安全检索技术

山东大学软件学院

第三章　安全检索技术

3.1 基本概念

3.2 早期安全检索技术

3.3 对称密文检索

3.4 非对称密文检索

3.5 密文区间检索

3.1.1 背景介绍

云存储是在云计算概念上衍生出来的，其继承了云计算的按需使用、高可扩展性、快速部署等特点，解决了当前政府和企业需要不断增加软硬件设备和数据库管理人员来自主地存储、管理和维护海量数据的问题。

由于云存储使得数据的所有权和管理权相分离，用户数据将面临多方面的安全威胁：

- 美国政府雇员窃取社保信息
- icloud好莱坞明星隐私泄露事件
- 著名的美国国家安全局的“棱镜”项目

3.1.1 背景介绍

为保证**云数据的安全性，一种通用的方法是：**

- **用户首先使用安全的加密机制（如DES、AES、RSA等）对数据进行加密，然后再将密文数据上传至云服务器。**
- **当用户需要对数据进行检索时，只能把全部密文下载到本地，将其解密后再查询。**

这个过程要求客户端具有较大的存储空间以及较强的计算能力，且没有充分发挥云存储的优势。

密文检索（Searchable Encryption，SE）技术支持云存储系统**在密文场景下对用户数据进行检索，然后将满足检索条件的密文数据返回给用户。**

在检索过程中，云服务器无法获得用户的敏感数据和查询条件，即密文检索可以同时保护数据机密性以及查询机密性。

3.1.2 密文检索概述

1. 系统模型：

密文检索主要涉及**数据所有者、数据检索者以及服务器3种角色**。其中，数据所有者是敏感数据的拥有者，数据检索者是查询请求的发起者，这二者通常仅具备有限的存储空间和计算能力；服务器为所有者和检索者提供数据存储和数据查询服务，其由云存储服务提供商进行管理和维护，并具有强大的存储能力和计算能力。

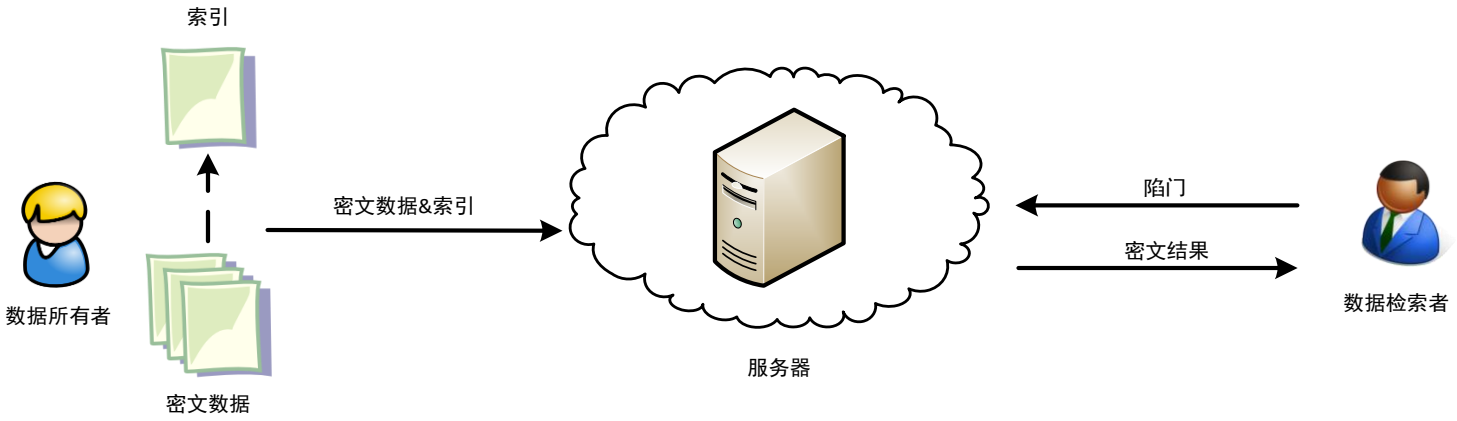


图3-1 密文检索结构图

3.1.2 密文检索概述

2. 系统流程：

① **数据所有者**首先为数据**构造支持检索功能的索引**，同时使用传统的加密技术**加密全部数据**，然后**将密文数据和索引共同存储到服务器**。

② 需要检索时，**数据检索者**为检索条件**生成相应的陷门**，并发送给服务器。

③ 服务器使用索引和陷门进行协议预设的运算，并将满足检索条件的密文数据返回给数据检索者。

④ 数据检索者使用密钥将检索结果解密，得到明文数据。有时服务器返回的密文数据中可能包含不满足检索条件的冗余数据，此时数据检索者还需要对解密后的明文数据进行二次检索，即在本地剔除冗余数据。

