云环境中支持隐私保护的可计算加密方法

黄汝维[1),2)] 桂小林[1)] 余 思[1)] 庄 威[1)]

[1)](西安交通大学电子与信息工程学院 西安 710049)

[2)](广西大学计算机与电子信息学院 南宁 530004)

摘 要 随着云计算的深入发展,隐私安全成为了云安全的一个关键问题.加密是一种常用的保护敏感数据的方法,但是它不支持有效的数据操作.为了提供云计算环境中的隐私保护,设计了一个基于矩阵和向量运算的可计算加密方案 CESVMC.通过运用向量和矩阵的各种运算,CESVMC 实现了对数据的加密,并支持对加密字符串的模糊检索和对加密数值数据的加、减、乘、除四种算术运算.安全分析和性能评估证明 CESVMC 是 IND-CCA 安全的,并能有效地实现对加密数据的计算.

关键词 云计算;向量和矩阵;字符串模糊检索;算术运算;可计算加密

中图法分类号 TP301 **DOI 号**: 10.3724/SP.J.1016.2011.02391

Privacy-Preserving Computable Encryption Scheme of Cloud Computing

HUANG Ru-Wei[1),2)] GUI Xiao-Lin[1)] YU Si[1)] ZHUANG Wei[1)]

[1)](*Department of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an* 710049)

[2)](*School of Computer and Electronics Information, Guangxi University, Nanning* 530004)

Abstract With the development of Cloud Computing, privacy has become the key problem of cloud security. Encryption is a well established technology for protecting sensitive data. But it makes effective data utilization a very challenging task. To solve the problem, we design a computable encryption scheme based on vector and matrix calculations (CESVMC). Through a variety of vector and matrix operations, CESVMC realizes the encryption of data, and supports fuzzy string search and basic arithmetic calculations (addition, subtraction, multiplication and division) on encrypted data. Security analysis and performance evaluation show that CESVMC is IND-CCA while realizing the goal of computations on encrypted cloud data effectively.

Keywords cloud computing; vector and matrix; fuzzy string search; arithmetic calculations; computable encryption

1 引 言

云计算作为一种新型的网络计算模式,以一种相比传统 IT 更经济的方式向用户提供按需的 IT 服务(计算、存储和应用等).由于云计算的发展理念符合当前低碳经济与绿色计算的总体趋势,它得到了世界各国政府和企业的大力倡导与推动,正带来计算领域、商业领域的巨大变革.

但在已经实现的云计算服务中,隐私安全问题一直令人担忧,并已经成为阻碍云计算发展和推广的主要因素之一.用户的隐私数据包括可用来识别

收稿日期:2011-06-20;最终修改稿收到日期:2011-10-31.本课题得到国家自然科学基金(60873071,91018011)、国家"八六三"高技术研究发展计划项目基金(2008AA01Z410)以及 IBM 共享大学研究资助项目(SUR201001X)资助.**黄汝维**,女,1978 年生,博士研究生,讲师,主要研究方向为服务计算、网络安全和同态加密.E-mail: ruweih@126.com. **桂小林**,男,1966 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为服务计算、网络安全、无线网络.E-mail: xlgui@mail.xjtu.edu.cn. **余 思**,男,1988 年生,博士研究生,主要研究方向为云计算、网络安全. **庄 威**,男,1989 年生,硕士研究生,主要研究方向为云计算、网络安全.

或定位个人的信息(例如电话号码、地址和信用卡号等)、敏感的信息(例如个人的健康状况、财务信息、公司的重要文件等).云计算的隐私安全问题源于云计算的数据外包和服务租赁的特点.用户数据存储到云环境中,人们失去了对数据的直接控制力,可能会导致个人隐私数据的泄露和滥用.而近年来发生的 Google、MediaMax 和 Salesforce.com 等云服务商泄露或丢失用户数据的事实证实了人们的担心[1].加密是一种常用的保护用户隐私数据的方法,但目前的大多数加密方案都不支持对密文的运算,如对加密的文件进行模糊检索、对加密的公司财务信息进行统计分析等,因而严重妨碍了云服务商为用户提供更进一步的数据管理和运算服务,从而削弱了云计算的优势.针对上述问题,本文提出了一个基于矩阵和向量运算的可计算加密方案 CESVMC (Computable Encryption Scheme based on Vector and Matrix Calculations).CESVMC 将云数据分为字符串和数值数据两大类,支持加密字符串的模糊检索和加密数值数据的加、减、乘、除四种基本算术运算,并保证了数据存储、运算过程中的隐私安全性.

本文第 2 节回顾相关研究的进展情况;第 3 节建立支持隐私保护的云计算模型并定义可计算加密技术;第 4 节具体介绍 CESVMC 的设计与实现;第 5 节和第 6 节分别就 CESVMC 的安全性和性能进行评估;第 7 节做出总结.

2 相关工作

可计算加密技术是一种加密方法,它通过加密保证数据安全,同时加密后的数据能够支持某些计算.目前已有的可计算加密技术可分为两类:支持检索的加密技术和支持运算的加密技术.

(1)支持检索的加密技术.Liu 等人[2]提出了一种基于对称加密的密文检索方法;Bonech 等人[3-5]提出了基于非对称加密的密文检索方法;Bellovin 等人[1,6-7]提出了基于 Bloom Filter 的密文检索方法.但这些方法只支持精确的字符串匹配,即两字符串是否相等.然而,在许多实际的情况下,错别字和格式不一致是不可避免的,因此,Li 等人[8]设计了一个支持加密字符串模糊检索的方案,它使用编辑距离来量化字符串的相似度,并为每个字符串附加一个基于通配符的模糊字符串组,用多个精确匹配来实现模糊检索.该方法的不足是它不能对满足检索条件的字符串进行相似度排序,而且计算、存储/通信负载很大.对于一个长度为 l 的字符串,为了能处理 d 位的错别字和格式不一致,需要进行 $O(l^d)$ 次 Hash 运算并产生 $O(l^d \times 160)$ 位的存储/通信负载.Wang 等人[9]提出一个基于保序加密技术 OPSE[10]的分级字符串检索方案,能够根据某一指标对检索的关键字分级,并按用户的要求返回前 N 个符合要求的结果.该方案要求数据拥有者在外包文件前对每个文件进行全文扫描,计算出每个关键字在该文件中的出现频率,这对于数据拥有者来说是一件非常麻烦的事情.Hacıgümüş 等人[11]提出了基于同态加密技术的密文聚集查询方案,但是要求数据拥有者自己建立一个加密的索引表.

(2)支持运算的加密方法.Agrawal 等人[12]提出一个基于桶划分和分布概率映射思想的保序对称加密算法 OPES,支持对加密数值数据的各种比较操作.Boldyreva 等人[10]提出一个基于折半查找和超几何概率分布的保序对称加密算法 OPSE,支持对加密数据的各种比较操作,但是由于在计算超几何概率时需要进行多次组合运算,其计算负载较大.以上两种保序加密算法都是确定性的加密方案,这使得它们不具有语义安全性.Wong 等人[13]设计了一个基于向量标量积的对称加密方案,该方案支持对加密数据库进行 KNN(k-nearest neighbor)计算.除此之外,目前已有一些同态加密算法,例如 unpadded_RSA、ElGamal、Goldwasser-Micali、Benaloh 和 Paillier 等,但它们只支持加法同态和乘法同态运算中的一种.Gentry[14-15]首次设计出了一种基于理想格的全同态加密方案,该方案能同时支持加法和乘法同态.之后,Smart 等人[16-17]对 Gentry 的工作进行了改进.但是目前已有的全同态方案都太复杂且计算量太大,还不适合应用到云计算的环境中.

根据以上分析,我们发现:(1)目前还没有一种加密方案能够同时支持字符串的检索和数值数据(包括整数和浮点数)的算术运算;(2)目前对加密字符串的模糊检索还没有一个实际可行的方案;(3)目前还没有一种支持密文运算的方案能轻松地同时解决整数和浮点数的加、减、乘、除法运算;(4)已有的一些方案往往要求数据拥有者在数据外包前做大量的准备工作,这会使用户的使用体验大打折扣.针对以上问题,我们设计了一个支持隐私保护的可计算加密方案 CESVMC(Computable Encryption Scheme based on Vector and Matrix Calculations).CESVMC 基于向量和矩阵运算,支持对加密字符串的模糊检索和对加密数值数据的加、减、乘、除法运算.

3　问题描述

3.1　支持隐私保护的云计算模型

如图 1 所示，支持隐私保护的云计算模型反映了数据拥有者(Owner)、用户(User)和服务提供者(Service Provider，SP)之间的交互，具体过程如下：

(1) Owner 用加密算法 E 对敏感数据 d_i ($i\in[1,n]$，$n\geq 1$)加密得到 $E(d_i)$，然后存储到 SP 的服务器上；

(2) User 获得 Owner 的授权后，对敏感计算参数($para$)加密得到 $E(para)$，并将 $E(para)$和计算要求($type$)提交给 SP；

(3) SP 验证 User 的权限，然后根据 User 的计算要求，对其权限范围的 $E(d_i)$和计算参数 $E(para)$进行计算，得到计算结果 $E(result)$，并将 $E(result)$返回给 User.

(4) User 对 $E(result)$进行解密，得到结果的明文 $result$.

