

安全可验证的行列式云外包计算及其电子交易方案

孙瑞^{1,2}, 田有亮^{2,3}

- (1. 贵州大学数学与统计学院, 贵州 贵阳 550025;
2. 贵州省公共大数据重点实验室, 贵州 贵阳 550025;
3. 贵州大学计算机科学与技术学院, 贵州 贵阳 550025)

摘要: 针对现有的云外包计算协议中服务端可能存在的用户信息被泄露、篡改等问题, 提出了一个云环境下的安全、高效、可验证的矩阵行列式外包计算协议。首先, 基于矩阵模糊技术构造云外包计算协议, 它能够在不需要任何困难性假设的前提下保证用户信息的安全性; 其次, 通过构造一类特殊的变换矩阵对明文矩阵进行处理, 使用户在收到返还结果后, 能有效验证所反馈的计算结果是否被篡改, 性能分析表明, 此协议可以有效提高云外包计算的效率; 最后, 给出一个行列式外包计算的电子交易框架, 能够有效应用于电子商务等领域。

关键词: 云外包计算; 行列式计算; 可验证性

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2016.00109

Secure and verifiable outsourcing of determinant computing in the cloud and its electronic trading scheme

SUN Rui^{1,2}, TIAN You-Liang^{2,3}

- (1. College of Mathematics and Statistics, Guizhou University, Guiyang 550025, China;
2. Guizhou Provincial Key Laboratory of Public Big Data, Guiyang 550025, China;
3. College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract: In view of the problem that users' informations are leaked and tampered possibly exists at service terminal in existing outsourcing cloud computing protocols, a secure efficient and verifiable outsourcing protocol about determinant computing in the cloud was proposed. Firstly, the outsourcing computing protocol was based on the fuzz matrix technology and needed not under the premise of any difficulty assumptions to ensure the security of users' informations. Secondly, by means of generating a special kind of transformation matrixes to deal with plaintext matrix, then after the users receive the returned results, the correctness of these results can be verified effectively, and the performance analysis shows this protocol can effectively improve the efficiency of outsourcing cloud computing. Finally, an electronic trading framework for determinant outsourcing computing was proposed, which could be effectively applied to e-commerce and other fields.

Key words: outsourcing cloud computing, determinant calculation, verifiability

收稿日期: 2016-09-10; 修回日期: 2016-10-23; 通信作者: 田有亮, youliangtian@163.com

基金项目: 国家自然科学基金资助项目 (No.61363068); 贵州大学博士基金资助项目 (No.2012-024); 贵州省教育厅科技拔尖人才支持基金资助项目 (No.黔教合 KY 字[2016]060)

Foundation Items: The National Natural Science Foundation of China (No.61363068), Doctoral Fund Project in Guizhou University (No.2012-024), Science and Technology Top-notch Talent Support Project in Guizhou Province Department of Education (No.Qian Education Combined KY word [2016]060)

1 引言

云计算^[1]是继互联网、物联网之后信息技术领域引发的一次颠覆性变革^[2], 然而, 近几年来云环境下屡发的用户隐私被侵犯等问题使其成为制约云计算发展的关键因素。如何在为用户正常提供云端服务的同时, 保障其隐私不被泄露或篡改已成为相关人员面临的重要研究课题。云外包计算是云计算的关键技术之一, 近年来愈加受到学术界、商业界等众多领域的广泛关注。云外包计算, 即云环境下的委托计算, 是用户在需要完成较为复杂的计算任务, 又受限于自身设备的计算能力时, 考虑将计算任务外包给具有海量计算能力的云端, 后由云端将计算结果反馈给客户端的过程。它能够集中网络平台中闲置的各种计算资源构成设备平台, 向用户提供具有超大规模计算能力的服务, 既拓展了弱计算能力设备的功能, 又为用户节省了购买计算设备的昂贵成本^[3]。

云外包在给人们带来经济、高效和灵活的服务的同时, 也存在致命的安全隐患, 如云服务端会因管理人员失职或黑客入侵造成用户信息被泄露、篡改等问题。因此, 云计算环境中用户信息的安全性和计算结果的可验证性是云外包技术发展、应用过程中面临的关键性问题。1996 年, Atallah 等^[4]提出了一个关于矩阵的乘法、求逆以及求线性方程组等线性代数的安全外包方案, 随后, 他们利用伪装技术提出了关于数值计算的安全外包方案^[5], 但当时都未给出验证反馈结果是否正确的方法。2010 年, Atallah 等^[6]基于文献[7]中的秘密共享方案对文献[4]中矩阵的乘法运算进行改进, 所构建的外包方案实现了可验证性。Wang 等^[8]提出了一个求解大型线性方程组的外包计算方案, 能够安全、可验证地解决线性方程组的求解问题。Lei 等^[9]提出了一个云环境下的大矩阵求逆外包计算方案, 通过置换技术和蒙特卡罗^[10]技术相结合, 将客户端计算的时间复杂度从 $O(n^{2.373})$ 降低为 $O(n^2)$, 满足了高效性, 并且也是可验证的。还有一部分学者站在同态、全同态加密的角度, 研究线性代数的委托计算问题。Gentry^[11]基于理想格提出了一个全同态加密方案, 但其运用到实际外包计算中的困难较大。2013 年, 靳方元^[12]分别基于最大近似公约数困难

性假设和求公约矩阵困难性假设, 构造了针对矩阵运算的同态加密方案 HME 与全同态加密方案 FHME, 前者只能满足有限次的矩阵乘法运算, 而后者虽然密文长度有所减少、实用性增强, 但也只适用于部分矩阵运算的范围。可见对于外包计算方案的发展, 其主要趋势是从满足用户数据的安全性, 到实现外包方案的高效与可验证性, 再上升到减少密文长度与增强实用性的方向逐步完善。