3. 安全模型：

通常认为**数据所有者和数据检索者是完全可信**的，而服务器属于攻击者，其对用户的敏感数据和检索条件比较好奇。目前**大部分密文检索方案均假设服务器是诚实且可信的（Honest-but-Curious，HBC），即服务器会忠实地执行数据检索者提交的检索请求，并返回相应的检索结果，同时其可能会利用自己所掌握的一切背景知识来进行分析，期望获得真实的敏感数据和检索条件**。

3.1.3 密文检索分类

根据应用场景的不同，密文检索技术可以分为**对称密文检索SSE**和**非对称密文检索ASE**两大类：

- **对称密文检索**：在对称密钥环境下，只有数据所有者拥有密钥，也只有数据所有者可以提交敏感数据、生成陷门，即**数据所有者和数据检索者为同一人**。
- **非对称密文检索**：在非对称密钥环境下，任何可以获得数据检索者公钥的用户都可以提交敏感数据，但只有拥有私钥的数据检索者可以生成陷门。

表3-1 对称密文检索和非对称密文检索的比较

特性	对称密文检索	非对称密文检索
密文和索引的构建	由私钥生成	由公开参数生成
密钥管理	单用户场景	多用户场景
性能	高效	低效
解决的问题	不可信服务器存储	不可信服务器路由

3.1.3 密文检索分类

根据检索的数据类型的不同，密文检索技术可以分为**关键词检索**和**区间检索**两大类：

- **关键词检索：**主要用于检索**字符型/文本型数据**，如查询包含关键词“云存储”的文档。最初，关键词检索的研究以**单关键词检索**为主，后来逐渐扩展到**多关键词检索、模糊检索以及Top-k检索**。
- **区间检索：**主要用于对**数值型数据**进行范围查询，如查询学生信息表中年龄属性小于18的学生。根据属性的数目，区间检索又可进一步分为**单维区间检索**和**多维区间检索**。

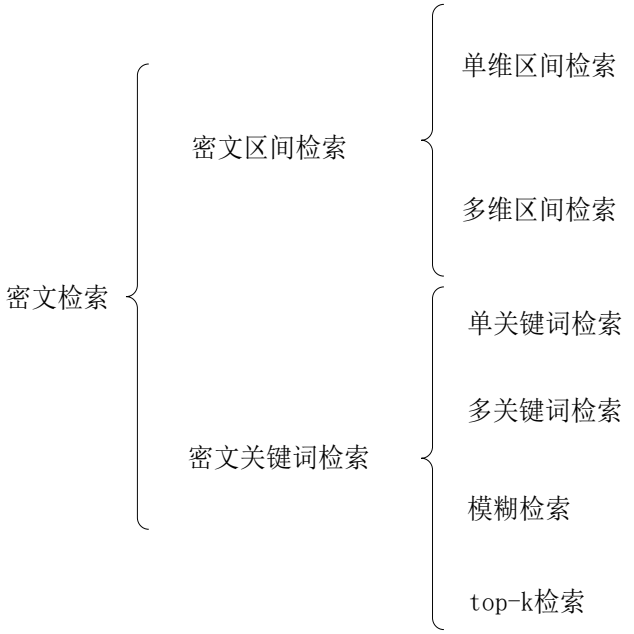


图3-2 密文检索的功能分类

3.2 早期安全检索技术

3.2.1 PIR技术

3.2.2 扩展：PIRK技术以及SPIR技术

3.2.3 ORAM技术

3.2.1 PIR技术

PIR技术的研究**主要针对公开数据库**，其**目标是允许用户在不向服务器暴露查询意图的前提下，对服务器中的数据进行查询并取得指定内容**。

根据服务器的数目以及用户与服务器之间的交互轮数的不同，可将**PIR技术分为4大类：单服务器的、多服务器的、单轮交互的以及多轮交互的**。目前主要研究的是单轮交互的PIR问题。