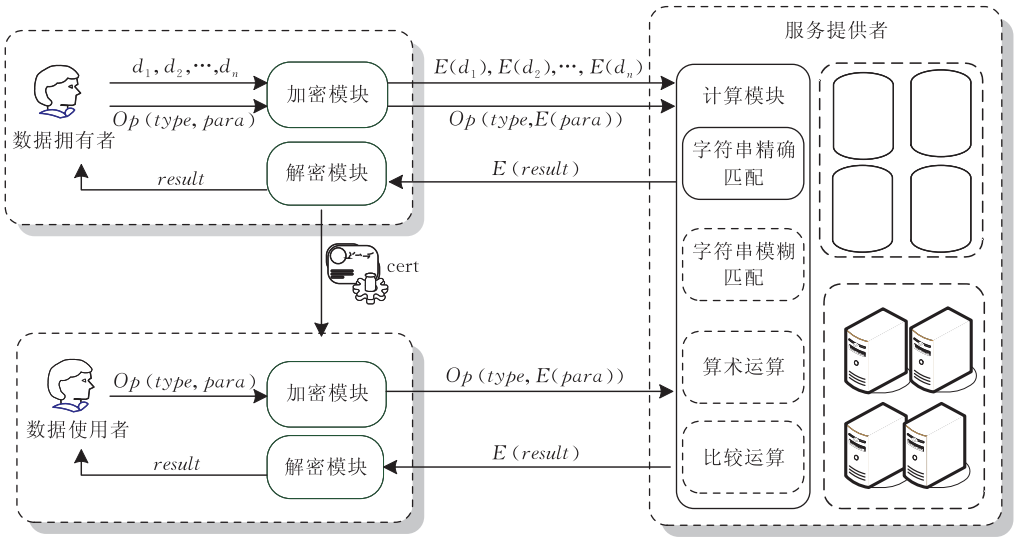


图 1　支持隐私保护的云计算模型

在这个过程中，由于 Owner 和 User 分别对敏感的外包数据和计算参数进行了加密处理，使得 Owner 和 User 的隐私得到了很好的保护. 但同时也带来了一个新的问题：SP 如何对加密的数据进行计算呢？如果这个问题不能得到有效的解决，Owner 和 User 就不能利用云计算中的计算资源对敏感数据进行处理，从而削弱了云计算的优势. 因此，本文提出的可计算加密方案是解决这一矛盾的关键技术.

3.2　可计算加密方案的定义

定义 1(可计算加密方案).　可计算加密方案 $\Sigma=(Gen,Enc,Dec,Cal)$由以下 4 个算法组成：

(1) 密钥生成算法 Gen 为用户 U 产生密钥 K，$K\leftarrow Gen(U,d)$，d 为安全参数；

(2) 加密算法 Enc 可能为概率算法，D 和 V 分别为该算法的定义域和值域，对于数据 $m\in D$，$c\leftarrow Enc(K,m)$且 $c\in V$；

(3) 解密算法 Dec 为确定算法，对于密文 c，$m\cup\{\perp\}\leftarrow Dec(c,K)$，$\perp$ 表示无解；

(4) 密文计算算法 Cal 可能为概率算法，对于密文集合$\{c_1,c_2,\cdots,c_t\}$，其中 $c_i\in V$，$Cal'(Dec(c_1,K),Dec(c_2,K),\cdots,Dec(c_t,K),op)\leftarrow Dec(Cal(c_1,c_2,\cdots,c_t,op))$，$op$ 为计算类型(例如模糊匹配、算术运算等)，Cal' 是与 Cal 对应的对明文数据运算的算法.

定义 2(正确性).　可计算加密方案 $\Sigma=(Gen,Enc,Dec,Cal)$是正确的：

(1) $\forall m\in D,\exists Dec(Enc(m,K))=m$；

(2) $\forall\{m_1,m_2,\cdots,m_t\}$，其中 $m_i\in D$，$\exists Cal'(m_1,m_2,\cdots,m_t,op)=Dec(Cal(Enc(m_1,K),Enc(m_2,K),\cdots,Enc(m_t,K),op))$.

定义 3(安全性).　可计算的加密方案 $\Sigma=(Gen,Enc,Dec,Cal)$是安全的：

(1) Σ 在提供外包数据的加密 Oracle 和解密 Oracle 的情况下是 IND-CCA 安全的；

(2) Σ 保证 SP 在进行 Cal 运算的过程中不能推断出原始明文、中间或最终结果的任何信息.

4　可计算的加密方案 CESVMC

4.1　CESVMC 定义

本节将构造一个基于向量和矩阵运算的可计算

加密方案 CESVMC. CESVMC 在确保 Owner 和 User 数据安全的前提下，支持加密字符串的模糊检索和加密数值数据的加、减、乘、除法运算. 下面对 CESVMC 进行定义.

定义 4(CESVMC). $CESVMC=(Gen,Enc,Dec,Cal)$由以下 4 个算法组成：

(1) 密钥生成算法 Gen：$\{M,W,S\}\leftarrow Gen(n,k)$，其中 n、k 分别表示计算元素个数和随机元素个数，且 $n,k\in N$；一个向量 $\boldsymbol{P}$ 由计算元素和随机元素组成，假设 d 表示 $\boldsymbol{P}$ 的维数，则有 $d=n+k$；Gen 生成的密钥由 3 部分组成，一个 $d\times d$ 可逆矩阵 $\boldsymbol{M}$，一个随机数数组 $W=\{w_1,w_2,\cdots,w_{k-3},w_{k-2}\}(k\geq 4,w_i\in R$ 且 $i\in[1,k-2])$和一个分裂串 $S=\{0,1\}^d$ 组成，分裂串只用于字符串中.

(2) 加密算法 Enc：假设 D 和 V 分别为 Enc 的定义域和值域，对于数据 $m\in D$，$\boldsymbol{P}\leftarrow Enc(m,\boldsymbol{M},W,S,U/O)$，其中 $\boldsymbol{P}$ 为 m 对应的 d 维密文向量且 $\boldsymbol{P}\in V$，U/O 表示是数据使用者或数据拥有者，用户身份不同时，加密算法略有不同.

(3) 解密算法 Dec：$m\leftarrow Dec(\boldsymbol{P},\boldsymbol{M},S,op)$，其中 op 是计算类型，当 $op=$“original”时，表示对原始数据的密文进行解密；当 $op=$“add”时，表示对加法运算的结果进行解密；当 $op=$“sub”时，表示对减法运算的结果进行解密；当 $op=$“mul”时，表示对乘法运算的结果进行解密；当 $op=$“div”时，表示对除法运算的结果进行解密.

(4) 密文计算算法 Cal：$P'\leftarrow Cal(P_1,P_2,\cdots,P_t,op)$，根据计算类型 op 的不同，对 $P_1,P_2,\cdots,P_t$ ($P_i\in V$ 且 $i\in[1,t],t\in N$)进行不同的运算，主要有模糊检索、加法、减法、乘法和除法等运算，P' 是计算结果的密文且 $P'\in V$.

4.2 算法描述

CESVMC 将云数据分为字符串和数值数据两类，支持加密字符串的模糊检索和加密数值数据的加、减、乘、除法运算.

4.2.1 字符串

为了更好地设计模糊检索机制，首先观察人们的查询习惯. CESVMC 的字符串模糊检索目前主要是针对英文数据，但其原理也可以应用到中文数据的模糊检索中. 根据英文单词的构成方式，人们经常认为“cloudy”与“clout”相比，“cloudy”的意思与“cloud”更为相近，因为“cloudy”以“cloud”为前缀，称之为前缀匹配；另外，对于大多数人来说，检索一个单词在一个句子或词组中的情况比该单词是构成另一个单词的一部分更有意义，例如单词“alone”在“be alone”中和它在单词“abalone”中，检索前者比后者更有意义，称之为关键字匹配. 因此，对一个字符串进行加密即是要构造它的前缀匹配密文和关键字匹配密文. 但在检索的时候，对前缀匹配密文和关键字匹配密文都可以采用前缀匹配的方法进行，从而实现模糊检索. 在进行模糊检索时将按照以下规则返回有序的检索结果：(1) 与检索参数一样的字符串构成了最匹配的结果集；(2) 以检索参数开始的字符串构成了第 2 匹配的结果集；(3) 假设检索参数的字符数为 m，前 $m-1$ 个字符与检索参数一致，但第 m 个字符与检索参数不一样的字符串构成了第 3 匹配的结果集，以此类推；(4) 在同一个结果集中，字符串与检索参数的第一个不同字符 ASCII 码值的差越小，它们就越相似.

4.2.1.1 字符串转换为向量

假设有一个字符串 np，Owner 首先生成 np 的子串. 其基本思想是按照空格将字符串分割，例如，字符串“cloud computing”对应的子串分别为“cloud”和“computing”. 然后，Owner 对 np 及其子串执行以下操作：假设计算元素组 ce 由 n 个元素组成，算法允许的字符串最大长度为 $len=(n-1)\times 6$，即每 6 个字符构成一个元素；np 的长度为 len'，则可形成前$\lceil len'/6\rceil$个元素，后面的$(n-\lceil len'/6\rceil)$个元素都为 0. 对于第 i 个元素，将其中每个字符转换为对应的 ASCII 码减去 23，从而确保了每一个字符都用一个两位数来表示；接着将每个字符的对应两位数连接起来，形成数字 v_i，再计算 $v_i'=v_i\times 10^m$，其中 $m=12\times(n-i-1)$. 通过以上运算，Owner 将字符串 np 转换为一个 $(n-1)$ 维的向量 $\boldsymbol{p}=(v_1',v_2',\cdots,v_{n-1}')$. 当 $len'>len$ 时，np 可以转换为$\lceil len'/len\rceil$个 $(n-1)$ 维向量. 由于每个 $(n-1)$ 维向量的后续操作都是一样的，因此，本文就假设 np 只转换成了一个 $(n-1)$ 维向量. 当 User 准备发出检索请求时，也是用同样的方法来转换检索参数.