对于矩阵行列式外包计算的研究, Mohassel^[13]首先提出了一个关于行列式计算的同态加密外包思想, 并且它是建立在密码学假设的基础之上。2013 年, 胡杏等^[14]针对包含矩阵行列式在内的多个运算, 通过矩阵拆分的方法, 提出了一个可验证的外包计算协议, 并保证了用户信息的安全性。郭国欣^[15]基于双服务器模型, 采用行与列拆分和矩阵模糊技术相结合的方式构造了一个安全的行列式外包计算方案, 实现了抗双服务器合谋和可验证性的目标。

从相关的研究中可以看出, 对于线性代数外包计算方案的构建有 2 种思路: 一种是利用同态或全同态加密的方式, 便于解密, 但其安全性依赖于困难性假设, 并且仍然存在密文长、通信量大和解密效率低的缺陷, 有些方案也并不适用于所有的线性代数运算, 很难达到实用的标准; 另一种思路则是通过矩阵变换的方法进行盲化处理, 不需要建立在任何困难性假设的基础上, 且容易实现结果的可验证性, 却仍然存在密文长、通信量大的缺点。

在分析上述研究的优点与不足的基础上, 本文延续第 2 条思路——基于矩阵模糊技术^[15], 提出一个针对矩阵行列式的安全、可验证的外包计算协议。通过构造一类变换矩阵对明文矩阵进行矩阵模糊, 整个外包过程中, 在保证用户信息安全性的基础上, 有效降低了用户的计算量, 并实现反馈结果的可验证性目标。

2 外包计算模型

本文提出一个解决矩阵行列式的外包计算协议。在整个外包过程中, 存在客户端(委托方)和云端(计算方)两方作为参与者。其中客户端需要解决大规模矩阵行列式的复杂计算任务, 但计算所需购买的设备十分昂贵; 云端拥有闲置的

计算能力强大的设备平台为解决各类复杂的计算问题提供服务。在外包过程中,为了保障用户信息的安全性,需要对明文矩阵进行加密,由客户端的加密模块完成,其中包含密钥生成阶段和预计算阶段。本文协议在用户采取加密之前,分别先将明文矩阵中的每行元素进行处理,得到 m 个初级矩阵 $X_1, X_2, \dots, X_m$, 另外再构造一个新的矩阵 M , 为后续验证做准备。在经过置换矩阵加密后,客户端把计算密文矩阵 $Y_1, Y_2, \dots, Y_m, M'$ 行列式的任务外包给云端,随后得到云端所反馈的结果。最后客户端进入到解密和验证阶段,若结果通过验证,则说明云端诚实工作;否则拒绝此结果,协议结束。具体的外包过程如图1所示。

本协议所提出的外包计算系统所面对的安全威胁主要来源于云端,按照云端的行为特征可划分为两类威胁模型:半诚实模型^[16]和恶意模型^[17,18]。

半诚实模型:云服务端会诚实地按照协议完成计算任务,并不会篡改用户的信息,但在此过程中仍会窃听用户的数据,以此得到用户的敏感信息。

恶意模型:计算方没有诚实地执行协议,可能会出现泄露、篡改用户信息或随时终止协议等情况。

为了提高安全性,本协议主要是针对恶意模型下云服务端可能会存在的攻击而构造。因此,本协议最终需要达到的目标有以下几个。

1) 正确性:若云端诚实履行协议步骤,则其返还的结果一定为正确的。

2) 私密性:云服务端从输入的数据中不能获得初始行列式的具体内容,也不能从输出值中获取用户所求结果的信息。

3) 可验证性:当云端所反馈的结果正确时,用户一定能够验证通过;当所反馈的计算结果错误时,用户能验证出并拒绝此结果。

4) 高效性:用户使用本协议后的计算量远远小于其独立完成行列式运算时的计算量以及云服务端在外包协议中的计算量。

3 委托计算协议

首先引入克罗内克函数的定义

$$\delta_{i,j} = \begin{cases} 1, & x = y \\ 0, & x \neq y \end{cases}$$

设所求 n 阶方阵 X 对应的行列式为 $\det(X) =$

$$\begin{vmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{vmatrix}, \text{对于矩阵 } P \in R^{n \times n}, \text{用 } P(i, j)、$$

