第 42 卷 第 6A 期
2015 年 6 月

计 算 机 科 学
Computer Science

Vol. 42 No. 6A
June 2015

基于关键字的云加密数据隐私保护检索

俞志斌 周彦晖

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 云计算技术的蓬勃发展,使得越来越多的企业和个人将数据外包存储到云端并通过云服务提供对数据的管理。查询是用户访问云端数据获取信息的必不可少的操作。这样,保护用户查询隐私和云端数据隐私以及快速提供满足用户需求的查询结果成为了关键问题。私有信息检索可使查询用户和数据库持有者在双方私有信息互不泄露的情况下完成查询操作。针对现有私有信息检索方案在云环境、大容量数据中的局限性,提出一个基于同态密码体制以及 MapReduce 的计算安全信息检索协议并使用批量查询方法降低协议通信复杂度。将提出的协议作为构建块,利用完美哈希函数工具构建基于关键字的云加密数据检索方案,方案兼具隐私性、查询的高效率以及基于关键字的实用性。

关键词 私有信息检索,加法同态加密,并行查询处理,关键字检索

中图法分类号 TP391.3 文献标识码 A

Keyword-based Privacy-preserving Retrieval over Cloud Encrypted Data

YU Zhi-bin ZHOU Yan-hui

(School of Computer and Information Science, Southwest University, Chongqing 400715, China)

Abstract More and more organizations and individuals outsource their data storage into cloud and using cloud-based services provide data management. Query providing user capabilities of accessing cloud data and obtaining information, is an important component of cloud services. Thus, how to protect both privacy of user queries and data privacy in cloud while providing query services and eventually providing query results meeting needs quickly and timely has become a vital concern. The private information retrieval (PIR) protocol allows user to perform a query from a database without revealing his private information; meanwhile the privacy of the database will be protected. Corresponding to limitations encountered in the applications of existing PIR protocol in the cloud environment and large capacity data scenario, a computational PIR protocol based on additively homomorphic encryption scheme and MapReduce was proposed. The protocol uses the batch queries method to lower the communication complexity. Further, we used proposed protocol as the building-block as well as perfect hash function tool to construct a cloud keyword-based encrypted data retrieval scheme. The scheme combines high query efficiency and the practicality of keyword-based retrieval.

Keywords Private information retrieval, Additively homomorphic encryption, Parallel query processes, Keyword-based retrieval

1 引言

以资源动态分配、按需服务为理念的云计算技术[1]可使用户便利地获取可扩展、高可用、按需分配的计算和存储资源。基于云端存储和处理的巨大便利和优势,越来越多的企业以及个人将自己的数据以及应用外包到云中。然而,用户数据由云服务商进行存储和管理,处于用户不可控域中,用户面临着数据安全和隐私泄露的风险[2,3,10]。缺乏安全和隐私保障已成为企业和政府机构采用云计算的主要障碍[4,5]。

云环境下用户一般通过向云平台发送查询请求来获取云端存储的信息。考虑如下场景,某零售商想评估某地一定年龄段的目标客户群,人口数据信息存储于云端,在构造查询时涉及区域名称、年龄段区间等敏感信息即查询隐私,零售商不想将这些利益攸关的信息泄露给云端,同时也不希望云端通过查询推导出自己的隐私信息,比如云端通过查看零售商检索的人员记录信息推断该人员为零售商的潜在客户。用户希望对数据的搜索标准(searching criterion)[9,23]、访问模式(access pattern)[9,23]、检索的结果等保持隐私性。

私有信息检索(Private Information Retrieval, PIR)技术[7-9]通过隐藏访问模式解决用户和云端查询交互的隐私和信息泄露问题。在 PIR 中,云端数据库存有 n 个长度为 l 的比特串,PIR 允许用户"茫然"地获取一个位串而不向云端泄露具体哪个串被检索。现有 PIR 方案[7,11,14]的构建主要基于数学上的困难假设,使得服务器端无法在多项式时间内获得查询请求内容信息从而实现保证用户隐私的查询。但是,使用这种方案服务端需要对整个数据库数据进行昂贵的密码操

本文受科技支撑计划项目(2012BAH77F02,2012BAH77F05)资助。

俞志斌(1987—),男,硕士生,主要研究方向为信息安全、云计算,E-mail:soberyu@126.com;周彦晖(1972—),男,副教授,硕士生导师,主要研究方向为信息安全、软件工程,E-mail:xiaohui@swu.edu.cn。

作,在云端包含大容量数据情况下,需要大量的处理开销,导致查询效率低下,难以满足基于互联网连接的交互用户的查询响应时间要求[12]。在有效使用查询服务以及保证隐私前提下快速产生查询结果响应用户成了关键需求。

现有的大量 PIR 研究主要基于理想模型,云端数据被索引好且用户知道目标数据位置[7,8,12]。现实应用环境很难满足这样的要求,协议的实用性较差。对现有协议进行扩展使用户可通过关键字检索云端数据,同时向云端隐藏检索关键字能提高协议的实用性。

本文对具有加法同态特性的加密体制[20]实现的计算安全 PIR 协议[9,10]进行基于 MapReduce 的并行扩展,利用 MapReduce 的"并行(parallelization)"和"聚集(aggregation)"实现 PIR 协议查询处理的并行化,提高查询效率,使之能用于云环境大容量数据快速查询,我们称该协议为 hPIRMR。进一步地提出批量查询降低协议通信复杂度。针对协议的实用性问题将 hPIRMR 作为构建块并利用完美哈希函数工具构建基于关键字的私有检索方案。方案可保障查询效率、云端数据安全以及用户查询关键字隐私。