4.2.1.2 外包字符串的加、解密

假设外包字符串 np_i 对应的 $(n-1)$ 维向量为 $\boldsymbol{p}_i$. Owner 创建一个 d 维向量 $\boldsymbol{p}_i'=(p_i,-0.5\times\|p_i\|^2,r_1,w_2,\cdots,r_{k-3},w_{k-2},-(\sum_{j=1}^{k/2-1}r_{2\times j-1}\times w_{2\times j-1}),1)^{\mathrm{T}}$，其中 $\|p_i\|^2$ 是 $\boldsymbol{p}_i$ 的标量积，r_j 是随机数且 $r_j\in R$. 也就是说，计算元素组 $ce=(\boldsymbol{p}_i,-0.5\times\|p_i\|^2)$，随机元素组为 $re=(r_1,w_2,\cdots,r_{k-3},w_{k-2},-(\sum_{j=1}^{k/2-1}r_{2\times j-1}\times w_{2\times j-1}),1)$. 然后 Owner 根据分裂串 S 将 p_i' 分割成

两个 d 维子向量 $\boldsymbol{p}'_{i1}$ 和 $\boldsymbol{p}'_{i2}$：假设 $S[z]$ 表示分裂串 S 的第 z（$z\in N$ 且 $z\in[1,d]$）位，如果 $S[z]$ 为‘0’，则 $\boldsymbol{p}'_i$ 的第 z 个元素 $p'_i[z]$ 分裂为 x_1 和 x_2，使得 $p'_i[z]=x_1+x_2$，分别作为 $p'_{i1}[z]$ 和 $p'_{i2}[z]$；如果 $S[z]$ 为‘1’，则 p'_i 的第 z 个元素 $p'_i[z]$ 不用分裂，$p'_{i1}[z]=p'_{i2}[z]=p'_i[z]$；加密向量 $\boldsymbol{p}'_{i1}$ 和 $\boldsymbol{p}'_{i2}$ 得到 $\boldsymbol{P}_{i1}=\boldsymbol{M}\times\boldsymbol{p}'_{i1}$ 和 $\boldsymbol{P}_{i2}=\boldsymbol{M}\times\boldsymbol{p}'_{i2}$，并将 $\boldsymbol{P}_i=(\boldsymbol{P}_{i1},\boldsymbol{P}_{i2})$ 存储到 SP 处.

对加密的外包字符串 $\boldsymbol{P}_i$ 解密是加密算法的逆过程，所以不再详述.

4.2.1.3　检索参数的加密

当 User 想检索字符串 nq 时，他首先选择一个随机数 r（$r>0$ 且 $r\in R$）并根据以上字符串转换方法生成对应的 $(n-1)$ 维向量 $\boldsymbol{q}$，然后将 $\boldsymbol{q}$ 扩充为一个 d 维向量 $\boldsymbol{q}'=r\times(\boldsymbol{q},1,w_1,r_2,\cdots,w_{k-3},r_{k-2},1,-(\sum_{j=1}^{k/2-1}r_{2\times j}\times w_{2\times j}))$，其中 r_j 是随机数且 $r_j\in R$. 也就是说，计算元素组为 $ce'=(\boldsymbol{q},1)$，随机元素组为 $re'=(w_1,r_2,\cdots,w_{k-3},r_{k-2},1,-(\sum_{j=1}^{k/2-1}r_{2\times j}\times w_{2\times j}))$. 然后 User 根据分裂串 S 将 $\boldsymbol{q}'$ 分割成两个 d 维子向量 $\boldsymbol{q}'_1$ 和 $\boldsymbol{q}'_2$：如果 $S[z]$ 为‘1’，则 $\boldsymbol{q}'$ 的第 z 个元素 $q'[z]$ 分裂为 x_1 和 x_2，使得 $q'[z]=x_1+x_2$，分别作为 $q'_1[z]$ 和 $q'_2[z]$；如果 $S[z]$ 为‘0’，则 $\boldsymbol{q}'$ 的第 z 个元素 $q'[z]$ 不用分裂，$q'_1[z]=q'_2[z]=q'[z]$；加密 $\boldsymbol{q}'_1$ 和 $\boldsymbol{q}'_2$ 得到 $\boldsymbol{Q}_1=\boldsymbol{q}'_1\times(\boldsymbol{M}^{-1})$ 和 $\boldsymbol{Q}_2=\boldsymbol{q}'_2\times(\boldsymbol{M}^{-1})$，并将 $\boldsymbol{Q}=(\boldsymbol{Q}_1,\boldsymbol{Q}_2)$ 提交给 SP.

4.2.1.4　加密字符串的检索

当 SP 接到检索请求时，将对 User 权限范围内的 P_i 进行如下运算：

$$\begin{aligned}Q\times P_i&=((q'_1,q'_2)\times M^{-1})\times(M\times(p'_{i1},p'_{i2}))\\&=(q'_1,q'_2)\times M^{-1}\times M\times(p'_{i1},p'_{i2})\\&=(q'_1,q'_2)\times(p'_{i1},p'_{i2})=q'_1\times p'_{i1}+q'_2\times p'_{i2}\\&=r\times(q,1,w_1,r_2,\cdots,w_{k-3},r_{k-2},1,\\&\quad-\sum_{j=1}^{k/2-1}r_{2\times j}\times w_{2\times j})\times\\&\quad(p_i,-0.5\parallel p_i\parallel^2,r_1,w_2,\cdots,r_{k-3},w_{k-2},\\&\quad-(\sum_{j=1}^{k/2-1}r_{2\times j}\times w_{2\times j}),1)^{\mathrm{T}}\\&=r\times(p_i\times q-0.5\parallel p_i\parallel^2+\sum_{j=1}^{2k}r_j\times\\&\quad w_j-\sum_{j=1}^{2k}r_j\times w_j)\\&=r\times(p_i\times q-0.5\parallel p_i\parallel^2)\end{aligned}\tag{1}$$

然后 SP 比较结果的值，其值越大，nq 与 np_i 越相似. 其原因如下：假设 $\boldsymbol{P}_1$ 和 $\boldsymbol{P}_2$ 是两个外包字符串 np_1 和 np_2 对应的加密向量，$\boldsymbol{Q}$ 是检索参数 nq 的加密向量：

$$\begin{aligned}\boldsymbol{Q}\times\boldsymbol{P}_1-\boldsymbol{Q}\times\boldsymbol{P}_2&=r\times(p_1\times q-0.5\times\parallel p_1\parallel^2-\\&\quad p_2\times q+0.5\times\parallel p_2\parallel^2)\\&=r\times(p_1\times q-0.5\times\parallel p_1\parallel^2-p_2\times q+\\&\quad 0.5\times\parallel p_2\parallel^2-0.5\times\parallel q\parallel^2+0.5\times\parallel q\parallel^2)\\&=-0.5\times r\times(\parallel p_1\parallel^2-2p_1q+\parallel q\parallel^2)+\\&\quad 0.5\times r\times(\parallel p_2\parallel^2-2p_2q+\parallel q\parallel^2)\\&=0.5\times r\times[d^2(p_2,q)-d^2(p_1,q)]\end{aligned}\tag{2}$$

其中，$d(\boldsymbol{p},\boldsymbol{q})$ 表示向量 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 的欧几里得距离. 由式(2)得

因为 $d(p_i,q)\geq0$ 且 $r\in R^+$

所以 $\boldsymbol{Q}\times\boldsymbol{P}_1-\boldsymbol{Q}\times\boldsymbol{P}_2=0.5\times r\times(d^2(\boldsymbol{p}_2,\boldsymbol{q})-d^2(\boldsymbol{p}_1,\boldsymbol{q}))>0\Leftrightarrow d(\boldsymbol{p}_2,\boldsymbol{q})>d(\boldsymbol{p}_1,\boldsymbol{q})$　(3)

所以 SP 可以通过对 P_i 和 Q 的标量积进行从大到小的排序从而得到一个按相似度由高到低排列的序列，从而实现了模糊检索.

4.2.2　数值数据

对数值数据的操作可以简单地分为 4 种基本算术运算：加法、减法、乘法和除法. 为了实现加/减法和乘/除法，计算元素组 ce 由加法因子 $addF$ 和乘法因子 $mulF$ 组成，其中 $addF$ 由 d_a（$d_a\in N$ 且 $d_a>3$）个元素组成，$mulF$ 由 d_m（$d_m\in N$ 且 $d_m\geq2$）个元素组成. 也就是说，$ce=(addF,mulF)$ 且 $n=d_a+d_m$.