3.2.1 PIR技术

定义3-1（单轮交互的PIR问题）Private Information Retrieval

设存在$k \geq 1$个服务器，其存储的内容完全相同，均为n个比特的信息$X = \{x_1, x_2, \dots, x_n\}$，且服务器之间不会进行相互通信。A希望对服务器中的数据进行查询，并得到x_i，其具体查询过程如下：

1. A生成一个随机数r，并根据r和i生成k个查询$\{q_1, q_2, \dots, q_k\}$，然后将其分别发送给k个服务器；
2. 各服务器分别返回相应的查询结果：$\{Ans(q_1), \dots, Ans(q_k)\}$；
3. A根据r和$\{Ans(q_1), \dots, Ans(q_k)\}$计算得到正确的$x_i$。

如果在上述查询过程中，任意服务器均不了解关于i的任何信息，则称这一交互是PIR的。换句话说，如果A进行了两次查询，分别访问了x_i和$x_{i'}$，则服务器j对这两次查询所见的视图在概率分布上没有区别，即

$$\forall i, i', j: \Pr\left[View_j(X, i) = view\right] = \Pr\left[View_j(X, i') = view\right]$$

3.2.1 PIR技术

用户想要查询的 index: 2 （秘密）
- 索引集 q1: {0, 1}
- 索引集 q2: {0, 1, 2}

Server1 返回 r1: r1 = DB[0] XOR DB[1] = 0

Server2 返回 r2: r2 = DB[0] XOR DB[1] XOR DB[2] = 1

DB[1]：表示DB上索引为1的二进制位上的二进制数据

3.2.2 扩展：PIRK技术以及SPIR技术

1. PIRK技术

PIR方案假设数据是二进制的，且客户端已经了解待获取的数据在数据集中的位置。但是在实际检索场景中，客户端一般都是输入一个感兴趣的关键词，然后服务器根据该关键词找到对应的数据内容。为此，提出了PIRK（Private Information Retrieval by Keywords）技术。

定义3-3（PIRK问题）

设存在k个服务器，其存储的内容完全相同，均为n个长度为l的字符串$S=\{s_1,s_2,\dots,s_n\}$，且服务器之间不会进行相互通信。A感兴趣的关键词是一个长度为l的字符串w。

如果存在一个协议使A能够得到所有满足$s_j=w$的j，且任意服务器均不了解关于w的任何信息，则称该协议是$PIRK(l,n,k)$的。

需要注意的是，此定义只包含了找到$s_j=w$的过程，至于找到s_j之后如何获取s_j对应的数据内容，则可以通过运行一般的PIR协议来完成。

3.2.3 ORAM技术(Oblivious Random Access Machine)

ORAM技术是面向秘密数据库的，其**目标是在读写过程中向服务器隐藏用户的访问模式**。这里，**访问模式是指客户端向服务器发起访问所泄露的信息，包括操作是读还是写、操作的数据地址、操作的数据内容等**。

ORAM与PIR的不同之处:

- **PIR只考虑保护客户端的查询意图，整个数据库的内容对服务器是可见的**;
- **ORAM则认为整个服务器的存储介质都是不安全的，因此要求数据是加密的，同时向服务器隐藏读写两种操作**。

3.2.3 ORAM技术

定义3-4（ORAM系统）

- 用户的输入序列Y：定义为一组输入$(o_1, o_2, \dots, o_n)$。
- 用户的输入o：代表用户的操作类型、操作数据内容、操作地址$o = (op, data, i)$。其中当op是读时$op = read, data = \emptyset$；当$op$是写时$op = write$，$data$是用户写入的明文内容。
- 访问模式$A(Y)$：对于用户发起的一个输入序列$Y = (o_1, o_2, \dots, o_n)$，假设经过翻译后，在服务器实际实施的序列为$Y = (O_1, O_2, \dots, O_m)$。$A(Y)$记录了$Y$中各输入的访问地址$i$，以及$Y$是读还是写。当操作是写时，$A(Y)$还记录了用户希望服务器写入的内容。具体地，$A(Y) = ((op_1, Edata_1, i_1), \dots, (op_m, Edata_m, i_m))$，其中$Edata$是从服务器的角度写入的密文内容。当操作是读时，$Edata = \emptyset$。