2 相关工作

2.1 私有信息检索

私有信息检索协议用于保护用户的数据访问隐私,它是用户和存有 n 个数据元素的云端服务器的一系列交互,结果是用户能检索获取一个数据库元素而向云端服务器隐藏该元素被检索以及检索的结果,更形象化的描述是云端服务器不能以大于 $1/n$ 的机率猜对哪个元素被用户所查询。现有 PIR 主要可分为信息论的私有信息检索[6]计算性的私有信息检索(Computational PIR)[7,17]、基于安全硬件的私有信息检索[18]。这里只介绍与本文相关的计算性的私有信息检索,表示为 cPIR。

定义 1(cPIR_n^l 协议) 服务端存有 n 个大小为 l 比特的串 $DB=\{x_1,x_2,x_3,\cdots,x_n\},x_i\in\{0,1\}^l$,协议 $P=(Q,R,D)$ 包含 3 个算法:Q 为用户查询生成算法,R 是服务器端查询响应算法,D 是用户结果重构算法。

Q 基于用户想要查询的元素 j 生成一种类型查询,在不感兴趣数据项上生成另一种类型查询,构成查询向量并发送给服务器端。查询向量元素基于密码原语生成,服务器端计算能力有限(bounded computational power),在多项式时间内服务端无法区分两种类型的查询,且服务器是半诚实的(honest-but-curious);根据用户生成的查询请求向量与数据库元素执行 R,进行"茫然"查询处理,生成查询响应传输给用户;D 将服务器端回复转换为正确的明文查询结果。

Kushilevitz E 等[22]基于数论中的二次剩余假设提出了第一个 cPIR 协议。该方案具有很高的通信复杂度,为 $O(n^\lambda)$,其中 $\lambda>0$,并且服务端生成查询响应需要进行大量的密码学操作。现有的大量研究关注协议通信复杂度的优化,像 Chang Y C[14]将 Paillier 加法同态加密体制[20]用于构造 cPIR,其通信复杂度为 $O(\log n)$;Lipmaa H[18]基于合成剩余假设构造了通信复杂度为 $O(\log^2 n)$ 的协议。在 cPIR 协议中服务器端的计算复杂度为 $O(n\cdot l)$[22],因为响应生成算法 R 必须处理服务端存储的每个比特位,即使是一个数据位被忽略服务端就能推断该信息用户不感兴趣,查询隐私被破坏。大量研究者对 cPIR 方案的客户端和服务端的通信复杂度作了优化,但是方案局限性在于昂贵的计算开销导致服务端响应生成算法 R 需要大量的处理时间,查询效率低下,很难达到客户对响应时间的要求。

2.2 基于同态密码体制的 cPIR 与 MapReduce

第一个基于同态密码体制的 cPIR 协议由 Stern[19]提出,方案将 PIR 查询处理看作服务器端执行的一系列同态操作。

定义 2(加法同态加密体制) (ε,D) 为一个加法同态加密体制,ε 是一个概率加密算法,D 为相应的解密算法,两个消息 a 和 b 加密为 $\varepsilon(a)$、$\varepsilon(b)$,服务端无法区分密文。G 和 H 分别明文和密文群,大多数情况 $G\subseteq Z,H\subseteq Z^*$,"+"及"·"为明文群 G 和密文群 H 上加法和乘法操作的表示。加法同态加密体制有如下的性质:

$$D(\varepsilon(a)\cdot\varepsilon(b))=D(\varepsilon(a))+D(\varepsilon(b))=a+b \tag{1}$$

$$D(\varepsilon(a)^c)=ca \tag{2}$$

具有上述性质的加密方案可称为 Paillier 加密方案[20]。基于同态加密体制的 cPIR 协议可描述为如下步骤。

1. $Q(j)$ 为用户根据想要查询元素的索引 j,构造查询向量 $\{q_i\}_{i=1}^n$ 并传送给服务器。

$$q_i=\varepsilon(1),i=j,q_i=\varepsilon(0),\forall i\neq j \tag{3}$$

上式有 $q_i\in H$,基于对加密体制的假设,查询向量中的每个元素是不可区分的,服务器端无法获取关于查询的信息。

2. $R(DB,\{q_i\}_{i=0}^{n-1})$ 为服务器端根据收到的查询向量进行查询处理,执行式(4)的操作,生成返回给用户的查询响应。

$$R=\prod_{i=0}^{n-1}q_i^{x_i} \tag{4}$$

查询的效率主要取决于该阶段在服务端大容量数据上高开销的计算。

3. $D(R)$ 为用户解密服务端的响应提取查询结果,对服务器端响应进行如下操作:

$$D(R)=D(\prod_{i=0}^{n-1}q_i^{x_i})=\sum_{i=0}^{n-1}D(q_i)\cdot x_i \tag{5}$$

由同态性质进一步约简

$$\sum_{i=0}^{n-1}D(q_i)\cdot x_i=D(q_j)\cdot x_j=x_j \tag{6}$$

式(5)的成立应用了同态密码体制的性质(1)和(2),由于 $i\neq j$ 时 $D(q_i)=0$,故式(6)成立,用户滤去了不感兴趣的元素,巧妙获取了需要的 x_j。