4.2.2.1　数值数据的转换和加、解密

转换一个数值数据 np 为 d 维向量的过程可以分为以下 4 步：首先，选择 (d_a-1) 个随机数 $\{ar_1,ar_2,\cdots,ar_{d_a-1}\}$（$ar_i\in R$ 且 $i\in[1,d_a-1]$），并计算 $ar_{d_a}=np-\sum_{i=1}^{d_a-1}ar_i$，从而得到一个 d_a 维的向量 $\boldsymbol{p}=(ar_1,ar_2,\cdots,ar_{d_a})^{\mathrm{T}}$；再生成 d_a 个随机数 $\{cr_1,cr_2,\cdots,cr_{d_a}\}$（$cr_i\in R$ 且 $i\in[1,d_a]$），使 $\boldsymbol{p}$ 变为 $\boldsymbol{p}'=(ar_1+cr_1,ar_2+cr_2,\cdots,ar_{d_a}+cr_{d_a})^{\mathrm{T}}$. 然后，随机选择 (d_m-1) 个随机数 $\{mr_1,mr_2,\cdots,mr_{d_m-1}\}$（$mr_i\in R$ 且 mr_i 的倒数都是有限小数，$i\in[1,d_m-1]$），计算 $mr_{d_m}=np/\prod_{i=1}^{d_m-1}mr_i$. 这样，Owner 就将 $\boldsymbol{p}'$ 转换为一个 (d_a+d_m) 维的向量 $\boldsymbol{p}''=(ar_1+cr_1,ar_2+cr_2,\cdots,ar_{d_a}+cr_{d_a},mr_1,mr_2,\cdots,mr_{d_m})^{\mathrm{T}}$. 第 3 步，Owner 通过加入随机元素组 re 扩充 $\boldsymbol{p}''$ 从而构成了一个 (d_a+d_m+k) 维的向量 $\boldsymbol{p}'''=(ar_1+cr_1,ar_2+cr_2,\cdots,ar_{d_a}+cr_{d_a},mr_1,mr_2,\cdots,mr_{d_m},r_1,\cdots,r_{k-1},-\sum_{i=1}^{d_a}cr_i)^{\mathrm{T}}$，其中 $k\in N$ 且 $k\geq2$，r_j 是随机数且 $r_j\in R$

($j\in[1,k-1]$).最后,Owner 加密 $\boldsymbol{p}'''$形成了外包向量 $\boldsymbol{P}=\boldsymbol{M}\times\boldsymbol{p}'''$,并存储到 SP 的服务器上.对加密的外包数值数据 $\boldsymbol{P}$ 解密是加密算法的逆过程,所以不再详述.

User 将检索参数转换为向量的步骤与 Owner 的步骤一样.只是在进行加/减法运算时,User 可以设置 $mulF=(\underbrace{0,0,\cdots 0}_{d_m})$.由于外包数据和检索参数的转换过程是一样的,因此两个数值数据的运算,无论它们同时为外包数据,或者一个是外包数据一个是检索参数,或者两个都是检索参数,处理的方法都是一样.所以,我们假设两个数值数据 np 和 nq 对应的加密向量为 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$.接下来,我们将阐述如何实现各种算术运算.

4.2.2.2 加法

SP 直接对 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 执行加法运算:

$$\begin{aligned}\boldsymbol{P}+\boldsymbol{Q}&=\boldsymbol{M}\times\boldsymbol{p}'''+\boldsymbol{M}\times\boldsymbol{q}'''\\&=\boldsymbol{M}\times[(ar_{p1}+cr_{p1},ar_{p2}+cr_{p2},\cdots,ar_{pd_a}+cr_{pd_a},mr_{p1},mr_{p2},\cdots,mr_{pd_m},r_{p1},\cdots,r_{p(k-1)},-\sum_{i=1}^{d_a}cr_{pi})^{\mathrm{T}}+(ar_{q1}+cr_{q1},ar_{q2}+cr_{q2},\cdots,ar_{qd_a}+cr_{qd_a},mr_{q1},mr_{q2},\cdots,mr_{qd_m},r_{q1},\cdots,r_{q(k-1)},-\sum_{i=1}^{d_a}cr_{qi})^{\mathrm{T}}]\\&=\boldsymbol{M}\times((ar_{p1}+ar_{q1}+cr_{p1}+cr_{q1}),\cdots,(ar_{pd_a}+ar_{qd_a}+cr_{pd_a}+cr_{qd_a}),(mr_{p1}+mr_{q1}),\cdots,(mr_{pd_m}+mr_{qd_m}),(r_{p1}+r_{q1}),\cdots,(r_{p(k-1)}+r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}-\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\end{aligned}\tag{4}$$

然后,SP 将计算结果返回给 User.User 对结果解密:

$$\begin{aligned}\boldsymbol{M}^{-1}\times(\boldsymbol{P}+\boldsymbol{Q})&=\boldsymbol{M}^{-1}\times\boldsymbol{M}\times((ar_{p1}+ar_{q1}+cr_{p1}+cr_{q1}),\cdots,(ar_{pd_a}+ar_{qd_a}+cr_{pd_a}+cr_{qd_a}),(mr_{p1}+mr_{q1}),\cdots,(mr_{pd_m}+mr_{qd_m}),(r_{p1}+r_{q1}),\cdots,(r_{p(k-1)}+r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}-\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\\&=((ar_{p1}+ar_{q1}+cr_{p1}+cr_{q1}),\cdots,(ar_{pd_a}+ar_{qd_a}+cr_{pd_a}+cr_{qd_a}),(mr_{p1}+mr_{q1}),\cdots,(mr_{pd_m}+mr_{qd_m}),(r_{p1}+r_{q1}),\cdots,(r_{p(k-1)}+r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}-\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\end{aligned}\tag{5}$$

通过式(5),User 得到一个 d 维向量,于是它通过公式$(\sum_{j=1}^{d_a}(ar_{pj}+aqr_{qj}+cr_{pj}+cr_{qj}))+(-\sum_{i=1}^{d_a}cr_{pi}-\sum_{i=1}^{d_a}cr_{qi})$将向量的前 d_a 个元素和最后一个元素相加从而得到最后的结果.该方案支持在不解密的情况下进行多次加法操作.

4.2.2.3 减法

SP 直接对 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 执行减法运算:

$$\begin{aligned}\boldsymbol{P}-\boldsymbol{Q}&=\boldsymbol{M}\times\boldsymbol{p}'''-\boldsymbol{M}\times\boldsymbol{q}'''\\&=\boldsymbol{M}\times[(ar_{p1}+cr_{p1},ar_{p2}+cr_{p2},\cdots,ar_{pd_a}+cr_{pd_a},mr_{p1},mr_{p2},\cdots,mr_{pd_m},r_{p1},\cdots,r_{p(k-1)},-\sum_{i=1}^{d_a}cr_{pi})^{\mathrm{T}}-(ar_{q1}+cr_{q1},ar_{q2}+cr_{q2},\cdots,ar_{qd_a}+cr_{qd_a},mr_{q1},mr_{q2},\cdots,mr_{qd_m},r_{q1},\cdots,r_{q(k-1)},-\sum_{i=1}^{d_a}cr_{qi})^{\mathrm{T}}]\\&=\boldsymbol{M}\times((ar_{p1}-ar_{q1}+cr_{p1}-cr_{q1}),\cdots,(ar_{pd_a}-ar_{qd_a}+cr_{pd_a}-cr_{qd_a}),(mr_{p1}-mr_{q1}),\cdots,(mr_{pd_m}-mr_{qd_m}),(r_{p1}-r_{q1}),\cdots,(r_{p(k-1)}-r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}+\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\end{aligned}\tag{6}$$

然后,SP 将计算结果返回给 User.User 对结果进行解密:

$$\begin{aligned}\boldsymbol{M}^{-1}\times(\boldsymbol{P}-\boldsymbol{Q})&=\boldsymbol{M}^{-1}\times\boldsymbol{M}\times((ar_{p1}-ar_{q1}+cr_{p1}-cr_{q1}),\cdots,(ar_{pd_a}-ar_{qd_a}+cr_{pd_a}-cr_{qd_a}),(mr_{p1}-mr_{q1}),\cdots,(mr_{pd_m}-mr_{qd_m}),(r_{p1}-r_{q1}),\cdots,(r_{p(k-1)}-r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}+\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\\&=((ar_{p1}-ar_{q1}+cr_{p1}-cr_{q1}),\cdots,(ar_{pd_a}-ar_{qd_a}+cr_{pd_a}-cr_{qd_a}),(mr_{p1}-mr_{q1}),\cdots,(mr_{pd_m}-mr_{qd_m}),(r_{p1}-r_{q1}),\cdots,(r_{p(k-1)}-r_{q(k-1)}),(-\sum_{i=1}^{d_a}cr_{pi}+\sum_{i=1}^{d_a}cr_{qi}))^{\mathrm{T}}\end{aligned}\tag{7}$$

通过式(7),User 得到一个 d 维向量,于是它通过公式$(\sum_{j=1}^{d_a}(ar_{pj}-aqr_{qj}+cr_{pj}-cr_{qj}))+(-\sum_{i=1}^{d_a}cr_{pi}+\sum_{i=1}^{d_a}cr_{qi})$将向量的前 d_a 个元素和最后一个元素相加从而得到最后的结果.该方案支持在不解密的情况下进行多次减法操作.