如果对于系统中任意两个输入序列Y和Y'，从服务器的角度来看，访问模式$A(Y)$和$A(Y')$是不可区分的，则认为这个系统是一个ORAM系统。

3.3 对称密文检索

3.3.1 概述

3.3.2 基于全文扫描的方案

3.3.3 基于文档-关键词索引的方案

3.3.4 基于关键词-文档索引的方案

3.3.5 扩展1：多关键词SSE检索

3.3.6 扩展2：模糊检索、Top-k检索、多用户SSE

3.3.7 扩展3：前向安全性扩展

3.3.8 小结

3.3.1 概述

在**对称密文检索方案**中，**数据所有者和数据检索者为同一方**。该场景适用于大部分第三方存储，也是近几年本领域的研究热点。一个典型的对称密文检索方案包括如下算法：

- **Setup算法：**该算法由数据所有者执行，**生成**用于加密数据和索引的**密钥**；
- **BuildIndex算法：**该算法由数据所有者执行，根据数据内容**建立索引**，并将加密后的索引和数据本身上传到服务器；
- **GenTrapdoor算法：**该算法由数据所有者执行，**根据检索条件生成相应的陷门**（又称“搜索凭证”），然后将其发送给服务器；
- **Search算法：**该算法由服务器执行，将接收到的陷门和本地存储的密文索引作为输入，并进行协议所预设的**密文查询计算**，最后**输出满足条件的密文结果**。

目前，SSE可根据检索机制的不同大致分为**三大类：基于全文扫描的方法、基于文档-关键词索引的方法以及基于关键词-文档索引的方法**。

3.3.2 基于全文扫描的方案

最早的对称密文检索方案由**加州伯克利大学Dawn Xiaodong Song等**人提出，其是**一种基于全文扫描的方案**。如图3-3所示，该方案的**核心思想是：(1) 对文档进行分组加密**，然后**将分组加密结果与一个伪随机流进行异或得到最终用于检索的密文。(2) 检索时**，用户将检索关键词对应的陷门发送给服务器，服务器对所有密文**依次使用陷门计算密文是否满足预设的条件**，若满足则返回该文档。

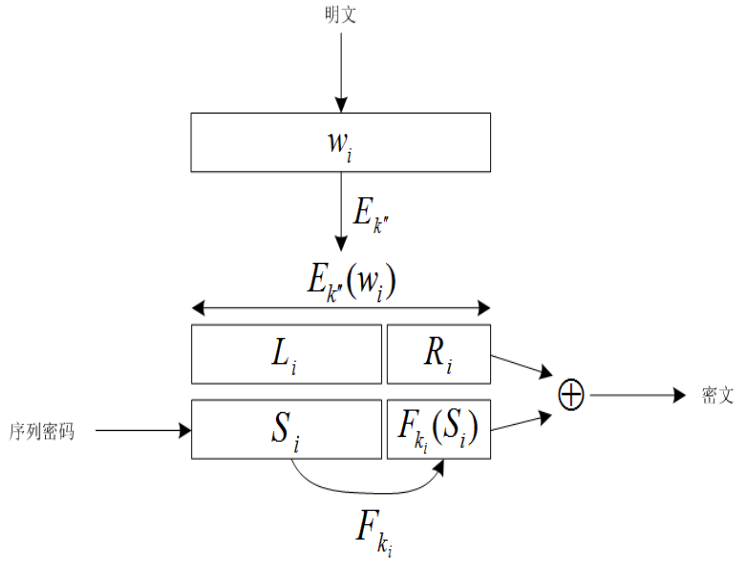


图3-3 基于全文扫描的方案示意图

3.3.2 基于全文扫描的方案

方案的具体步骤概述如下，该方案构造的密文和陷门与伪随机数具有不可区分性：

- **Setup算法：** 数据所有者生成密钥k', k''，伪随机数$S_1, S_2, \ldots, S_l$，伪随机置换E以及伪随机函数F, f；
- **BuildIndex算法：** 假设文档的内容为关键词序列$w_1, w_2, \ldots, w_l$。对于关键词w_i，数据所有者首先将其加密得到$E_{k''}(w_i)$，并将$E_{k''}(w_i)$拆分为L_i和R_i两个部分；然后，使用伪随机数S_i计算$F_{k_i}(S_i)$，其中$k_i = f_{k'}(L_i)$；最后，将$(S_i, F_{k_i}(S_i))$与(L_i, R_i)经过异或运算生成密文块C_i；
- **GenTrapdoor算法：** 当需要搜索关键词w时，数据所有者将$E_{k''}(w) = (L, R)$以及$k = f_{k'}(L)$发送给服务器；
- **Search算法：** 服务器依次将密文C_i与$E_{k''}(w)$进行异或运算，然后判断得到的结果是否满足$(S, F_k(S))$的形式。如果满足，则说明匹配成功，并将该文档返回。