步骤 2 中云端的查询处理任务是计算密集型的,式(4)的计算开销成了 PIR 协议的瓶颈。进一步观察云服务端式(4)的处理涉及模指(modulo exponentiation)运算,可将云端数据元素看作字节流,将其切分为小数据块,对小数据块进行模指运算处理,再进行聚集操作。这种处理方式可映射到云服务商广泛提供的 MapReduce 计算模型[21],云端并行查询处理提高了检索效率

通过将式(4)操作映射到 MapReduce 计算框架的"并行"和"聚集"两个阶段[13,15],分别编写 MapReduce 程序的两个函数:Map 和 Reduce 函数。Map 函数以 Key/Value 数据对作为输入,将输入数据经过业务逻辑计算产生仍旧以 Key/Value 形式表示的中间结果数据。Reduce 函数接收 Map 阶段传输过来的某个 Key 值以及对应的若干 Value 值等中间数据,函数逻辑对这个 Key 对应的 Value 内容进行处理。

3 基于同态密码体制和 MapReduce 的 cPIR 协议

加法同态密码体制的 cPIR 协议服务端查询处理可并行

处理[13,15]，打破 cPIR 方案服务端计算瓶颈，能有效解决查询效率低下问题。我们基于具有加法同态特性的 Paillier[20] 加密方案和 MapReduce 实现一个 cPIR 协议。该协议在保持传统方案保护隐私、通信复杂度不变情况下，利用 MapReduce 框架并行处理提高检索效率，能高效应用于云环境大容量数据的隐私保护检索，该协议被称为 hPIRMR。

3.1 hPIRMR 协议

hPIRMR 利用 MapReduce 计算框架以及云端的存储、计算资源，将分割的小数据集分发给 MapReduce 云中的不同工作节点，在大量工作节点上并行运行 PIR 实例，加速 cPIR 的响应生成，使 PIR 查询能快速地响应交互用户的请求。

数据持有者利用云服务商提供的接口上传的外包数据集，数据集为 n 个数据文件 $dataset=\{f_1,f_2,\cdots,f_n\}$，比如电子邮件、个人文档等，每个文件 l 比特，文件可存储在云端分布式文件系统中。此外，令 k 为数据文件分割成等长数据片(piece)的大小，为便于描述，考虑文件大小相同的情况，但可通过填充很容易地将其扩展为可变大小文件的场景。协议具体过程描述如下。

1. $Q(n,j)$ 为构造查询生成，并发送给云端。查询向量生成方法如 2.2 节步骤 1 中(3)所示，j 为需检索文件索引，此处 ε 是具有加法同态特性的 Paillier[20] 加密方案。

2. $R(DB,\{q_i\}_{i=1}^{n})$ 为云端生成响应，发送给用户。云端将文件“虚拟”地组织为类似表 1 的形式。云端将每个文件 i 分割成 l/k 个数据片 $\{B_{i,1},B_{i,2},\cdots,B_{i,l/k}\}$，并且每个数据片与查询向量作模指运算 $B'_{i,j}=q_i^{B_{i,j}}$。云端将相同的列的计算结果相乘创建结果向量 $R=(R_1,R_2,\cdots,R_{l/k})$，其中 $R_j=\prod_{i=1}^{n}q_i^{B_{i,j}}$，$1\leqslant j\leqslant l/k$，每个元素大小为 k，向量 R 的总大小为 l bit，可认为 R 是文件 f_x 的加密。向量 R 返回给客户端。

表 1 云端将文件分割成片

	1	2	3	…	l/k
$q_1^{\wedge}\rightarrow file_1$	$B_{1,1}$	$B_{1,2}$	$B_{1,3}$	…	$B_{1,l/k}$
$q_2^{\wedge}\rightarrow file_2$	$B_{2,1}$	$B_{2,2}$	$B_{2,3}$	…	$B_{2,l/k}$
…	…	…	…	…	…
$q_n^{\wedge}\rightarrow file_n$	$B_{n,1}$	$B_{n,2}$	$B_{n,3}$	…	$B_{n,l/k}$
	R_1	R_2	R_3	…	$R_{l/k}$

3. $D(R)$ 为用户重构恢复出需要的文件。恢复方式如 2.2 节步骤 3 的式(6)所示，对 R_j 解密，所有解密之后合并可得到需要的文件。

云端执行两个操作：查询向量与各文件片的模指运算以及每列的乘法运算。这两个操作各自可映射到 MapReduce 计算框架的 Map 和 Reduce 函数实现。上传到云端的文件集被均匀分发到所有参与节点中，每个 Mapper 会收到一个或者多个文件，Map 函数将文件以片为单位顺序读取文件 k 比特与对应的查询向量元素作模指运算，Mapper 中 Record Reader 读取的过程不解释数据的内容，独立于文件内容及格式，仅以字节流读取，故上传云端的文件可以是密文形式。云端各 Mapper 的输出为 Key/Value 对集合，其中 Key 是块索引或称为块位置 j，范围为 $(1\sim l/k)$，Value 为查询向量元素作模指运算的值 $q_i^{B_{i,j}}$。Reducer 接受同一个 Key 值集合即相同 j 的所有值，将这些同一 Key 的值相乘得到最终值。整个处理视图如图 1 所示。用户构造查询向量以及云端两个处理算法如表 2 所列。