$p_{i,j}$ 或 p_{ij} 来表示矩阵 P 中第 i 行第 j 列的元素,用 $\det(P)$ 或 $|P|$ 表示矩阵 P 的行列式。

用户端想要将求矩阵 X 行列式的任务交给计算方进行计算,需要通过如下步骤和算法。

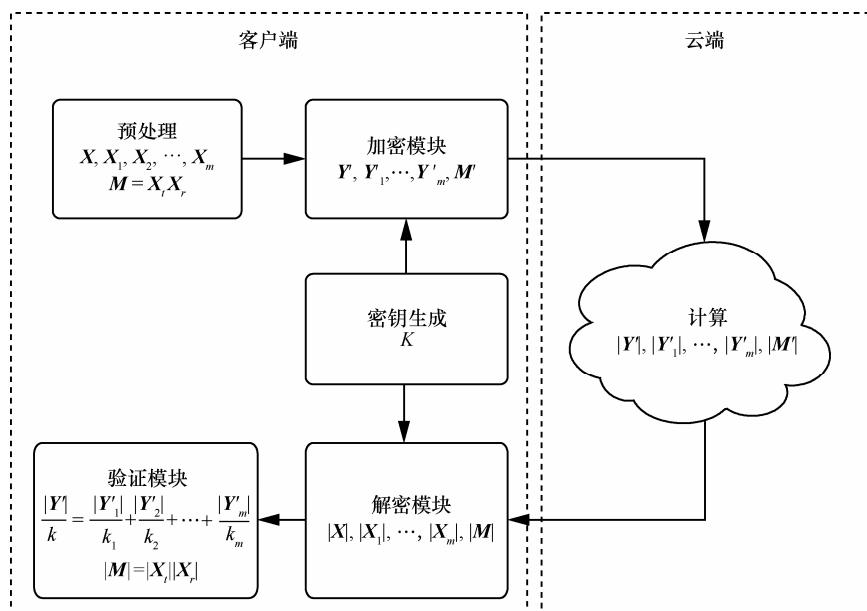


图1 协议中的外包计算模型

3.1 原始数据预处理算法 $DCpret(X)$

输入 原始矩阵 X 。

输出 随机矩阵 $X_1, X_2, \dots, X_m, M$ 。

第 1 步 随机生成 $i \in [1, n]$ 或 $j \in [1, n]$ 。

第 2 步 将矩阵 X 的第 i 行(或第 j 列) 的每个元素随机拆分成 $m(1 < m < n-2)$ 个元素之和, 将其对应的元素替换矩阵 X 的第 i 行(或第 j 列) 的相应元素, 分别得到一组矩阵 $X_1, X_2, \dots, X_m$, 使 $|X| = |X_1| + |X_2| + \dots + |X_m|$ 。

第 3 步 随机选择 $t, r \in [1, n]$ 且 $t \neq r$, 计算矩阵 $M = X_t X_r$ 。

第 4 步 输出矩阵 $X_1, X_2, \dots, X_m, M$ 。

3.2 密钥生成算法 $KeyGen$

输入安全参数 λ , 用户端调用算法 1 来生成密钥 $K: \pi, \pi_1, \dots, \pi_m, \pi_{m+1}, \{l_1, l_2, \dots, l_n\}, \{k_1^1, k_2^1, \dots, k_n^1\}, \{k_1^2, k_2^2, \dots, k_n^2\}, \dots, \{k_1^m, k_2^m, \dots, k_n^m\}, \{s_1, s_2, \dots, s_n\}, \{\alpha_1, \alpha_2, \dots, \alpha_n\}, \{\beta_1^1, \beta_2^1, \dots, \beta_n^1\}, \{\beta_1^2, \beta_2^2, \dots, \beta_n^2\}, \dots, \{\beta_1^m, \beta_2^m, \dots, \beta_n^m\}, \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ 。

3.2.1 算法 1 生成密钥

输入 一个安全参数 λ

输出 密钥 $K: \pi, \pi_1, \dots, \pi_m, \pi_{m+1}, \{l_1, l_2, \dots, l_n\}, \{k_1^1, k_2^1, \dots, k_n^1\}, \{k_1^2, k_2^2, \dots, k_n^2\}, \dots, \{k_1^m, k_2^m, \dots, k_n^m\}, \{s_1, s_2, \dots, s_n\}, \{\alpha_1, \alpha_2, \dots, \alpha_n\}, \{\beta_1^1, \beta_2^1, \dots, \beta_n^1\}, \{\beta_1^2, \beta_2^2, \dots, \beta_n^2\}, \dots, \{\beta_1^m, \beta_2^m, \dots, \beta_n^m\}, \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ 。

第 1 步 安全参数 λ 用来规定密钥空间, 客户端在 F_q 上均匀独立生成 $m+2$ 组非零随机数集合: $\{l_1, l_2, \dots, l_n\}, \{k_1^1, k_2^1, \dots, k_n^1\}, \{k_1^2, k_2^2, \dots, k_n^2\}, \dots, \{k_1^m, k_2^m, \dots, k_n^m\}, \{s_1, s_2, \dots, s_n\}$, 这里 $l_i \neq k_i^j \neq s_i$, 且 $i \in [1, n], j \in [1, m]$ 。

第 2 步 令 $l = \prod_{i=1}^n l_i, k_j = \prod_{i=1}^n k_i^j, s = \prod_{i=1}^n s_i$, 此时 $l \neq k_j \neq s$, 且不难算出 l, k_j 与 s 的值。

第 3 步 在 F_q 上均匀独立生成 $m+2$ 组 n 元非零随机数集合 $\{\alpha_1, \alpha_2, \dots, \alpha_n\}, \{\beta_1^1, \beta_2^1, \dots, \beta_n^1\}, \{\beta_1^2, \beta_2^2, \dots, \beta_n^2\}, \dots, \{\beta_1^m, \beta_2^m, \dots, \beta_n^m\}, \{\gamma_1, \gamma_2, \dots, \gamma_n\}$, 其中, $\alpha_i \neq \beta_i^j \neq \gamma_i (i \in [1, n], j \in [1, m])$ 。

第 4 步 用户在 n 元置换群 S_n 上调用算法 2 生成 $m+2$ 个不同的随机置换函数 $\pi, \pi_1, \dots, \pi_m, \pi_{m+1}$ 。

3.2.2 算法 2 生成随机置换函数 π

第 1 步 设置初始 $\pi = I_n$ (I_n 为恒等置换)。

第 2 步 for $j = 1:(n-1)$ 。

第 3 步 设置 i 在 $j \leq i \leq n$ 的随机数。

第 4 步 交换 $\pi[i]$ 和 $\pi[j]$ 。

第 5 步 End for。

3.3 加密算法 $DCend(X, K)$

输入矩阵 $X, X_1, \dots, X_m, M$ 和密钥 K , 用户端调用算法 3 进行加密得到行列式 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 。

算法 3 DC 加密

输入 初始矩阵 $X, X_1, \dots, X_m, M$ 和密钥 K :