4.2.2.4 乘法

首先,我们介绍实现乘法计算的原理.假设有两个数值数据 np 和 nq,它们对应的 6 维向量分别为

$\boldsymbol{v}_1=(x_1+c_1, y_1+c_2, a_1, b_1, r_1, -c_1-c_2)$ 和 $\boldsymbol{v}_2=(x_2+c_3, y_2+c_4, a_2, b_2, r_2, -c_3-c_4)$，其中 $np=x_1+y_1$，$np=a_1\times b_1$，$nq=x_2+y_2$ 和 $nq=a_2\times b_2$，c_1、c_2、c_3、c_4、r_1 和 r_2 是随机实数. 我们观察以下运算结果：

$$\boldsymbol{v}_1\times(\boldsymbol{v}_2)^{\mathrm{T}}=(x_1+c_1 \quad y_1+c_2 \quad a_1 \quad b_1 \quad r_1 \quad -c_1-c_2)^{\mathrm{T}}\times(x_2+c_3 \quad y_2+c_4 \quad a_2 \quad b_2 \quad r_2 \quad -c_3-c_4)=$$
$$\begin{pmatrix} x_1x_2+x_1c_3+c_1x_2+c_1c_3 & x_1y_2+x_1c_4+c_1y_2+c_1c_4 & x_1a_2+c_1a_2 & x_1b_2+c_1b_2 & x_1r_2+c_1r_2 & -x_1c_3-x_1c_4-c_1c_3-c_1c_4 \\ y_1x_2+y_1c_3+c_2x_2+c_2c_3 & y_1y_2+y_1c_4+c_2y_2+c_2c_4 & y_1a_2+c_2a_2 & y_1b_2+c_2b_2 & y_1r_2+c_2r_2 & -y_1c_3-y_1c_4-c_2c_3-c_2c_4 \\ a_1x_2+a_1c_3 & a_1y_2+a_1c_4 & a_1a_2 & a_1b_2 & a_1r_2 & -a_1c_3-a_1c_4 \\ b_1x_2+b_1c_3 & b_1y_2+b_1c_4 & b_1a_2 & b_1b_2 & b_1r_2 & -b_1c_3-b_1c_4 \\ r_1x_2+r_1c_3 & r_1y_2+r_1c_4 & r_1a_2 & r_1b_2 & r_1r_2 & -r_1c_3-r_1c_4 \\ -c_1x_2-c_1c_3-c_2x_2-c_2c_3 & -c_1y_2-c_1c_4-c_2y_2-c_2c_4 & -c_1a_2-c_2a_2 & -c_1b_2-c_2b_2 & -c_1r_2-c_2r_2 & c_1c_3+c_1c_4+c_2c_3+c_2c_4 \end{pmatrix} \tag{8}$$

根据式(8)可得到：

$$np\times nq=(a_1\times b_1)\times(a_2\times b_2)=(a_1\times a_2)\times(b_1\times b_2) \tag{9}$$

设 $matrix$ 表示 $\boldsymbol{v}_1\times(\boldsymbol{v}_2)^{\mathrm{T}}$ 的结果，我们可以通过公式 $\prod_{i=3}^{4} matrix[i][i]$ 来计算两个数的乘积. 根据以上分析，SP 可以对 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 进行如下操作：

$$\begin{aligned}\boldsymbol{P}\times(\boldsymbol{Q})^{\mathrm{T}}&=\boldsymbol{M}\times\boldsymbol{p}'''\times(\boldsymbol{M}\times\boldsymbol{q}''')^{\mathrm{T}}=\boldsymbol{M}\times\boldsymbol{p}'''\times(\boldsymbol{q}''')^{\mathrm{T}}\times\boldsymbol{M}^{\mathrm{T}}\\&=\boldsymbol{M}\times(ar_{p1}+cr_{p1}, ar_{p2}+cr_{p2}, \cdots, ar_{pd_a}+cr_{pd_a}, mr_{p1}, mr_{p2}, \cdots, mr_{pd_m}, r_{p1}, \cdots, r_{p(k-1)}, -\sum_{i=1}^{d_a} cr_{pi})^{\mathrm{T}}\times(ar_{q1}+cr_{q1}, ar_{q2}+cr_{q2}, \cdots, ar_{qd_a}+cr_{qd_a}, mr_{q1}, mr_{q2}, \cdots, mr_{qd_m}, r_{q1}, \cdots, r_{q(k-1)}, -\sum_{i=1}^{d_a} cr_{qi})\times\boldsymbol{M}^{\mathrm{T}}\end{aligned} \tag{10}$$

式(10)的结果是一个 $d\times d$ 矩阵. SP 将结果返回给 User. User 对结果进行解密：

$$\begin{aligned}\boldsymbol{M}^{-1}\times\boldsymbol{P}\times(\boldsymbol{Q})^{\mathrm{T}}\times(\boldsymbol{M}^{\mathrm{T}})^{-1}&=\boldsymbol{M}^{-1}\times\boldsymbol{M}\times(ar_{p1}+cr_{p1}, ar_{p2}+cr_{p2}, \cdots, ar_{pd_a}+cr_{pd_a}, mr_{p1}, mr_{p2}, \cdots, mr_{pd_m}, r_{p1}, \cdots, r_{p(k-1)}, -\sum_{i=1}^{d_a} cr_{pi})^{\mathrm{T}}\times(ar_{q1}+cr_{q1}, ar_{q2}+cr_{q2}, \cdots, ar_{qd_a}+cr_{qd_a}, mr_{q1}, mr_{q2}, \cdots, mr_{qd_m}, r_{q1}, \cdots, r_{q(k-1)}, -\sum_{i=1}^{d_a} cr_{qi})\times\boldsymbol{M}^{\mathrm{T}}\times(\boldsymbol{M}^{\mathrm{T}})^{-1}\\&=(ar_{p1}+cr_{p1}, ar_{p2}+cr_{p2}, \cdots, ar_{pd_a}+cr_{pd_a}, mr_{p1}, mr_{p2}, \cdots, mr_{pd_m}, r_{p1}, \cdots, r_{p(k-1)}, -\sum_{i=1}^{d_a} cr_{pi})^{\mathrm{T}}\times(ar_{q1}+cr_{q1}, ar_{q2}+cr_{q2}, \cdots, ar_{qd_a}+cr_{qd_a}, mr_{q1}, mr_{q2}, \cdots, mr_{qd_m}, r_{q1}, \cdots, r_{q(k-1)}, -\sum_{i=1}^{d_a} cr_{qi})\end{aligned} \tag{11}$$

式(11)的计算结果也是一个 $d\times d$ 矩阵，用 $matrix$ 表示. 接下来，User 通过公式 $\prod_{i=d_a+1}^{d_a+d_m} matrix[i][i]$ 来计算最终结果. 本方案在不解密的情况下只支持一次乘法运算.

4.2.2.5　除法

同样，我们先介绍实现除法操作的原理. 假设有两个数值数据 np 和 nq，它们对应的 6 维向量分别是 $\boldsymbol{v}_1=(x_1+c_1, y_1+c_2, a_1, b_1, r_1, -c_1-c_2)$ 和 $\boldsymbol{v}_2=(x_2+c_3, y_2+c_4, a_2, b_2, r_2, -c_3-c_4)$，其中 $np=x_1+y_1$，$np=a_1\times b_1$，$nq=x_2+y_2$ 和 $nq=a_2\times b_2$，c_1、c_2、r_1 和 r_2 是随机实数. 根据式(8)，我们得到以下结果：

$$\prod_{i=3}^{4}\left(\sum_{j=1}^{2} matrix[j][i]-matrix[5][i]\right)=(x_1+y_1)^2\times(a_2\times b_2)=np^2\times nq \tag{12}$$

$$\prod_{i=3}^{4}\left(\sum_{j=1}^{2} matrix[i][j]-matrix[i][5]\right)=(a_1\times b_1)\times(x_2+y_2)^2=np\times nq^2 \tag{13}$$

所以 np/nq 可以通过如下公式得到：

$$np/nq=\left(\frac{\prod_{i=3}^{4}\left(\sum_{j=1}^{2} matrix[j][i]-matrix[5][i]\right)}{\prod_{i=3}^{4}\left(\sum_{j=1}^{2} matrix[i][j]-matrix[i][5]\right)}\right)^{(2-1)} \tag{14}$$

根据式(12)～(14)，我们给出除法操作的方案.

SP 首先按照式(10)对 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 进行运算，然后将结果返回给 User. User 根据式(11)对结果进行解密得到矩阵 $matrix$. 为了得到最后的结果，User 还需要按照以下步骤进行处理：

1)　$$\prod_{i=d_a+1}^{d_a+d_m}\left(\sum_{j=1}^{d_a} matrix[j][i]-matrix[d-1][i]\right)=np^{d_m}\times nq \tag{15}$$

2)　$$\prod_{i=d_a+1}^{d_a+d_m}\left(\sum_{j=1}^{d_a} matrix[i][j]-matrix[i][d-1]\right)=np\times nq^{d_m} \tag{16}$$

3)　$$\sqrt[d_m-1]{\frac{np^{d_m}\times nq}{np\times nq^{d_m}}}=\sqrt[d_m-1]{\frac{np^{d_m-1}}{nq^{d_m-1}}}=np/nq \tag{17}$$

式(17)的结果就是最后的结果. 本方案在不解密的情况下只支持一次除法运算.

4.3 CESVMC 的正确性

(1) $\forall m\in \boldsymbol{M}$, 证明 $Dec(Enc(m,k))=m$.

证明. CESVMC 基于向量和矩阵运算,密钥 $\boldsymbol{M}$ 是一个 $d\times d$ 的可逆矩阵,因此,将明文数据 m 转换为对应的 d 维向量 $\boldsymbol{p}$, 则有 $Dec(Enc(m,k))=\boldsymbol{M}^{-1}\times(\boldsymbol{M}\times\boldsymbol{p})=\boldsymbol{p}$, 因此 $\boldsymbol{p}$ 是可恢复的,即 m 是可以恢复的.

所以 $\forall m\in D$, $\exists Dec(Enc(m,k))=m$. 证毕.

(2) $\forall\{m_1,m_2,\cdots,m_t\}$, 其中 $m_i\in D$, 证明 $Cal'(m_1,m_2,\cdots,m_t,op)=Dec(Cal(Enc(m_1,K),Enc(m_2,K),\cdots,Enc(m_t,K),op))$.

证明. 由矩阵和向量运算的性质易见,在进行加、减、乘和除法运算时,满足 $Cal'(m_1,m_2,\cdots,m_t,op)=Dec(Cal(Enc(m_1,K),Enc(m_2,K),\cdots,Enc(m_t,K),op))$. 因此,本证明重点分析字符串模糊检索的正确性.