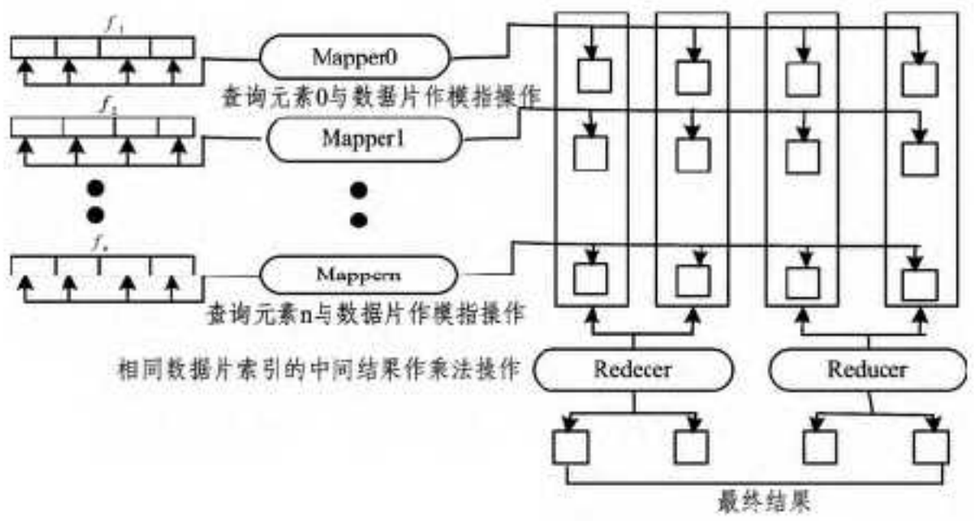

图 1 hPIRMR 的 Map 和 Reduce 处理视图

表 2 查询用户生成查询及云端 Map 阶段和 Reduce 阶段算法

查询用户

```
For each query user U:
    init q:= {},x
    for all i={1,…,n} do
      if i := x then qi :=ε(1)
      else
        qi :=ε(0)
      end
    end
  Emit q={ε(0)…ε(1)…ε(0)}
```

云端 Map 处理

```
For each Mapper M:
  init v :=1
  for all files in InputSplit(M) do
      read{Bi,1,Bi,2…,Bi,l/k}
      for all j={1,…,1/k} do
        v :=qiBi,j
        Emit(j,v)
      end
  end
```

云端 Reduce 处理

```
For each Reducer R:
init rValue :=1
for all {(j,v*)} in MapperOutput do
    for all i={1,…,n} do
        rValue :=rValue * vi
    end
  write{(j,rValue)}
end
```

用户通过 hPIRMR 方案在感兴趣元素上生成查询向量并上传，通过云端的查询处理，最终用户检索到了需要的文件 f_x，由协议过程描述的 2 和 3 步以及加法同态特性，无疑，hPIRMR 是正确的。

hPIRMR 方案继承了方案[18,19]的隐私特性。恶意云服务商可访问两方面的信息：用户上传的文件以及查询向量。上传的文件可以是公共数据文件或者是加密文件。因此，隐私性取决于查询向量 q。查询向量 q 由多个对“0”加密和一个对“1”加密构成。用户隐私保护地检索等价于对“0”加密和对“1”加密间的不可区分性(indistinguishing)，而 Paillier 加密方案[20]在合成剩余假设(Composite Residuosity Assumption)下被证明是安全的，因此 hPIRMR 方案能保护查询用户私有信息的隐私。

3.2 批量查询优化

一般的 PIR 方案检索 k 个数据库元素的方法是发送独立的 k 个 PIR 查询向量并且在服务端依次进行 PIR 处理，但存在的问题是需要在用户端和服务端传递大量的查询信息，导致通信复杂度很高。本文使用基于偏移的批量查询(batch

queries)优化[22]。在处理多查询时，重用单个 PIR 查询 k 次，用户基于第一个想要检索的元素索引 i_x 生成初始查询，后续查询基于初始查询生成。对于用户想要检索的后续元素 $\{i_t\}_{t=1}^{k-1}$，用户计算偏移量 $o_t = i_x - i_t$，将其存储为初始查询的后续查询。对于每个后续查询，数据库将元素"虚拟地"移动 o_t 个位置重新组织元素索引，这样初始查询就能有效应用于检索索引为 i_t 的元素。如图 3 所示，可将原始数据库元素"虚拟"复制 k 次并将元素移动相应的偏移量形成"虚拟"的 $k \times n$ 矩阵，这样用户使用初始查询向量就能隐私保护地检索矩阵 i_x 列的元素，即使用初始查询向量以及初始索引 i_x 完成批量的查询。

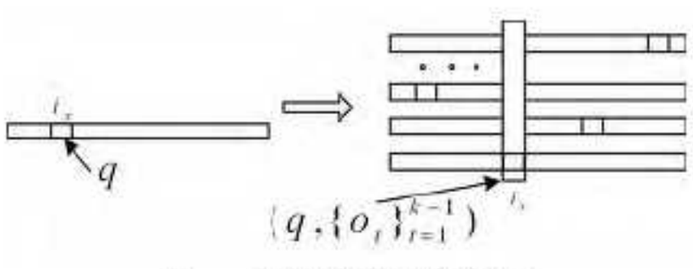

图 2　基于偏移查询视图

由于所用数据库元素索引是被 PIR 查询向量隐藏的以及加密方案保障了查询向量元素的不可区分性，恶意云端无法知道用户初始查询的是哪个元素，即无法知道用户查询起始点，所以这种优化是安全的。表 1 所列将数据库文件视为数据块元素，hPIRMR 可以很好地适应批量查询的优化，我们将在下面构造的方案中使用这种方法降低查询用户和云端的通信量。