3.3.2 基于全文扫描的方案

基于全文扫描的方案需要对每个密文块进行扫描并计算，在最坏的情况下，**检索一篇文档的时间与该文档的长度呈线性关系，检索效率较低**。

目前，人们主要集中于研究**基于文档-关键词索引和基于关键词-文档索引的密文关键词检索方案**，其将索引从密文中独立出来，即数据本身可以采用任意加密算法加密，检索功能由索引实现。

3.3.3 基于文档-关键词索引的方案

基于文档-关键词索引的密文检索方案，其核心思路是**为每篇文档建立单独的索引**，且服务器在检索时需要遍历全部索引，因此，这类方案的**检索时间复杂度与文档数目成正比**。

本小节首先介绍一种**基于布隆过滤器的密文关键词检索方案**，然后介绍选择关键词语义安全IND2-CKA以及模拟安全性。

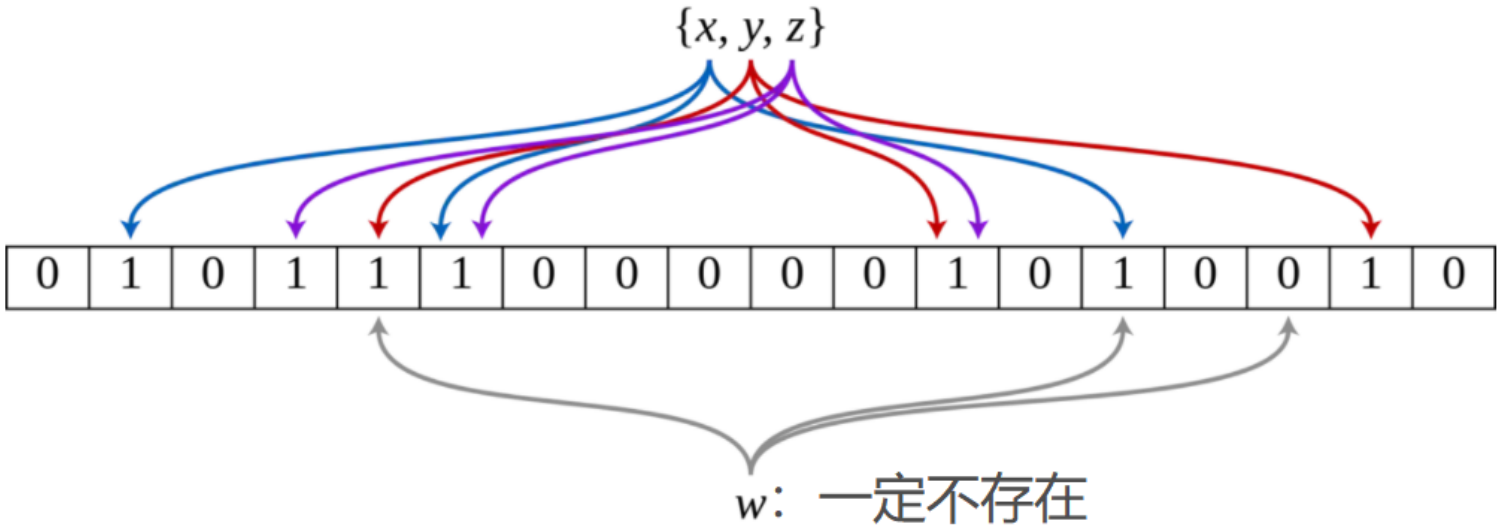
3.3.3 基于文档-关键词索引的方案

布隆过滤器利用位数组表示集合，并可以快速判断一个元素是否属于该集合。记位数组的长度为m，集合为$S = \{x_1, \dots, x_n\}$。首先，构造各位置均为0的初始数组BF，并选取k个Hash函数$h_1, \dots, h_k$，这些Hash函数可以将集合中的元素映射到位数组中的某一位。然后对于各元素x_i，为其计算k个Hash值$h_1(x_i), \dots, h_k(x_i)$，并将位数组中的相应位置设为1。