$\pi, \pi_1, \dots, \pi_m, \pi_{m+1}, \{l_1, l_2, \dots, l_n\}, \{k_1^1, k_2^1, \dots, k_n^1\}, \{k_1^2, k_2^2, \dots, k_n^2\}, \dots, \{k_1^m, k_2^m, \dots, k_n^m\}, \{s_1, s_2, \dots, s_n\}, \{\alpha_1, \alpha_2, \dots, \alpha_n\}, \{\beta_1^1, \beta_2^1, \dots, \beta_n^1\}, \{\beta_1^2, \beta_2^2, \dots, \beta_n^2\}, \dots, \{\beta_1^m, \beta_2^m, \dots, \beta_n^m\}, \{\gamma_1, \gamma_2, \dots, \gamma_n\}$ 。

输出 $Y' = PYP^{-1}, Y'_1 = P_1 Y_1 P_1^{-1}, Y'_2 = P_2 Y_2 P_2^{-1}, \dots, Y'_m = P_m Y_m P_m^{-1}, M' = P_{m+1} M P_{m+1}^{-1}$ 。

第 1 步 客户端在 n 阶方阵上构成的矩阵环 $M_n(F_q)$ 上生成 $m+2$ 个随机的可逆变换矩阵 $\{P, P_1, \dots, P_m, P_{m+1}\}$, 其中, $P(i, j) = \alpha_i \delta_{\pi(i), j}, P_1(i, j) = \beta_1^i \delta_{\pi_1(i), j}, \dots, P_m(i, j) = \beta_m^i \delta_{\pi_m(i), j}, P_{m+1}(i, j) = \gamma_i \delta_{\pi_{m+1}(i), j}$ 。

第 2 步 客户端分别求出 $P^{-1}, P_1^{-1}, \dots, P_m^{-1}$, 其中, $P^{-1}(i, j) = \alpha_j^{-1} \delta_{\pi^{-1}(i), j}, P_1^{-1}(i, j) = \left(\frac{1}{\beta_1^i}\right) \cdot \delta_{\pi_1^{-1}(i), j}, P_2^{-1}(i, j) = \left(\frac{1}{\beta_2^i}\right) \delta_{\pi_2^{-1}(i), j}, \dots, P_m^{-1}(i, j) = \left(\frac{1}{\beta_m^i}\right) \delta_{\pi_m^{-1}(i), j}, P_{m+1}^{-1}(i, j) = \gamma_j^{-1} \delta_{\pi_{m+1}^{-1}(i), j}$ 。

第 3 步 客户端计算 $Y' = \begin{pmatrix} l_1 & 0 & \dots & 0 \\ 0 & l_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_n \end{pmatrix} X$,

$Y_1 = \begin{pmatrix} k_1^1 & 0 & \dots & 0 \\ 0 & k_2^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_n^1 \end{pmatrix} X_1, \dots, Y_m = \begin{pmatrix} k_1^m & 0 & \dots & 0 \\ 0 & k_2^m & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & k_n^m \end{pmatrix} X_m$,

$M_1 = \begin{pmatrix} s_1 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_n \end{pmatrix} M$ 。

第4步 计算矩阵 $Y = PYP^{-1}$, $Y'_1 = PY_1P^{-1}$, $\dots$, $Y'_m = P_m Y_m P_m^{-1}$, $M' = P_{m+1} M_1 P_{m+1}^{-1}$, 其中, $Y'(i, j) = \alpha_i \alpha_j^{-1} l_{\pi(i)} X(\pi(i), \pi(j))$, $Y'_1(i, j) = \beta_i^1 \left(\frac{1}{\beta_j^1} \right) l_{\pi_1(i)} X_1(\pi_1(i), \pi_1(j))$, $\dots$, $Y'_m(i, j) = \beta_i^m \left(\frac{1}{\beta_j^m} \right) l_{\pi_m(i)} X_m(\pi_m(i), \pi_m(j))$, $M'(i, j) = \gamma_i \gamma_j^{-1} s_{\pi_{m+1}(i)} M(\pi_{m+1}(i), \pi_{m+1}(j))$ 。

第5步 将加密好的 $Y', Y'_1, \dots, Y'_m$ 和 M' 发送给云服务端 W 进行计算。

3.4 云服务端 W 运算算法 $DCSolve(Y)$

分别输入加密后的矩阵 $Y', Y'_1, \dots, Y'_m$ 和 M' , 云服务端调用算法4分别计算矩阵 $Y', Y'_1, \dots, Y'_m$ 和 M' 的行列式 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 。

算法4 行列式计算

输入 $Y', Y'_1, \dots, Y'_m$ 和 M'

输出 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$

第1步 W 从客户端接收到加密后的矩阵 $Y', Y'_1, \dots, Y'_m, M'$, 调用任何存在的算法分别计算行列式值 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 。

第2步 云服务端 W 将行列式值 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 反馈给客户端。

3.5 客户端解密算法 $DCDec(|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|, K)$

用户通过公式 $|X| = \frac{|Y'|}{l}$ 进行解密, 即可求出原始矩阵 X 的行列式 $|X|$ 。

3.6 验证 $DCVerify(|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|, K)$

若反馈结果同时满足下列2个式子, 则说明云端是诚实工作的; 否则客户端将拒绝接受所有结果, 并终止协议。

$$\frac{|Y'|}{l} = \frac{|Y'_1|}{k_1} + \frac{|Y'_2|}{k_2} + \dots + \frac{|Y'_m|}{k_m} \quad (1)$$

$$\frac{|M'|}{s} = \frac{|Y'_1|}{k_i} \frac{|Y'_r|}{k_r} \quad (2)$$

4 协议分析与比较

4.1 协议分析

下面主要从正确性、安全性、可验证性3个方面对本协议进行分析。

4.1.1 正确性

若云端严格按照协议完成计算, 则其输出结果一定正确。

1) 加密过程

$$Y = \begin{pmatrix} l_1 & 0 & \dots & 0 \\ 0 & l_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & l_n \end{pmatrix} X$$