4　外包数据基于关键字私有信息检索

对 PIR 协议的研究是基于理想的模型，用户检索是基于已索引的数据库元素且用户需要提前知道查询的目标数据的位置，这在实际应用中难以达到，导致协议实用性差。此外数据持有者将数据，如商业计划文件、电子邮件，外包到云端，用户希望通过关键字来检索云端数据获取信息，此外云端不完全可信使得对数据持有者外包数据进行保护成为必须。为提高协议实用性我们利用提出的 hPIRMR 协议以及公钥加密和完美哈希函数[24,25]工具，构建云加密文件基于关键字的检索方案。方案首先使用公钥体制实现数据持有者数据加密保护，同时对加密数据实现基于关键字的检索并向云端隐藏查询关键字、访问模式等用户私有信息。

4.1　方案详述

方案采用两步检索的策略，首先是基于关键字对包含关键字的文件地址的检索，其次，根据前一步骤获取的文件地址使用提出的 hPIRMR 协议对需要的文件进行检索。两步检索都使用批量查询优化，为进一步提高查询速度，在云端缓存不同用户的初始查询。

第一步使用完美哈希函数[24,25]将数据持有者设置的关键字集合映射到一个哈希值集合中，完美哈希函数将各关键字映射到唯一的哈希值，采用 Botelho F C 等[24]提出的外部完美哈希方案，相关内容参考文献[24,25]。在云端服务器中专门为用户建立一个长为 m 的索引表(Index Table)，索引表项(Table Slot)存储数据持有者外包到云端的文件的地址信息，索引表项中设置有一列表(list)结构存放多个地址，地址信息在索引表中存储的位置(Slot Position)由该文件关联的关键字的完美哈希值确定。

第二步根据获取的地址使用 hPIRMR 协议检索需要的文件，该文件为加密形式的，我们将存储加密文件的位置称为主数据库(MainDataBase)。方案涉及的两步都使用批量查询优化查询用户和云端通信量。

定义 3(云加密文件基于关键字检索)　云环境数据外包场景加密文件基于关键字检索，包括如下的概率多项式时间算法和协议。

$KeyGen(1^s)$：产生长度为 s 的公钥 pk 和私钥 sk。

$Send_{DO,COSP}(f, w_f, pk)$：一个两方的交互式协议，数据拥有者(Data Owner, DO)将文件 f 用公钥 pk 加密发送给云外包服务提供商(Cloud Outsourcing Service Provider, COSP)，文件设置关联的关键字集 w_f。f, w_f 是数据持有者的私有信息。

$Retrieval_{DU,COSP}(w, sk)$：数据用户(Data User, DU)和 $COSP$ 之间的一个两方协议，DU 检索与关键字 w 关联的所有文件，w, sk 是用户持有的私有信息。

为保证数据用户能使用关键字检索需要的文件，方案使用索引表建立关键字到文件地址的映射，将关键字经完美哈希函数作用得到哈希值并将此哈希值作为索引找到云端索引表的槽位(IndexTable Slot)，在其中的列表(list)中存储该文件地址，云端无法根据哈希值得到关键字。

结合以上的描述，方案整体流程如图 3 所示，数据持有者与云端有如下交互步骤，即 $Send_{DO,COSP}(f, w_f, pk)$ 交互过程。

第一步　数据持有者首先对要提交给云端的文件数据 f_i，比如电子邮件、商业计划文件等，设置关联的若干不同关键词 w。关键字对用户公布，并使用用户公钥对 f_i 加密形成 $e_{f_{ipk}}$，数据持有者上传加密文件存储到云端主数据库，反复执行直至所有文件完成上传。

第二步　云端数据库存储 $e_{f_{ipk}}$，并将该加密文件的存储地址 α_{f_i} 返回给数据持有者。

第三步　数据持有者将文件对应的关键字 w_u 输入到完美哈希函数产生 t 个哈希值输出，将 t 个哈希值 hs 作为索引表的索引地址即相当于数组下标，将文件地址 α_{f_i} 加密，加密的地址写入到索引表槽位的列表中。