根据第 4.2.1.1 节,将两个外包字符串 np_1 和 np_2 分别转换为两个 $(n-1)$ 维的向量 $\boldsymbol{p}_1=(v'_{11},v'_{12},\cdots,v'_{1(n-1)})$ 和 $\boldsymbol{p}_2=(v'_{21},v'_{22},\cdots,v'_{2(n-1)})$, 检索参数 nq 对应的 $(n-1)$ 维向量为 $\boldsymbol{q}=(v'_1,v'_2,\cdots,v'_{n-1})$, 则有

$$\begin{aligned}\boldsymbol{Q}\times\boldsymbol{P}_1-\boldsymbol{Q}\times\boldsymbol{P}_2&=r\times(\boldsymbol{p}_1\times\boldsymbol{q}-0.5\times\|\boldsymbol{p}_1\|^2-\boldsymbol{p}_2\times\boldsymbol{q}+0.5\times\|p_2\|^2)\\&=r\times(\boldsymbol{p}_1\times\boldsymbol{q}-0.5\times\|\boldsymbol{p}_1\|^2-\boldsymbol{p}_2\times\boldsymbol{q}+0.5\times\|\boldsymbol{p}_2\|^2-0.5\times\|\boldsymbol{q}\|^2+0.5\times\|\boldsymbol{q}\|^2)\\&=-0.5\times r\times((\|\boldsymbol{p}_1\|^2-2\boldsymbol{p}_1\boldsymbol{q}+\|\boldsymbol{q}\|^2)-(\|\boldsymbol{p}_2\|^2-2\boldsymbol{p}_2\boldsymbol{q}+\|\boldsymbol{q}\|^2))\\&=-0.5\times r\times(\sum_{i=1}^{n-1}(v'_{1i}-v'_i)^2)-\sum_{i=1}^{n-1}(v'_{1i}-v'_i)^2))\\&=-0.5\times r\times\sum_{i=1}^{n-1}((v'_{1i}-v'_i)^2-(v'_{2i}-v'_i)^2))\end{aligned}\tag{18}$$

对于每个字符串的第 i 段子串,由于向量元素 v'_i 是由该段的组成字符的 ASCII 码转换而来,因此越相近的字符之间的差值越小,从而保证了该段上的 v'_i 值之差的大小反映了第 i 段子串的相似程度.

对于不同段,根据前缀匹配原则,越前面的段对整个差值的影响也越大,因此,前后两段的关系为 $v'_{i+1}=v'_i\times10^{12}$(每段由 6 个字符组成,每个字符用 2 位数的 ASCII 值表示),从而实现了前缀匹配. 证毕.

5 CESVMC 的安全性

定理 1. CESVMC 是距离不可恢复的(distance-recoverable).

证明. 如果 CESVMC 是距离可恢复的,那么必然存在一个函数 f,使得对于 $\forall m_1,m_2\in D$, $f(Enc(m_1,M),Enc(m_2,\boldsymbol{M}))=d(m_1,m_2)$. 选择两个不同的密钥 $\boldsymbol{M}_1,\boldsymbol{M}_2$ 和 $x_1,x_2\in D$,CESVMC 满足

(1) $a_1=Enc(m_1,\boldsymbol{M}_1)=Enc(x_1,\boldsymbol{M}_2)$;

(2) $a_2=Enc(m_2,\boldsymbol{M}_1)=Enc(x_2,\boldsymbol{M}_2)$;

(3) $d(m_1,m_2)\neq d(x_1,x_2)$.

则由于 $f(a_1,a_2)=f(Enc(m_1,M_1),Enc(m_2,M_1))=d(m_1,m_2)$, $f(a_1,a_2)=f(Enc(x_1,M_2),Enc(x_2,M_2))=d(x_1,x_2)$;

所以 $d(m_1,m_2)=d(x_1,x_2)$.

以上结论与条件(3)冲突,因此函数 f 是不存在的. 所以 CESVMC 是距离不可恢复的. 证毕.

定义 5(方程不可解). 假设 $\boldsymbol{P}$ 为 d 维向量集合,$\boldsymbol{M}$ 为 $d\times d$ 矩阵集合,方程 $f(P_i,M_j)=P_k$ 是不可解的,如果已知 $\forall P_i\in\boldsymbol{P}$ 和 $\forall M_j\in\boldsymbol{M}$, 在方程 $f(P_i,M_j)=P_k(P_k\in\boldsymbol{P},\ i,j,k\in N)$ 中,方程等号两边的未知元素个数分别为 s_1 和 s_2,且保持 $s_1>s_2$.

定理 2. 从字符串加密与检索的角度,(1) 当适应性选择密文攻击者只获得外包数据的加密和解密 Oracle 的情况下,CESVMC 是 IND-CCA 安全的;(2) 当适应性选择密文攻击者获得了检索参数的加密 Oracle 后,CESVMC 是不安全的.

证明. (1) 当一个适应性选择密文攻击者 A 只获得外包数据的加密和解密 Oracle 的情况下,CESVMC 是 IND-CCA 安全的.

1) A 可以通过 CESVMC 的外包数据加密和解密 Oracle 获得 $t(t\in N)$ 对明/密文对,假设 CESVMC 使用 $d(d\in N,d=n+k,n$ 表示计算元素个数,k 表示随机元素个数)维向量和 $d\times d$ 矩阵,根据 4.2.1 节可知,由于分裂串 S 的存在,当 A 在不知道 S 的具体值的时候,使得分裂后的两个方程组的元素对于 A 来说都变成了未知数,因此方程等号两边的未知数个数的关系为 $d\times d+t\times2\times d>t\times2\times d\Rightarrow d\times d>0$. 所以,CESVMC 的矩阵方程组是不可解的;当 A 对分裂串进行穷举法攻击时,其破解函数的时间复杂度为 $O(2^d)$. 根据文献[18],当系统的复杂度按指数形式增长,那么系统为实际有效安全的.

2) A 发起以下攻击:(1) A 选择一些明/密文向 CESVMC 进行加/解密询问,E 将加/解密的结果返回给 A;(2) A 选择两个数据 m_0,m_1 发给 CESVMC,CESVMC 随机加密其中一条消息 $m_b(b\in\{0,1\})$,并将产生的密文 c 返回给 A;(3) A 继续选择一些明/密文向 CESVMC 进行加/解密询问,唯一的限制就是不能要求解密 c,CESVMC 将加/解密的结果返回给 A, 这时 A 一共得到 t 组明/密文对

$\{(m_1,P_1),(m_2,P_2),\cdots,(m_t,P_t)\}$；(4) A 输出 b' 作为对 b 的猜测.

根据 4.2.1.2 和 4.2.1.3 节可知，对于外包字符串 np 和检索参数 nq，它们对应的密文分别是 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$(未分裂)：

$$\boldsymbol{P}=\boldsymbol{M}\times(\boldsymbol{p}_i,-0.5\times\|\boldsymbol{p}_i\|^2,r_1,w_2,\cdots,r_{k-3},w_{k-2},-(\sum_{j=1}^{k/2-1}r_{2\times j-1}\times w_{2\times j-1}),1)^{\mathrm{T}} \tag{19}$$

$$\boldsymbol{Q}=r\times(\boldsymbol{q},1,w_1,r_2,\cdots,w_{k-3},r_{k-2},1,-(\sum_{j=1}^{k/2-1}r_{2\times j}\times w_{2\times j}))\times\boldsymbol{M}^{-1} \tag{20}$$

其中 r 和 r_i 都是随机实数，它们在每个字符串中的取值都是随机的. 根据矩阵和向量运算的规则，如式(21)所示，

$$\boldsymbol{M}\times(\boldsymbol{v})^{\mathrm{T}}=\begin{pmatrix}a_{11}&a_{12}\\a_{21}&a_{22}\end{pmatrix}\times\begin{pmatrix}x\\r\end{pmatrix}^{\mathrm{T}}=\begin{pmatrix}a_{11}\times x+a_{12}\times r\\a_{21}\times x+a_{22}\times r\end{pmatrix} \tag{21}$$

结果向量的每个元素是由作为操作数的矩阵和向量的多个元素共同决定的，并且都受随机数 r 的影响，这就保证了同一字符串在相同的密钥下加密多次会产生不同的值. 再根据定理 1 可知，CESVMC 是距离不可恢复的. 由于 CESVMC 的值域为实数域 R，设 $|R|$ 表示 R 中的元素个数，所以 A 的优势概率为

$$Adv(A)=\left|Pr[b'=b]-\frac{1}{2}\right|=\left|\left(\frac{1}{2}+\frac{1}{|R|}\right)-\frac{1}{2}\right|=\frac{1}{|R|}.$$

因此，A 的优势概率是可忽略的，CESVMC 是 IND-CCA 安全的.

(2) 当适应性选择密文攻击者获得了检索参数的加密 Oracle 后，CESVMC 是不安全的.