即

$$Y' = \begin{pmatrix} l_1 x_{11} & l_1 x_{12} & \dots & l_1 x_{1n} \\ l_2 x_{21} & l_2 x_{22} & \dots & l_2 x_{2n} \\ \vdots & \vdots & & \vdots \\ l_n x_{n1} & l_n x_{n2} & \dots & l_n x_{nn} \end{pmatrix}$$

又因为 $P(i, j) = \alpha_i \delta_{\pi(i), j}$, $P^{-1}(i, j) = \alpha_j^{-1} \delta_{\pi^{-1}(i), j}$, 令 $Y' = PYP^{-1}$, 则

$$Y' = \begin{pmatrix} \alpha_1 \alpha_1^{-1} l_{\pi(1)} x_{\pi(1), \pi(1)} & \dots & \alpha_1 \alpha_n^{-1} l_{\pi(1)} x_{\pi(1), \pi(n)} \\ \vdots & & \vdots \\ \alpha_i \alpha_i^{-1} l_{\pi(i)} x_{\pi(i), \pi(1)} & \dots & \alpha_i \alpha_n^{-1} l_{\pi(i)} x_{\pi(i), \pi(n)} \\ \vdots & & \vdots \\ \alpha_n \alpha_n^{-1} l_{\pi(n)} x_{\pi(n), \pi(1)} & \dots & \alpha_n \alpha_n^{-1} l_{\pi(n)} x_{\pi(n), \pi(n)} \end{pmatrix}$$

所以, $Y'(i, j) = \alpha_i \alpha_j^{-1} l_{\pi(i)} x_{\pi(i), \pi(j)}$

同理, $Y'_1(i, j) = \beta_i^1 \left(\frac{1}{\beta_j^1} \right) k_{\pi_1(i)}^1 X_1(\pi_1(i), \pi_1(j))$

$$Y'_m(i, j) = \beta_i^m \left(\frac{1}{\beta_j^m} \right) k_{\pi_m(i)}^m X_m(\pi_m(i), \pi_m(j))$$

$$M' = P_3 M_1 P_3^{-1} = \gamma_i \gamma_j^{-1} s_{\pi_{m+1}(i)} M(\pi_{m+1}(i), \pi_{m+1}(j))$$

2) 验证和解密

① 验证阶段

因为 $Y' = PYP^{-1}$, 那么, $|Y'| = |P||Y||P^{-1}| =$

$$\left(\prod_{i=1}^n l_i \right) |P| |X| |P^{-1}|$$

又由 $|P||P^{-1}| = 1$

所以 $|Y'| = \left(\prod_{i=1}^n l_i \right) |X|$

$$|X| = \frac{|Y'|}{\prod_{i=1}^n l_i} = \frac{|Y'|}{l}$$

$$\begin{aligned} \text{同理} \quad |X_1| &= \frac{|Y_1|}{\prod_{i=1}^n k_i^1} = \frac{|Y_1|}{k_1} \\ &\vdots \\ |M| &= \frac{|M'|}{s} \end{aligned}$$

②解密阶段

因为 $|X| = |X_1| + |X_2| + \dots + |X_m|$

$$\text{所以,} \quad \frac{|Y'|}{l} = \frac{|Y_1|}{k_1} + \frac{|Y_2|}{k_2} + \dots + \frac{|Y_m|}{k_m}$$

$$\text{又因为} \quad M = X_l X_r$$

$$\text{所以} \quad |M| = \frac{|M'|}{s} = |X_l| |X_r| = \frac{|Y_l|}{k_l} \frac{|Y_r|}{k_r}$$

$$\text{即} \quad \frac{|M'|}{s} = \frac{|Y_l|}{k_l} \frac{|Y_r|}{k_r}$$

因此, 若云端诚实履行协议, 其反馈结果一定正确。

4.1.2 安全性

当矩阵规模和密钥空间规模非常大时, 敌手获取输入、输出信息的概率是可以忽略的。

1) 获取输入信息的概率

在输入前, 客户端分别利用系数集合 $\{l_1, l_2, \dots, l_n\}$, $\{k_1^1, k_2^1, \dots, k_n^1\}$, $\{k_1^2, k_2^2, \dots, k_n^2\}$, $\dots$, $\{k_1^m, k_2^m, \dots, k_n^m\}$, $\{s_1, s_2, \dots, s_n\}$ 对矩阵 $X, X_1, \dots, X_m, M$ 进行处理, 再通过 $Y' = PYP^{-1}$, $Y'_1 = P_1 Y_1 P_1^{-1}$, $Y'_2 = P_2 Y_2 P_2^{-1}$, $\dots$, $Y'_m = P_m Y_m P_m^{-1}$, 对矩阵 $Y, Y_1, \dots, Y_m$ 中的每个元素及元素的位置进行深层加密; 另又构造新的矩阵 M_1 , 同理由 $M' = P_{m+1} M_1 P_{m+1}^{-1}$ 对 M_1 进行矩阵模糊。

敌手要想从 $Y', Y'_1, \dots, Y'_m, M'$ 中获得关于 X 的信息, 首先从 $Y', Y'_1, \dots, Y'_m, M'$ 中确定 Y' 的概率为 $\frac{1}{m+2}$; 其次要猜出 π 和 $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$, 其中随机置换 π 总共有 $n!$ 种可能性, 敌手猜出 $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ 的概率为 $\frac{1}{|F_q|^n}$, 此时便可得到矩阵 Y ; 最后, 从 Y 中还原出明文矩阵 X , 就要先知道 $\{l_1, l_2, \dots, l_n\}$, 其概率为 $\frac{1}{|F_q|^n}$ 。所以敌手通过 Y' 猜

$$\text{出明文矩阵 } X \text{ 的概率为 } \frac{1}{m+2} \frac{1}{n!} \left(\frac{1}{|F_q|^n} \right)^2。$$