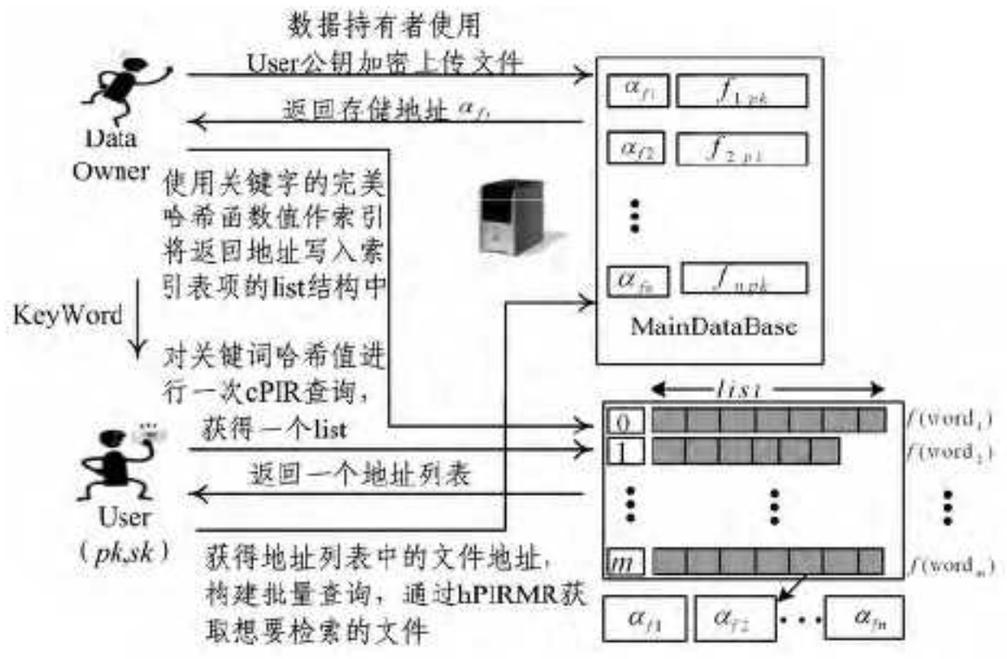

图 3　关键字搜索方案总体流程示意图

用户基于数据持有者设定的与文件相关联的关键词 w 检索数据持有者存储在云端的数据，用户与云端有如下的交互步骤，即 $Retrieval_{DU,COSP}(w, sk)$ 交互过程。

第一步　用户使用与数据持有者相同的完美哈希函数，输入一个关键词 w 产生哈希输出，这些输出是索引表的索引。

第二步　用户对哈希输出索引构造 cPIR 查询，获得对云端索引表的一个列表，查询中服务端不知道用户检索了哪个

列表，所得列表中包含用户要检索文件的地址信息。

第三步　从云端返回的列表中得到用户想要检索文件的地址信息 α_f'，α_f' 含有多个地址。

第四步　用户使用 α_f' 构造批量查询，在云端执行上文提出的 hPIRMR，得到包含关键字的文件，再对文件使用私钥进行解密。

4.2　方案讨论和分析

上述基于关键字检索采用两步策略：用户对索引表的检索以及对需要文件的检索。云端索引表是轻量级的，起到了链接关键字和文件存储地址的作用。第一步使用的方案为 2.2 节中介绍的 cPIR 方案。整个方案关键的优化可进行多关键字的检索，相比文献[23]有了突出的优势。此外，在云端我们缓存不同用户的初始查询，后续用户检索只需发送偏移即可，进一步降低了交互双方通信量。

对第一步，相异关键字总数 m 同时也是索引表的行数，设 t 为最长索引表中的 list 的长度。用户需要发送给云端 s 维查询向量 y，其中每个元素均为模 N 的数，长度为 $k$bits，云端对用户的索引表进行的 cPIR 总通信量为 $k(t+m)$。云端需要根据数据索引表中每一列的每一位元素对向量 y 进行模 N 的乘运算，因此用户检索索引表的总计算量为 $k \cdot s \cdot t$。

在第二步中，云端的计算由查询向量与 $n \cdot l/k$ 个数据片作模指运算以及执行 $(n-1) \cdot l/k$ 个乘法运算组成，方案的计算复杂度为 $O(n \cdot l \cdot E(k)+(n-1) \cdot l \cdot M(k))$，$E(k)$、$M(k)$ 分别是加法同态加密方案中模指、乘法运算的代价。通信复杂度由两部分组成，分别为用户向云端发送查询向量以及云端将查询响应返回给用户，Paillier 加密方案密文长度为 k，共有 n 个查询向量元素，用户到云端通信复杂度 $O(n \cdot k)$，云端响应用户查询 l/k 个数据项，每个为 $O(k)$，故云端到用户为 $O(l)$。总的通信复杂度为 $O(n \cdot k+l)$。由于两阶段的检索一定程度上提高了云端总体的计算开销并延长了查询响应时间，但是，用户的访问模式、查询内容、数据持有者的数据内容都保障了强隐私，同时基于关键字的检索保证了方案的实用性。

结束语　云计算在提供各种便利服务的同时，存在着严重的隐私泄露的风险。本文首先提出面向隐私保护、基于 MapReduce 的私有信息检索协议 hPIRMR。通过该协议用户从不可信云端检索需要的文件，协议在查询处理阶段将外包的大容量数据分发给云环境中的不同节点，使用 MapReduce 的“并行”和“聚集”阶段将查询并行处理，发挥云端强大计算能力，提高查询效率。由于协议使用 Paillier[20] 加密方案，查询处理需要模指运算，对云端大文件检索效率不高，未来对协议改进的方式是用更高效率同态加密方案。

使用提出的 hPIRMR 协议并借助完美哈希工具，构造云端加密文件基于关键字的检索方案，方案保证查询交互过程中的用户访问隐私、外包数据安全。该方案需要两次执行 PIR 协议，导致交互双方更多的通信量以及查询响应时间。进一步地，方案在处理多查询时可将检索索引表和执行 hPIRMR 以流水线方式进行，优化查询响应时间。

参 考 文 献

[1] Vouk M A. Cloud computing — Issues, research and implementations[C] // 30th International Conference on Information Technology Interfaces, 2008(ITI 2008). IEEE, 2008: 31-40

[2] Jansen W A. Cloud Hooks: Security and Privacy Issues in Cloud Computing[J]. Hawaii International Conference on System Sciences, 2011: 1-10

[3] Ren K, Wang C, Wang Q. Security Challenges for the Public Cloud[J]. Internet Computing, IEEE, 2012, 16(1): 69-73