1	0	0	1	1	1	1	1	0	…	值, 0或1
0	1	2	3	4	5	6	7	8	…	索引

当想要判断元素y是否属于集合S时，我们同样使用Hash函数$h_1, \dots, h_k$为其计算k个值$h_1(y), \dots, h_k(y)$，如果位数组BF中的相应位置均为1，我们则认为y是S中的元素。

3.3.3 基于文档-关键词索引的方案



但实际上，由于Hash函数的计算结果可能存在冲突， y有可能并不属于S 。如图3-4所示的例子，如果$h_1(y) = h_2(x_i)$ 且$h_2(y) = h_1(x_j)$ ，则会发生误判。

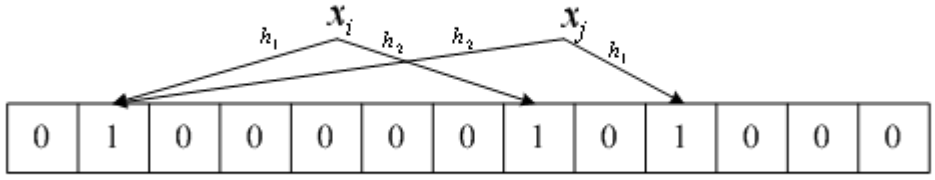


图3-4 布隆过滤器示意图

3.3.3 基于文档-关键词索引的方案

基于布隆过滤器的密文关键词检索方案ε：使用布隆过滤器为每篇文档分别构造索引，并使用伪随机函数为每个关键词计算两遍伪随机数，其一将关键词作为输入，其二将文档标识作为输入，从而同一关键词在不同文档中的计算结果不一致。具体方案概述如下：

- Setup算法：数据所有者生成r个密钥$k_1, \dots, k_r$以及伪随机函数f；
- BuildIndex算法：对于包含t个关键词$w_1, \dots, w_t$的文档D，数据所有者首先为其生成一个位数组BF(D)，并置BF(D)所有位均为0。然后，对于每个关键词w_i：
  - 以关键词w_i作为输入计算r个值：$x_1 = f(w_i, k_1), \dots, x_r = f(w_i, k_r)$；
  - 以文档标识id作为输入计算r个值：$y_1 = f(id, x_1), \dots, y_r = f(id, x_r)$；
  - 将BF(D)中$y_1, \dots, y_r$这r个值对应的位置设为1，并对BF(D)进行随机填充；
- GenTrapdoor算法：数据所有者为检索关键词w计算r个值：$x'_1 = f(w, k_1), \dots, x'_r = f(w, k_r)$，然后将这$r$个值发送给服务器；
- Search算法：根据陷门，服务器为文档D计算r个值$y'_1 = f(id, x'_1), \dots, y'_r = f(id, x'_r)$，并检查D对应的索引BF(D)中，这$r$个值对应的位置是否都为1。若是，则说明文档D包含$w$，并将其返回给用户。

上述方案在检索判定时只需要计算若干次伪随机数，速度比基于全文扫描的方法提高很多。然而，由于布隆过滤器的特性，会有一定的概率返回不包含查询关键词的文档。

3.4 非对称密文检索

3.4.1 概述

3.4.2 BDOP-PEKS方案

3.4.3 KR-PEKS方案

3.4.4 DS-PEKS方案

3.4.5 扩展：多关键词检索、多对多PEKS

3.4.6 小结

3.4.1 概述

非对称密文检索是指数据所有者（数据发送者）和数据检索者（数据接收者）不是同一方的密文检索技术。与非对称密码体制类似，数据所有者可以是了解公钥的任意用户，而只有拥有私钥的用户可以生成检索陷门。一个典型的非对称密文检索过程如下：

- Setup算法：该算法由数据检索者执行，生成公钥PK和私钥SK；
- BuildIndex算法：该算法由数据所有者执行，根据数据内容建立索引，并将公钥加密后的索引和数据本身上传到服务器；
- GenTrapdoor算法：该算法由数据检索者执行，将私钥和检索关键词作为输入，生成相应的陷门（又称“搜索凭证”），然后将陷门发送给服务器；
- Search算法：该算法由服务器执行，将公钥、接收到的陷门和本地存储的索引作为输入，进行协议所预设的计算，最后输出满足条件的搜索结果。