因此, 敌手获取输入信息的概率为 $\frac{1}{m+2}$ 。

$\frac{1}{n!} \left(\frac{1}{|F_q|^n} \right)^2$, 当 n 和密钥空间规模极大时, 此概率值非常小。

2) 获取输出信息的概率

若敌手想从输出结果 $|Y'|$ 中获得 $|X|$ 的值, 首先从 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 中确定 $|Y'|$ 的概率为 $\frac{1}{m+2}$, 然后敌手还要猜出 $\{l_1, l_2, \dots, l_n\}$, 其概率为 $\frac{1}{|F_q|^n}$, 所以敌手从 $|Y'|$ 中获得 $|X|$ 的概率为 $\frac{1}{m+2} \frac{1}{|F_q|^n}$ 。

若敌手想从输出结果 $|Y'_1|, \dots, |Y'_m|$ 中获取 $|X|$ 的信息, 首先要从 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 中确定 $|Y'_1|, \dots, |Y'_m|$, 其概率为 $\frac{1}{C_{m+2}^m}$, 又因为 $|Y'_1| = |Y_1| =$

$\prod_{i=1}^n k_i^1 |X_1| = k_1 |X_1|$, 若敌手猜对 k_1 , 则可以求出 $|X_1|$, 其概率为 $\frac{1}{|F_q|^n}$, 同样敌手求出其他 $|X_i|$ 的概率相同, 当求出所有的 $|X_1|, \dots, |X_m|$ 时, 便可求其和获得 $|X|$ 。所以敌手想要从 $|Y'_1|, \dots, |Y'_m|$ 中获取 $|X|$ 的概率为 $\frac{1}{C_{m+2}^m} \frac{1}{|F_q|^{n-m}}$ 。

因此, 敌手获取输出信息的概率为 $\left(\frac{1}{m+2} + \frac{1}{C_{m+2}^m |F_q|^m} \right) \frac{1}{|F_q|^n}$, 当 n 与密钥空间的规模非常大时, 其概率是极小的。

综上分析, 当密钥空间与矩阵规模非常大时, 敌手获取输入和输出信息的概率都是可以忽略的, 因此本协议保证了用户输入信息的隐私性和输出信息的安全性。

4.1.3 可验证性

对于云服务端反馈的计算结果, 用户能够验证出反馈结果是否被篡改。

$$\frac{|Y'|}{l} = \frac{|Y'_1|}{k_1} + \frac{|Y'_2|}{k_2} + \dots + \frac{|Y'_m|}{k_m}$$

$$\frac{|M'|}{s} = \frac{|Y'_t|}{k_t} \cdot \frac{|Y'_r|}{k_r}$$

若用户要篡改输出结果, 并通过上面2个验证条件时, 首先要在 $|Y'|, |Y'_1|, \dots, |Y'_m|, |M'|$ 中确定 $|Y'|$ 和 $|M'|$, 共有 $C_{m+2}^1 C_{m+1}^1$ 种可能; 其次要猜对系数集 $\{l_1, l_2, \dots, l_n\}$ 和 $\{s_1, s_2, \dots, s_n\}$, 可能性为 $\frac{1}{|F_q|^{2n}}$, 还要从 $|Y'_1|, \dots, |Y'_m|$ 中找到 $|Y'_r|, |Y'_t|$, 共有 C_m^2 种可能; 对于 $|Y'_1|, \dots, |Y'_m|$ 相应的系数 $k_1, \dots, k_m$, 共有 $|F_q|^m$ 种可能; 猜对 $k_r k_t$ 的概率为 $\frac{1}{|F_q|^2}$ 。所以, 在篡改的情况下, 反馈结果仍然满足上面两式的概率是 $\frac{1}{(m+2)(m+1)C_m^2 |F_q|^{2n+m+2}}$, 在密钥空间和矩阵规模非常大时其概率值极小。

所以, 当云端诚实履行协议时, 用户所接收到的计算结果可以同时满足式(1)和式(2), 反馈结果一定能够被客户端验证通过并接受; 当云端具有不诚实的行为时, 便不能同时满足上面的验证条件, 从而客户端将拒绝此结果。因此, 此协议为可验证的。

4.2 性能分析与比较

本协议能够有效降低用户的计算量, 使其效率有所提高。

在整个协议中, 客户端进行的各个阶段有密钥生成、加密、解密和验证这4个部分, 其主要

的计算量在加密、解密和验证阶段, 其时间复杂度最高为 $O(n^2)$ 。假设云端计算行列式所用为常用的矩阵LC分解法^[19], 则其时间复杂度为 $O(n^3)$; 而客户端独立计算 X 原行列式的时间复杂度为 $O(n^3)$, 由此可知, 通过本协议可以大大降低用户的计算量。

现将本文提出的行列式外包计算协议与相关文献中的方法进行比较, 主要从效率、安全性、可验证性和密钥长度来分析。具体的比较结果如表1所示, 其中, n 表示行列式的阶数, $|S_n|$ 和 $|F_q|$ 分别表示变换群 S_n 和有限域 F_q 中元素的长度, m ($1 < m < n-2$)为把矩阵某行(或某列)元素分解为多个元素的个数。

由表1可知, 这4种方法具有相同的计算复杂性, 都是 $O(n^2)$ 。其中文献[20]的方法信息被获取的概率为 $\left(\frac{1}{n!}\right)\left(\frac{1}{|F_q|^n}\right)$, 且不满足可验证性。在

文献[14]、文献[15]和本文的协议中, 敌手获取输入输出信息的概率相差不大, 并且都能满足结果的可验证性, 相对于文献[20]的方法有较明显的优势。从表1中可看出, 虽然当 $m \leq 2$ 时, 本协议中密钥长度才有可能小于文献[15]方法中的密钥长度, 但在文献[15]中, 若云服务端 W 将正确的计算结果同时乘以一个常数 $k(k \neq 0, 1)$ 反馈给用户时, 其反馈结果仍可满足用户的验证条件, 但已受到了篡改。因此, 文献[15]的方法不能抵抗此种攻击, 而文献[14]和本文协议中的方法则能够抵抗此种攻击。此外由表1可以看出, 本文提出的方法在相同的安全级别与效率下, 与文献[14]