假设数据库中有 t 个数据 $\{np_1,np_2,\cdots,np_t\}$ 对应的密文 $\{\boldsymbol{P}_1,\boldsymbol{P}_2,\cdots,\boldsymbol{P}_t\}$，攻击者 A 可以通过以下步骤展开攻击：

① A 首先选取一个极小的检索参数 nq_0，例如由一个空格组成的字符串，通过检索参数加密 Oracle 后得到对应的密文 Q_0. 根据 4.2.1.4 节可知，A 可做如下运算：

$$\boldsymbol{Q}_0\times\boldsymbol{P}_i-\boldsymbol{Q}_0\times\boldsymbol{P}_j=d^2(\boldsymbol{p}_j,\boldsymbol{q}_0)-d^2(\boldsymbol{p}_i,\boldsymbol{q}_0)>0\Rightarrow|np_i-nq_0|<|np_j-nq_0|$$

从而对 $\{np_1,np_2,\cdots,np_t\}$ 进行排序，相似度由大到小排序的结果为 $\{np'_1,np'_2,\cdots,np'_t\}$. 由于 nq_0 足够小，所以 $\{np'_1,np'_2,\cdots,np'_t\}$ 反映了数据 $\{np_1,np_2,\cdots,np_t\}$ 的大小关系；

② A 重复步① k 次，选择的 k 个检索参数 $\{nq_1,nq_2,\cdots,nq_k\}$ 满足 $nq_i<nq_j(i\in[0,k-1],j\in[1,k],i<j)$，可以猜测出密文 $\boldsymbol{P}$ 对应的明文值，原理如下：

因为 $dis=\boldsymbol{Q}\times\boldsymbol{P}=\boldsymbol{p}\times\boldsymbol{q}-0.5\times\|p\|^2$，$dis$ 的值随着 nq 逐渐靠近 np 逐渐变小；当 $nq=np$ 时，dis 最小；当 nq 逐渐远离 np 变大时，dis 的值又逐渐变大；

又因为 k 个检索参数 $\{nq_1,nq_2,\cdots,nq_k\}$ 满足 $nq_i<nq_j$ 所以 A 可以通过逐渐逼近的方法获得密文 np 对应的明文.

故当 A 获得检索参数加密 Oracle 情况下，CESVMC 是不安全的.　　证毕.

定理 3.　从数值数据的加解密和算术运算的角度，当加法因子的个数 $d_a>3$ 时，CESVMC 是 IND-CCA 安全的.

证明.　(1) 一个适应性选择密文攻击者 A 使用 CESVMC 的外包数据加密和解密 Oracle 获得了 $t(t\in N)$ 对明密文对，假设 CESVMC 使用 $d(d\in N,d=n+k,n$ 表示计算元素个数，k 表示随机元素个数)维向量和 $d\times d$ 矩阵，根据 4.2.2 节可知，方程等号两边的未知数个数的关系为 $d\times d+t\times(d_a+(n-2)+k-1)\times d>t\times d\Rightarrow(d+d_a-3)>d\Rightarrow d_a>3$. 所以，当 $d_a>3$，CESVMC 的矩阵方程组是不可解的.

(2) A 发起以下攻击：① A 选择一些明/密文向 CESVMC 进行加/解密询问，CESVMC 将加/解密的结果返回给 A；② A 选择两个数据 m_0,m_1 发给 CESVMC，CESVMC 随机加密其中一条消息 $m_b(b\in\{0,1\})$，并将产生的密文 c 返回给 A；③ A 继续选择一些明/密文向 CESVMC 进行加/解密询问，唯一的限制就是不能要求解密 c，CESVMC 将加/解密的结果返回给 A，这时 A 一共得到 t 组明/密文对 $\{(m_1,\boldsymbol{P}_1),(m_2,\boldsymbol{P}_2),\cdots,(m_t,\boldsymbol{P}_t)\}$；④ A 输出 b' 作为对 b 的猜测.

根据 4.2.2.1 节，对于数值数据 np，其对应的密文为

$$\boldsymbol{P}=\boldsymbol{M}\times(ar_1+cr_1,ar_2+cr_2,\cdots,ar_{d_a}+cr_{d_a},mr_1,mr_2,\cdots,mr_{d_m},r_1,\cdots,r_{k-1},-\sum_{i=1}^{d_a}cr_i)^{\mathrm{T}} \tag{22}$$

其中除 ar_{d_a}、mr_{d_m} 和 $-\sum_{i=1}^{d_a}cr_i$ 外都是随机实数，它们在每个数值数据中的取值都是随机的. 根据矩阵和向量运算的规则，如式(21)所示，结果向量的每个元素是由作为操作数的矩阵和向量的多个元素共同决定的，并且都受随机数的影响，这就保证了同一数值在相同的密钥下加密多次会产生不同的值. 由于 CESVMC 的值域为实数域 R，设 $|R|$ 表示 R 中的元

素个数，所以 A 的优势概率为

$$Adv(A)=\left|Pr[b'=b]-\frac{1}{2}\right|$$
$$=\left|\left(\frac{1}{2}+\frac{1}{|R|}\right)-\frac{1}{2}\right|=\frac{1}{|R|}.$$

因此，当 $d_a>3$ 时，A 的优势概率是可忽略的.

(3)在进行算术运算时，根据 4.2.2.2 节至 4.2.2.5 节易见，运算过程并没有破坏密文数据的不确定性，在没有密钥的情况下，A 的优势概率如(2).

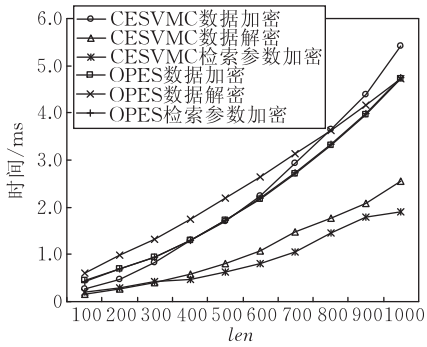
综合(1)，(2)和(3)可知，当加法因子的个数 $d_a>3$时，CESVMC 是 IND-CCA 安全的. 证毕.

6 CESVMC 的性能

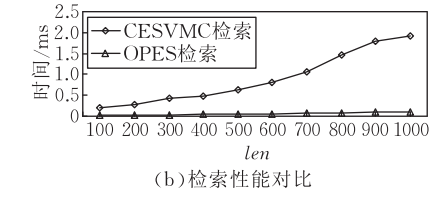
本节将通过对 CESVMC 和已有的方案进行比较，从而评估 CESVMC 的性能. 对比方案包括基于 OPES[12] 的加密字符串模糊检索方案，支持乘法同态的 unpadded_RSA 和支持加法同态的 Paillier. 性能指标包括加解密性能、计算(检索和算术运算)性能以及存储/通信负载等. 实验部署在由本研究小组自行研发的青云实验平台 3.0 上. 该平台是一个基于 Web 的面向校园的云计算环境，以 KVM 和 Hadoop 的 HDFS 为底层支撑技术，并部署在由 10 台服务器构成的集群上.

6.1 字符串

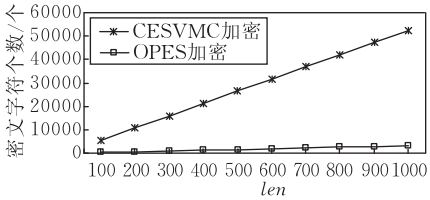
实验 1 对不同长度(len)的字符串进行了加解密、检索操作，其中 CESVMC 的向量维度 $d=10$，OPES 的输入分布桶数和输出分布桶数分别为 $b_1=6$ 和 $b_2=4$，结果如图 2 所示.



(a) 加、解密性能对比



(b)检索性能对比



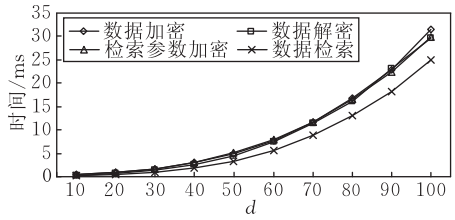
(c) 存储/通信负载对比

图 2 CESVMC($d=10$)与 OPES($b_1=6$，$b_2=4$)的性能对比

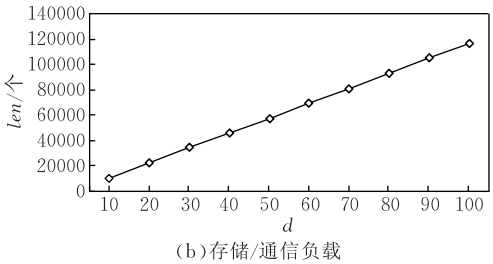
根据图 2(a)，当 $len\leq500$ 时，CESVMC 的数据加密时间比 OPES 少，当 $len>500$ 后，CESVMC 的加密时间比 OPES 多，且随着 len 增加，二者的差距缓慢变大；CESVMC 的解密时间和检索参数加密时间始终比 OPES 小，且随着 len 的增加，差距显著增加. 根据图 2(b)和 2(c)，CESVMC 的检索时间和存储/通信负载明显比 OPES 大. 本节所说的检索时间都是指进行一次比较操作所花的时间，而真正的检索时间应与数据库中的记录数有关.

实验 2 评估了 CESVMC 在向量维度 d 取不同值的情况下对长度 $len=200$ 的字符串加解密和检索的性能，结果如图 3 所示.

根据图 3(a)和 3(b)，CESVMC 的加解密时间、检索时间和存储/通信负载随着 d 的增加而增加，其中加解密时间受 d 的影响最大. 其原因是，加解密的计算复杂度为 $O(d^2)$，检索的复杂度为 $O(d)$，存储/通信复杂度为 $O(d)$.



(a) CESVMC的加解密及检索性能



(b)存储/通信负载

图 3 维数不同的情况下 CESVMC 的性能

6.2 数值数据

实验 3 比较了 CESVMC 与 unpadded_RSA、Paillier 在加解密、计算以及存储/通信负载三方面

的性能，其中，unpadded_RSA 和 Paillier 均采用两个 7 位素数作为密钥，结果如表 1 所示. 其中“*SP*”表示服务提供者，“加法_*SP*”表示由服务提供者承担的加法操作的计算时间；“*U*”表示用户，“加法_*U*”表示用户对服务提供者返回的加法运算结果进行解密的计算时间，其它表示与此类似.