[4] Bugiel S, Nurnberger S, Sadeghi A. Twin clouds: all architecture for secure computing [C] // Workshop on Cryptography and Security in Clouds. Zurich, Switzerland, 2011: 1-11

[5] Pearson S, Shen Y, Mowbray M. A Privacy Manager For Cloud Computing[J]. Cloud Computing, 2009, 5931: 90-106

[6] Chor B, Goldreich O, Kushilevitz E, et al. Private information retrieval[C] // Foundations of Computer Science, 1995. IEEE, 1995: 41-50

[7] Kushilevitz E, Ostrovsky R. Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval (Extended Abstract)[C] // PROC. of 38th Annu. IEEE Symp. on Foundation of Computer Science. 1997: 364-373

[8] Yoshida R, Cui Y, Sekino T, et al. Practical Searching over Encrypted Data by Private Information Retrieval[J]. Global Telecommunications Conference, IEEE, 2011: 1-5

[9] Rafail O, William E, Skeith III. A survey of single-database private information retrieval: techniques and applications[C] // Public Key Cryptography(PKC 2007). 2007: 393-411

[10] Boneh D, Kushilevitz E, Ostrovsky R, et al. Public Key Encryption That Allows PIR Queries[J]. Lecture Notes in Computer Science, 2007, 4622: 50-67

[11] Beimel A, Ishai Y, Malkin T. Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing[C] // CRYPTO 2000. 2000: 56-74

[12] Sion R. On the Computational Practicality of Private Information Retrieval[C] // Proceedings of the Network and Distributed Systems Security Symposium, 2007. Stony Brook Network Security and Applied Cryptography Lab Tech Report, 2007

[13] Kamara S, Raykova M. Parallel Homomorphic Encryption[J]. Lecture Notes in Computer Science, 2011

[14] Chang Y C. Single Database Private Information Retrieval with Logarithmic Communication[C] // Information Security and Privacy: 9th Australasian Conference, Sydney, Australia, 2004. Berlin, Germany: Springer, 2004: 50-61

[15] Pietro R D, Önen M, Blass E, et al. PRISM-Privacy-Preserving Search in MapReduce[J]. Privacy Enhancing Technologies, volume 7384 of Lecture Notes in Computer Science, 2012

[16] Melchor C A, Crespin B, Gaborit P, et al. High-speed private information retrieval computation on GPU[C] // IEEE SECURWARE. 2008: 263-272

[17] Beimel A, Ishai Y, Malkin T. Reducing the Servers Computation in Private Information Retrieval: PIR with Preprocessing[C] // CRYPTO 2000. 2000: 56-74

[18] Lipmaa H. An Oblivious Transfer Protocol with Log-Squared Communication[C] // Proc. Information Security Conf. (ISC '05). 2005

[19] Stern J P. A New and Efficient All-Or-Nothing Disclosure of Secrets Protocol[J]. Lecture Notes in Computer Science, 1998: 357-371

(下转第 401 页)

图 2 中，CPK 密钥管理中心为虚拟桌面系统的用户分发用于身份认证的 CPK 密钥。第一次登录的用户应先在 CPK 密钥管理中心申请获得 ID 证书，获得 ID 证书后用户向虚拟桌面管理服务器发送申请，请求登录虚拟桌面系统。虚拟桌面管理服务器为通过认证的用户分配虚拟机，然后将虚拟机序列号 UIID 与用户标识 UID 进行绑定，形成联合标识，并将其作为虚拟机的 CPK 密钥系统的标识 VMID，即 $VMID=UID \| UID$，CPK 密钥系统根据 VMID 为虚拟机生成 ID 证书。

虚拟机对用户进行认证流程如下（VM 为虚拟机，AS 为应用服务器）：

1）$U:R=\{UID, ASK_U'', r\}, sign=E_{csk_U}[Hash(R)]$；

2）$U\rightarrow VM:R \| sign$；

3）$VM:$从 R 中提取用户标识 UID，与 VMID 中的绑定的用户标识进行比对，若相等，则继续；

4）$VM:CPK_U'', sign^{-1}=E_{CPK_U}[sign]=Hash'(R)$；VM 根据标识 UID 计算 U 的二阶复合公钥，若 $Hash'(R)=Hash(R)$，VM 对 U 的注册身份认证通过。根据第 3）步可知该用户 U 即为虚拟桌面服务器为本虚拟机分配的使用者，所以 U 可以使用虚拟机访问相应资源。

用户 U 利用虚拟机 VM 访问外部应用服务器时，应用服务器对用户身份和虚拟机同时进行认证，由于虚拟机的标识为联合标识 VMID，通过对虚拟机的认证能够确定用户与虚拟机的绑定关系，再与用户身份认证相结合则能够有效防范虚拟机冒用问题。认证流程如下：

1）$VM:R_d=\{VMID, ASK_{VM}'', r\}, SIGN=E_{csk_{VM}}[Hash(R_d)]$

2）$VM\rightarrow AS:R_d \| R \| SIGN \| sign$；

3）$AS:CPK_{VM}'', SIGN^{-1}=E_{CPK_{VM}}[SIGN]=Hash(R_d)$，$CPK_U'', sign^{-1}=E_{CPK_U}[sign]=Hash'(R)$；应用服务器 AS 首先提取出 VMID 中的用户标识与 UID 进行比对，一致后计算 VM 和 U 的二阶复合公钥，并对 VM 和 U 的签名分别进行验证，若 $Hash'(R_d)=Hash(R_d)$，$Hash'(R)=Hash(R)$，同时完成对 VM 和 U 的认证。

