为 n , 首先需要对邻接矩阵 A 中的所有数据进行加密. A 的大小为 $n \times n$, 因此计算量为 n^2 个加密操作;之后需要对 A 中所有的位置执行伪随机函数 F , 并与之前的密文做异或操作,因此计算量是 n^2 个 F 和 n^2 个异或操作;最后需要对 A

中所有的位置执行两次伪随机置换 P , 即 $2n^2$ 个 P . 综上所述, 初始化阶段的计算量为 n^2 个加密操作 + n^2 个 $F + 2n^2$ 个 $P + n^2$ 个异或操作. 通常来讲, 加密操作的计算复杂度最高, 因此方案在生成加密索引时的计算复杂度为 $O(n^2)$ 个加密操作.

在查询阶段, 客户端首先根据查询语句 q , 使用算法 Token 生成查询令牌 τ . 在算法中需要 2 个伪随机置换 P 和 1 个伪随机函数 F ; 之后客户端将 τ 发送给服务器, 并收到查询结果 c ; 最后客户端使用 Dec 算法对查询结果进行解密, 其计算量为 1 个解密操作. 综上所述, 查询阶段客户端的计算量为 2 个 $P + 1$ 个 $F + 1$ 个解密操作. 通常来讲, 解密操作的计算复杂度最高, 因此方案在查询阶段时客户端的计算复杂度为 $O(1)$ 个解密操作.

在收到查询令牌后, 服务器执行 Query 算法, 在加密索引 δ 中进行查找与计算. 算法首先查找 δ 中一个位置的值, 之后进行 1 次异或操作. 所以方案在查询阶段时服务器的计算复杂度为 $O(1)$ 个异或操作.

5 结 语

针对目前现有的密文搜索方案仅支持简单数据结构的问题, 本文提出了支持邻接关系查询的图结构密文搜索方案模型, 并给出了方案的形式化描述和安全性定义. 使用现实实验和理想实验的方法, 对方案的安全性进行了证明, 并对效率进行分析. 对比传统密文搜索方案, 该方案支持更加灵活的查询, 并拥有更高的效率, 同时保护了用户图数据的机密性和查询的隐私性, 在大数据环境下拥有广泛的应用前景.

参考文献:

- [1] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data [C]// IEEE Symposium on Security and Privacy. Oakland, 2000: 44 – 55.
- [2] Goh E J. Secure indexes [J/OL]. [2017 – 02 – 04]. <https://gnunet.org/sites/default/files/secureindex.pdf>.
- [3] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: improved definitions and efficient constructions [C]// ACM Conference on Computer and Communications Security. Alexandria, 2006: 79 – 88.
- [4] van Liesdonk P, Sedghi S, Doumen J, et al. Computationally efficient searchable symmetric encryption [C]// Workshop on Secure Data Management. Singapore, 2010: 87 – 100.
- [5] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption [C]// Proceedings of the 2012 ACM Conference on Computer and Communications Security. Raleigh, 2012: 965 – 976.
- [6] Li J, Wang Q, Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing [C]// 2010 Proceedings IEEE INFOCOM. San Diego, 2010: 1 – 5.
- [7] Chuah M, Hu W. Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data [C]// 2011 31st International Conference on Distributed Computing Systems Workshops (ICDCSW). Minneapolis, 2011: 273 – 281.
- [8] Liu C, Zhu L, Li L, et al. Fuzzy keyword search on encrypted cloud storage data with small index [C]// 2011 IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS). Beijing, 2011: 269 – 273.
- [9] Wang C, Ren K, Yu S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data [C]// 2012 Proceedings IEEE INFOCOM. Orlando, 2012: 451 – 459.
- [10] Stefanov E, Papamanthou C, Shi E. Practical dynamic searchable encryption with small leakage [C]// NDSS Symposium 2014. San Diego, 2014: 72 – 75.
- [11] Yang Y, Li H, Liu W, et al. Secure dynamic searchable symmetric encryption with constant document update cost [C]// 2014 IEEE Global Communications Conference (GLOBECOM). Austin, 2014: 775 – 780.
- [12] Cash D, Jaeger J, Jarecki S, et al. Dynamic searchable encryption in very-large databases: data structures and implementation [C]// NDSS Symposium 2014. San Diego, 2014: 23 – 26.
- [13] Kamara S, Papamanthou C. Parallel and dynamic searchable symmetric encryption [C]// International Conference on Financial Cryptography and Data Security. Okinawa, 2013: 258 – 274.
- [14] Chai Q, Gong G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers [C]// 2012 IEEE International Conference on Communications (ICC). Ottawa, 2012: 917 – 922.
- [15] Zheng Q, Xu S, Ateniese G. VABKS: verifiable attribute-based keyword search over outsourced encrypted data [C]// 2014 Proceedings IEEE INFOCOM. Toronto, 2014: 522 – 530.
- [16] Zhou F, Li Y, Liu A X, et al. Integrity preserving multi-keyword searchable encryption for cloud computing [C]// International Conference on Provable Security 2016. Nanjing, 2016: 153 – 172.
- [17] Alderman J, Janson C, Martin K M, et al. Extended functionality in verifiable searchable encryption [C]// International Conference on Cryptography and Information Security in the Balkans. Koper, 2015: 187 – 205.
- [18] Cash D, Jarecki S, Jutla C, et al. Highly-scalable searchable symmetric encryption with support for boolean queries [M]. Berlin: Springer, 2013: 353 – 373.
- [19] Moataz T, Shikfa A. Boolean symmetric searchable encryption [C]// Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security. Hangzhou, 2013: 265 – 276.
- [20] Ryu E K, Takagi T. Efficient conjunctive keyword-searchable encryption [C]// 21st International Conference on Advanced Information Networking and Applications Workshops. Ontario, 2007: 409 – 414.