表1 本文协议与相关协议的比较

方案	计算复杂度	获取输入信息的概率	获取输出信息的概率	可验证性	密钥长度	
					$ S_n $	$ F_q $
文献[20]	$O(n^2)$	$\left(\frac{1}{n!}\right)\left(\frac{1}{ F_q ^n}\right)$	$\left(\frac{1}{n!}\right)\left(\frac{1}{ F_q ^n}\right)$	×	1	n
文献[14]	$O(n^2)$	$\left(\frac{1}{n!}\right)^2\left(\frac{1}{ F_q ^n}\right)^2$	$\left(\frac{1}{n!}\right)^2\left(\frac{1}{ F_q ^n}\right)^2$	√	$2(m+2)$	$2(m+2)n$
文献[15]	$O(n^2)$	$\frac{1}{2}\left(\frac{1}{n}\right)\left(\frac{1}{n!}\right)^2\left(\frac{1}{ F_q ^{2n+1}}\right)$	$\frac{1}{2}\left(\frac{1}{n}\right)\left(\frac{1}{n!}\right)^2\left(\frac{1}{ F_q ^{2n+1}}\right)$	√	8	$8n$
本文协议	$O(n^2)$	$\frac{1}{m+2}\frac{1}{n!}\left(\frac{1}{ F_q ^n}\right)^2$	$\left(\frac{1}{m+2} + \frac{1}{C_{m+2}^m F_q ^m}\right)\frac{1}{ F_q ^n}$	√	$m+2$	$2(m+2)n$

相比密钥长度更短，从而减小了用户的开销。

5 电子交易应用方案

在本文协议的基础上，提出一个针对用户的行列式外包计算的数据交易实例。本实例由用户、云服务端 W 和可信第三方 3 部分组成，图 2 为此外包计算交易的流程。

交易的具体过程如下。

1) 首先，用户调用协议中的预处理算法 $DCPret(X)$ 、密钥生成算法 $KeyGen$ 和加密算法 $DCEnd$ 对原始数据进行处理。接着用户在相应的交易机构进行注册，提供合法的身份 ID 并获取用来签名的密钥对 $ID = \langle pk_{ID}, sk_{ID} \rangle$ ，再选择合作的云服务端 W 以及业务，合成订单并发送给可信第三方，同时向可信第三方支付外包费用。收款后可信第三方向用户发送支付凭证，随后用户将订单和密文信息发送给云服务端 W 提出交易。

2) 云服务端 W 收到订单与密文矩阵后调用协议中的算法 $DCSolve$ 计算所求矩阵的行列式 $|Y'|, |Y_1|, \dots, |Y_m|, |M'|$ ，得到结果后将计算结果反馈给用户。

3) 用户调用协议中的解密算法 $DCDec$ 和验证算法 $DCVer$ 对计算结果进行验证，判断反馈结

果是否满足验证条件。若满足，则接受此结果，并发送反馈结果有效的证明凭证给可信第三方，可信第三方核实凭证后公布订单实现状态，并将用户支付的外包费用发放给云服务端 W ；若反馈结果不满足验证条件，则用户拒绝此结果并发送反馈结果无效证明给可信第三方，可信第三方核实后发布订单取消状态，并退款给用户，交易结束。

由于交易订单是否能够实现取决于云服务端的反馈结果能否通过用户的验证条件，所以，此交易方案可以有效避免因云服务端 W 的恶意篡改或为节约成本而随意编造计算结果等行为对用户造成的经济损失。因此，通过执行本方案，不但能够节约用户完成行列式计算的时间和成本，还保障了用户的利益不受损失。

6 结束语

本文针对矩阵行列式运算提出了一个云环境下高效的、可验证的外包计算协议。经过分析表明，本文协议在保证用户信息安全性与可验证性的基础上，能够实现高效性的目标。此外，在本文协议的基础上给出了一个行列式外包计算的电子交易框架，可以有效地应用于电子商务等领域