整个虚拟资源使用过程中的认证流程如图 3 所示。

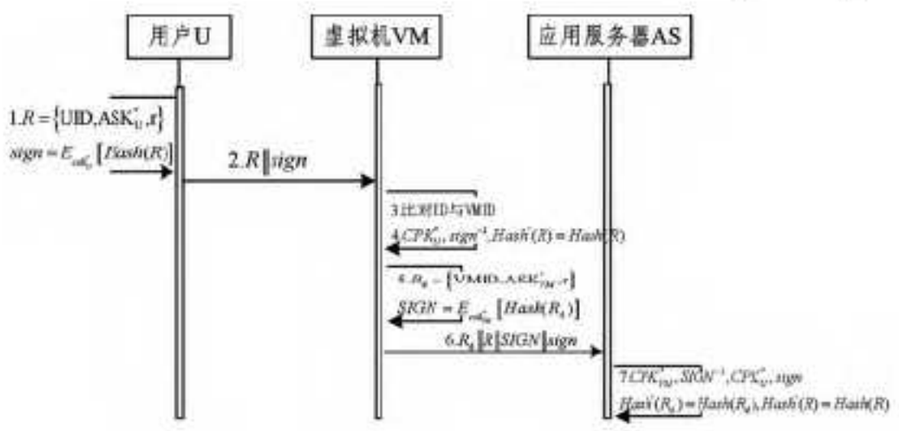

图 3 虚拟资源应用认证流程

5 安全与性能分析

在实用性方面，用户与虚拟云桌面服务器、用户与虚拟机、虚拟机与应用服务器之间认证方法均基于 CPK 的设计，因此虚拟桌面系统可进行统一管理和维护。CPK 具有密钥产生规模化、动态分发静态管理模式等优点，能实现超大规模的密钥分发，而且不需第三方在线参与，因此能够满足大量用户认证需求，不会形成性能瓶颈。

在安全性方面，所给出方法将用户标识和虚拟机 UIID 绑定成联合标识，可实现用户身份和虚拟机身份的绑定，若与用户认证相结合，则可有效防止虚拟机被攻击后的冒用问题。此外，虚拟云桌面系统中的所有认证均都基于 CPK 密码体制，鉴于 CPK 自身安全性，系统整体具有较高的安全性。

结束语　本文针对虚拟桌面的特有安全隐患，基于 CPK 设计了虚拟资源申请和虚拟资源使用两种场景下的认证方法，并给出了具体认证流程，最后对所提方法进行了安全和性能分析。身份认证是信息安全的基本技术，本文所提认证方法若与身份管理、策略管理相结合，则能够设计出更加有效的虚拟桌面防护方法。此外，CPK 本身特点使得所提方法能够支撑大用户量的认证需求，但由于所提方法为抵御虚拟机被攻击后的冒用问题，将用户与虚拟机进行了绑定，因此该方法适用于用户稳定，且虚拟资源分配较固定的场合。

参 考 文 献

[1] 郑志勇，吕远大，王毅. 虚拟桌面系统应用安全性分析与对策[J]. 网络安全技术与应用，2012，10(10)：50-52

[2] 孙宇，陈煜欣. 桌面虚拟化及其安全技术研究[J]. 信息安全与通信保密，2012，33(6)：87-88，92

[3] 宁芝，方正. 涉密信息系统虚拟化安全初探[J]. 保密科学技术，2012，22(2)：70-74

[4] 南湘浩. CPK 密码体制与网际安全[M]. 北京：国防工业出版社，2008

[5] 南湘浩. CPK 组合公钥体制(v8.0)[J]. 信息安全与通信保密，2013，34(3)：39-44

[6] 周加法，马涛，李益发. PKI、CPK、IBC 性能浅析[J]. 信息工程大学学报，2005，6(3)：26-31

[7] 王嘉林. 基于 PKI 和 CPK 的大规模网络认证方案的对比分析[J]. 保密科学技术，2012，6：44-49

[8] 汤维. 基于组合公钥密码体制的云安全研究[D]. 武汉：华中科技大学，2011

[9] 马宇驰，赵远，邓依群，等. 基于 CPK 的可信平台用户登录认证方案[J]. 计算机工程与应用，2010，46(1)：90-94

(上接第 369 页)

[20] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]// UROCRYPT. 1999：223-238

[21] Dean J，Ghemawat S. MapReduce：simplified data processing on large clusters[C]// Proceedings of 6th Conference on Symposium on Operatins Systems Design and Implementation (OSDI'04). 2004

[22] Ishai Y，Kushilevitz E，Ostrovsky R，et al. Batch Codes and Their Applications[C]// Proceedings of the 36th Annual ACM Symposium on Theory of Computing. 2004：262-271

[23] Yoshida R，Cui Y，Shigetomi R，et al. The Practicality of the Keyword Search Using Pir[C]// International Symposium on Information Theory and Its Applications(ISITA 2008). 2008：1-6

[24] Botelho F C，Galinkin D，Meira W，et al. External perfect hashing for very large key sets[C]// Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Mana(CIKM '07). 2007：653-662

[25] Botelho F C，Lacerda A，Menezes G V，et al. Minimal perfect hashing：A competitive method for indexing internal memory [J]. Information Sciences，2011，181(13)：2608-2625