表 1　CESVMC 对数值数据加解密和计算性能

名称		指标/ms											密文长度（个字符数）
		数据加密时间	数据解密	计算参数加密	加法_*SP*	减法_*SP*	加法_*U*	减法_*U*	乘法_*SP*	除法_*SP*	乘法_*U*	除法_*U*	
unpadded_RSA	两个 7 位素数作为密钥	0.025	0.031	0.026	—	—	—	—	0.001	0.001	0.019	0.019	9
Paillier	两个 7 位素数作为密钥	0.066	0.047	0.065	0.003	0.003	0.051	0.051	—	—	—	—	19
CESVMC	向量维数*d* 10	0.051	0.064	0.052	0.001	0.001	0.017	0.020	0.003	0.003	0.053	0.045	42
	20	0.249	0.255	0.249	0.001	0.001	0.014	0.017	0.008	0.007	0.171	0.169	85
	30	0.741	0.795	0.740	0.001	0.001	0.018	0.020	0.014	0.014	0.495	0.493	128
	50	3.277	3.171	3.166	0.001	0.001	0.029	0.030	0.036	0.035	2.098	2.097	214
	80	12.929	12.736	12.790	0.002	0.001	0.062	0.061	0.128	0.110	8.680	8.680	343
	100	24.745	24.690	24.664	0.002	0.002	0.092	0.099	0.206	0.264	17.341	17.218	429

根据表 1，(1) 当 $d \leq 10$ 时，CESVMC 的加解密时间小于 Paillier，大于 unpadded_RSA；CESVMC 的存储/通信负载始终大于 Paillier 和 unpadded_RSA；加解密的计算复杂度为 $O(d^2)$，存储/通信复杂度为 $O(d)$. (2) CESVMC 的加减法运算的时间几乎一样，当 $d \leq 100$ 时，CESVMC 的加减法运算时间远小于 Paillier，但是由于 CESVMC 在进行加减法运算时的计算复杂度为 $O(d)$，所以，当 d 达到一定值的时候会超过 Paillier；(3) CESVMC 的加减法运算结果的解密时间几乎一样，当 $d < 80$ 时，其值小于 Paillier；加减法运算结果的解密复杂度为 $O(d^2)$. (4) CESVMC 的乘除法运算的时间几乎一样，且远大于 *unpadded_RSA*，其计算复杂度为 $O(d^2)$. (5) CESVMC 的乘除法运算结果的解密时间也几乎是一样的，且均大于 unpadded_RSA，其计算复杂度为 $O(d^3)$.

总的说来，CESVMC 具有以下特点：(1) 加解密性能很好；(2) 加减法效率很高，但字符串模糊检索和乘除法运算的时间相对较长；(3) 加减法运算结果的解密时间适中，但乘除法运算结果的解密时间较长；(4) 存储/通信负载较大；(5) 各性能指标的值随着维度的增加而增加.

7　结束语

针对云计算中的隐私保护问题，本文提出了支持隐私保护的云计算模型，并设计了一种支持隐私保护的可计算加密方案 CESVMC. CESVMC 支持加密字符串的模糊检索和加密数值数据的加、减、乘、除四种算术运算. 通过安全分析证明：对于字符串，当攻击者只获得外包数据的加密和解密 Oracle 的情况下，CESVMC 是 IND-CCA 安全的；对于数值数据，当加法因子的个数大于 3 时，CESVMC 是 IND-CCA 安全的. 同时，性能评估证实 CESVMC 具有很好的加解密性能，能高效地实现加减法运算，但字符串模糊检索和乘除法运算的计算时间稍长，存储/通信负载较大，各性能指标的值随着维度的增加而增加.

下一步，我们将改进方案中乘除法运算的性能，使之支持多次乘/除法运算，同时减少存储/通信负载，并进一步研究数值数据排序问题. 我们也计划研究新的加密技术来实现安全的密文计算.

参　考　文　献

[1] Huang R W, Gui X L, Yu S, Zhuang W. Study of privacy-preserving framework for cloud storage. Computer Science and Information Systems, 2011, 8(3): 801-819

[2] Liu Q, Wang G J, Wu J. An efficient privacy preserving keyword search scheme in cloud computing//Proceedings of the 12th IEEE International Conference on Computational Science and Engineering (CSE'09). Vancouver, Canada, 2009: 715-720

[3] Bonech D, Crescenzo G D, Ostrovsky R, Persiano G. Public-key encryption with keyword search//Proceedings of the Eurocrypt 2004. Interlaken, Switzerland, 2004: 506-522

[4] Song D X, Wagner P, Perrig P. Practical techniques for searches on encrypted data//Proceedings of the 2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, 2000: 44-55

[5] Wang W C, Li Z W, Owens R, Bhargava B. Secure and efficient access to outsourced data//Proceedings of the 2009 ACM Workshop on Cloud Computing Security. Chicago, Illinois, USA, 2009: 55-66

[6] Bellovin S M, Cheswick W R. Privacy-enhanced searches using encrypted bloom filters. Technical Report 2004/022, IACR ePrint Cryptography Archive, 2004

[7] Ohtaki Y. Partial disclosure of searchable encrypted data with support for boolean queries//Proceedings of the 3th International Conference on Availability, Reliability and Security(ARES'2008). Barcelona, Spain, 2008: 1083-1090

[8] Li J, Wang Q, Wang C et al. Fuzzy keyword search over encrypted data in cloud computing//Proceedings of the 29th Conference on Computer Communications (INFOCOM 2010). San Diego, California, USA, 2010: 1-5

[9] Wang C, Cao N, Li J, Ren K, Lou W J. Secure ranked keyword search over encrypted cloud data//Proceedings of the 30th International Conference on Distributed Computing Systems(ICDCS'2010). Genoa, Italy, 2010: 253-262

[10] Boldyreva A, Chenette N, Lee Y, O'Neill A. Order-preserving symmetric encryption//Proceedings of the 28th Annual International Conference on Advances in Cryptology (Eurocrypt 2009). Cologne, Germany, 2009: 224-241

[11] Hacıgümüş H, Iyer B, Mehrotra S. Efficient execution of aggregation queries over encrypted relational databases//Proceedings of the 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004). Jeju Island, Korea, 2004: 633-650

[12] Agrawal R, Kiernan J, Srikant R, Xu Y. Order-preserving encryption for numeric data//Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data (SIGMOD'04). Paris, France, 2004: 563-574

[13] Wong W K, Cheung D W, Kao B, Mamoulis N. Secure *k*NN computation on encrypted databases//Proceedings of the 35th SIGMOD International Conference on Management of Data (SIGMOD 2009). Rhode Island, USA, 2009: 139-152

[14] Gentry C. Fully homomorphic encryption using ideal lattices//Proceedings of the 41st ACM Symposium on Theory of Computing(STOC'09). Bethesda, Maryland, USA, 2009: 169-178

[15] Gentry C. Computing arbitrary functions of encrypted data. Communications of the ACM, 2010, 53(3): 97-105

[16] Smart N P, Vercauteren F. Fully homomorphic encryption with relatively small key and ciphertext sizes//Proceedings of the 13th International Conference on Practice and Theory in Public Key Cryptography(PKC 2010). Paris, France, 2010: 420-443

[17] Dijk M V, Gentry C, Halevi S, Vaikuntanathan V. Fully homomorphic encryption over the integers//Proceedings of the 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology(EUROCRYPT 2010). French Riviera, 2010: 24-43

[18] Wolfram S. Origins of randomness in physical system. Physical Review Letters, 1985, 55(5): 449-452

HUANG Ru-Wei, born in 1978, Ph. D. candidate, lecturer. Her research interests include service computing, network security and homomorphic encryption technology.

GUI Xiao-Lin, born in 1966, professor, Ph. D. supervisor. His research interests include service computing, network security, and wireless network.

YU Si, born in 1988, Ph. D. candidate. His research interests include cloud computing and virtualization security.

ZHUANG Wei, born in 1989, Master candidate. His research interests include cloud computing and virtualization security.

Background

Cloud computing provides on-demand, scalable and QoS guaranteed storage and computation resources which are delivered as services, and users can visit those services anytime and anywhere. Facing the powerful and appealing advantages of cloud computing, however, a lot of people and companies are hesitant to put their data or conduct computations in cloud. The main reason is that people and companies are afraid of loss of control on their data and computations. In order to protect people's privacy, encryption is a commonly used method. Unfortunately, encryption makes effective data utilization a very challenging task, namely, it doesn't support computations on encrypted data. There have been some works on the subject, but they have some disadvantages: (i) there is no any encryption scheme which can support string retrieval and arithmetic calculations at the same time; (ii) there isn't a practical solution to realize fuzzy string retrieval on encrypted data currently; (iii) basic arithmetic calculations on encrypted numeric data, for example, addition, subtraction, multiplication and division, have always be a challenge in cryptology; (iv) some existing schemes demands the owner to do too much preparation jobs before the data is outsourced, which would render customer usage experiences very frustrated.

Therefore, how to enable an encryption scheme with support of secure computations over encrypted cloud data is a key problem in cloud computing. In this paper, we propose a computable encryption scheme based on vector and matrix calculations (CESVMC) which supports fuzzy retrieval on encrypted strings and basic arithmetic calculations on encrypted numeric data. Security analysis and performance evaluation show that CESVMC is IND-CCA, which can protect the privacies of owner and user well; and at the same time, it can realize fuzzy string retrieval and numeric calculations correctly and effectively, but the encryption and post-processing overheads in numeric calculations are a little large.

This Research was supported by the National Natural Science Foundation of China under Grant No. 60873071 and No. 91018011, the National High Technology Research and Development Program (863 Program) of China under Grant No. 2008AA01Z410, and IBM' Shared University Research Plan under Grant No. SUR201001X.