3.4.2 BDOP-PEKS方案

考虑如下**应用场景：**

邮件发送者B在向邮件接收者A发送邮件时，首先使用A的公钥对邮件包含的各关键词$w_1, w_2, \ldots, w_n$分别构造相应的索引$C_1, C_2, \ldots, C_n$，并将其附在发送的消$E(msg)$后面，一同交由服务器存储。其中E为标准的公钥加密算法，msg为邮件内容。检索时，A使用自己的私钥为查询关键词w生成陷门T_w，并将其发送给服务器，从而服务器能够判断邮件中是否包含关键词w。在这个过程中，服务器无法获得关于邮件内容和查询关键词的任何有用信息。

3.4.2 BDOP-PEKS方案

BDOP-PEKS方案是基于BF-IBE实现的，其安全性可归结为BDH假设。给定两个阶为p的群G_1和G_2，双线性映射$e:G_1\times G_1\rightarrow G_2$，以及两个哈希函数$H_1$和$H_2$，其中$H_1$可以将输入值映射到群$G_1$。

Let $\mathbb{G}_0$ and $\mathbb{G}_T$ be two groups with the prime order p. A map $e:\mathbb{G}_0\times\mathbb{G}_0\rightarrow\mathbb{G}_T$ with the following properties is said to be bilinear:

- Bilinearity: For all $g,h\in\mathbb{G}_0$ and $a,b\in\mathbb{Z}_p$, we have $e(g^a,h^b)=e(g,h)^{ab}$.
- Non-degeneracy: There exists $g,h\in\mathbb{G}_0$ such that $e(g,h)\neq 1$.
- Computability: There is a polynomial time algorithm to compute $e(g,h)\in\mathbb{G}_T$ for any $g,h\in\mathbb{G}_0$.

3.4.2 BDOP-PEKS方案

BDOP-PEKS方案的**具体步骤**如下：

- Setup算法：A选择随机数$\alpha = Z_p^*$以及群G1的生成元g，输出私钥$sk = \alpha$以及公钥$pk = (g, h = g^{\alpha})$；
- BuildIndex算法：对于关键词w_i，B首先选取随机数$r = Z_p^*$，同时计算$t = e(H_1(w_i), h^r) \in G_2$，随后输出$w_i$对应的索引$C_i = (g^r, H_2(t))$；
- GenTrapdoor算法：对于查询关键词w，A使用私钥sk计算相应的陷门$T_w = H_1(w)^{\alpha} \in G_1$；
- Search算法：对于索引C_i =(A,B)，如果$H_2\big(e(T_w, A)\big) = B$，则匹配成功，否则，匹配失败。

3.4.2 BDOP-PEKS方案

在选择关键词攻击下的不可区分性安全IND-CKA的定义基于游戏或实验ExpIND-CKA，其具体步骤如下：

- 挑战者执行Setup算法得到公钥pk和私钥sk，将pk发送给攻击者；
- 攻击者自适应询问若干次陷门，即攻击者将查询关键词发送给挑战者，挑战者执行GenTrapdoor算法生成对应的陷门，并将其返回给攻击者；
- 攻击者将挑战关键词W_0和W_1交给挑战者。挑战者随机选取$b\in\{0,1\}$，并执行BuildIndex算法得到W_b对应的索引C_b，然后将C_b返回给攻击者；
- 攻击者在自适应询问若干次陷门后，输出判定值b'，如果b'=b，则表明攻击成功，否则，攻击失败。
- 攻击者的攻击优势定义为$Adv(A)=|2\cdot Pr[Exp^{IND\text{-}CKA}\rightarrow true]-1|$。

定理3-3 在随机预言模型下，BDOP-PEKS方案对选择关键词攻击语义安全。

3.4.3 小结

BDOP-PEKS拥有较低的通信量，但是加密和检索时都需要一次对运算，导致效率较差；KR-PEKS的服务器检索效率最优，但是为了抵抗恶意攻击，需要较大的服务器端存储量；DS-PEKS的检索和加密效率较高，但是服务器和用户之间的通信量较大。

表3-2 非对称密文检索方案对比

非对称密文检索方案	通信量	服务器检索效率	加密效率
BDOP-PEKS	|g|	1次对运算	1次对运算
KR-PEKS	6|g|	O(1)	O(k)
DS-PEKS	4|g|logN	O(N)	O(N)

注：|g|表示群中元素所占用的存储空间，N表示字典中的关键词个数，k为安全参数