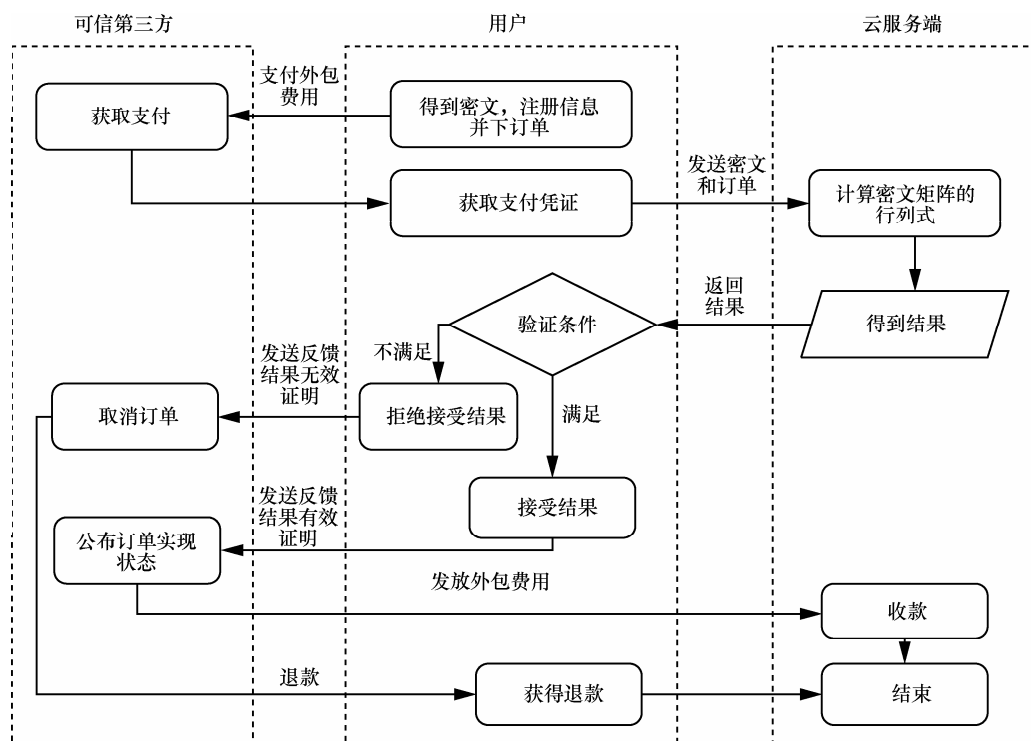


图 2 外包计算的电子交易流程

中, 保证用户在进行交易时利益不受损失。接下来的工作是进一步研究如何提高云外包计算协议的效率, 并将其运用到计算资源受限的应用环境。

参考文献:

- [1] ARMBRUST M, FOX A, GRIFFITH R, et al. Above the clouds: a berkeley view of cloud computing[R]. UCB / EECS-2009-28. Berkeley, USA: Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.
- [2] 冯登国, 张敏, 张妍, 等. 云计算安全研究[J]. 软件学报, 2011, 22(1):71-83.
- FENFG D G, ZHANG M, ZHANG Y, et al. Study of cloud computing security[J]. Journal of Software, 2011, 22(1):71-83.
- [3] 罗军舟, 金嘉晖, 宋爱波, 等. 云计算: 体系架构与关键技术[J]. 通信学报, 2011, 32(7):3-21.
- LUO J Z, JIN J H, SONG A B, et al. Cloud computing: the architecture and key technologies[J]. Journal on Communications, 2011, 32(7):3-21.
- [4] ATALLAH M J, KONSTANTINOS N, et al. Secure outsourcing of some computations[R]. Indiana: Purdue University, 1996.
- [5] ATALLAH M J, PANTAZOPOULOS K N, RICE J R, et al. Secure outsourcing of scientific computations[J]. Advances in Computers, 2002, 54(1):215-272.
- [6] ATALLAH M J, FRIKKEN K. Securely outsourcing linear algebra computations[C]//The 5th ACM Symposium on Information. 2010: 48-59.
- [7] SHAMIRA. How to share a secret[J]. Communications of the ACM, 1979, 22(11):612-613.
- [8] WANG C, REN K, WANG J. Harnessing the cloud for securely solving large-scale systems of linear equations [C]//The 31st International Conference on Distributed Computing Systems. 2011: 549-558.
- [9] LEI X Y, LIAO X F, HUANG T W, et al. Outsourcing large matrix inversion computation to a public cloud[J]. IEEE transactions on Cloud Computing, 2013, 1(1):78-87.
- [10] MOTWANIR, RAGHAVANP. Randomized algorithms[M]. Cambridge: Cambridge University Press, 1995: 195-196.
- [11] GENTRY C. Fully homomorphic encryption using ideal lattices[C]//ACM Symposium on Theory of Computing. 2009:169-178.
- [12] 靳方元. 基于矩阵同态委托计算方案的研究与设计[D]. 苏州: 苏州大学, 2013.
- JIN F Y. The research and design based on matrix homomorphism entrust calculation scheme[D]. Suzhou: Soochow University, 2013.
- [13] MOHASSEL P. Efficient and secure delegation of linear algebra. Cryptology ePrint Archive, Report 2011/605[DB/OL]. <http://eprint.iacr.org/2011/605>.
- [14] 胡杏, 裴定一, 唐春明. 可验证安全外包矩阵计算及其应用[J]. 中国科学: 信息科学, 2013, 43(7): 842-852.
- HU X, PEI D Y, TANG C M. The caculation and application of verifiable secure outsourcing matrix[J]. Scientia Sinica Informations, 2013, 43(7): 842-852.
- [15] 郭国欣. 矩阵运算可验证安全外包计算协议的研究[D]. 西安: 西安电子科技大学, 2014.
- WU G X. Study of outsourcing matrix matrix verifiable security outsourcing computation protocol[D]. Xi'an: Xidian University, 2014.
- [16] 罗文俊, 李祥. 多方安全矩阵乘法协议及应用[J]. 计算机学报, 2005, 28(7): 1230-1235.
- LUO W J, LI X. Multilateral security protocol and application of matrix multiplication[J]. Chinese Journal of Computer, 2005, 28(7): 1230-1235.
- [17] 邵志毅. 云环境下的隐私保护计算[D]. 西安: 陕西师范大学, 2015.
- SHAO Z Y. Privacy protection under the cloud environment[D]. Xi'an: Shaanxi Normal University, 2015.
- [18] YANG B, YU Y, YANG C H. A secure scalar product protocol against malicious adversaries[J]. Journal of Computer Science & Technology, 2013, 28(1):152-158.
- [19] 王开荣, 杨大地. 应用数值分析[M]. 北京: 高等教育出版社, 2010.
- WANG K R, YANG D D. Analysis of applied values[M]. Beijing: Higher Education Press, 2010.
- [20] ZHANG J, ZHU Y, JIN F. Practical and secure outsourcing of linear algebra in the cloud[C]//International Conference on Advanced Cloud and Big Data. 2013: 81-87.

作者简介:



孙瑞(1993-), 女, 新疆乌鲁木齐人, 贵州大学硕士生, 主要研究方向为密码学理论与工程。



田有亮(1982-), 男, 贵州六盘水人, 博士, 贵州大学教授、硕士生导师, 主要研究方向为算法博弈论、密码学与安全协议。