计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440453
61(10): 2448−2466, 2024

轻量级多用户可验证隐私保护基因序列分析方案

胡云舒　周　俊　曹珍富　董晓蕾
（华东师范大学软件工程学院 密码与网络安全系　上海　200062）
（hysss1136@gmail.com）

Lightweight Multi-User Verifiable Privacy-Preserving Gene Sequence Analysis Scheme

Hu Yunshu, Zhou Jun, Cao Zhenfu, and Dong Xiaolei
（*Department of Cryptography and Network Security*, *Soft Engineering Institute*, *East China Normal University*, *Shanghai* 200062）

Abstract　As the development of the emerging areas of network services such as big data and cloud computing, data element has played an increasingly critical role in the fields of intelligent e-health and scientific research. Gene sequencing technology is widely used in many fields to determine the cause and category of a patient's disease, by processing the patient's gene sequence. Due to constrained storage and computing resources, local users often need to rent resource-abundant cloud servers, unfortunately always working in untrusted environments, to fulfill the computationally-intensive task of large-scale gene sequencing function evaluation. To guarantee users' data privacy and the correctness of computing results, most of the state-of-the-art methods exploit the techniques of public key fully homomorphic encryption and secure multiparty computation to achieve data privacy, and the technique of Yao's garbled circuit or bilinear paring to achieve correctness verification. Owing to the fact that huge computational overhead and communication overhead are required in the cryptographic primitives mentioned above, they are inappropriate for efficiency needs of the resource-constrained local users in gene sequence analysis. To address this challenging issue, in this paper, a lightweight verifiable privacy-preserving gene sequence analysis scheme in the multi-user setting is proposed. Firstly, we design an efficient verifiable multi-key homomorphic data encapsulation mechanism VMK-HDEM. The proposed VMK-HDEM enables batch outsourced function evaluation on L different input instances over the encrypted domain. The usage time complexity of public key encryption on each user Sen_i's end is $O(L)$, independent of the dataset size n_i, which significantly decreases the computational cost of local users. For verification, the size of the proof is $O(deg_F)$ where deg_F denotes the degree of the function, independent to the size of user's dataset n_i. Furthermore, based on our constructed cryptographic primitive VMK-HDEM, a lightweight and efficient verifiable privacy-preserving gene sequence analysis scheme LWPPGS is proposed. It not only can preserve the privacy of both users' gene datasets and the results of gene sequence analysis, but also efficiently verify the correctness of the outcome. Finally, formal security proof and experimental simulation results show the security and practicability of our proposed VMK-HDEM and LWPPGS.

Key words　multi-key fully homomorphic data encapsulation mechanism; verifiability; gene sequence analysis; privacy-preserving; efficiency

收稿日期: 2024-06-05; **修回日期**: 2024-07-17
基金项目: 国家自然科学基金项目(62172161, 62132005, 62172162)
This work was supported by the National Natural Science Foundation of China (62172161, 62132005, 62172162).
通信作者: 周俊(jzhou@sei.ecnu.edu.cn)

摘　要　随着云计算、大数据等新兴网络服务的蓬勃发展，数据要素在智慧医疗、科学研究等领域中占据重要地位. 基因组测序技术通过对病人基因组序列进行处理来判断病人的患病原因和类别，在多个领域有广泛应用. 由于存储、计算资源受限，本地用户通常需要租用资源丰富但工作在不可信环境下的云服务器来完成复杂的大规模基因组测序处理函数计算任务. 为了保护用户数据隐私及验证计算结果的正确性，国内外现有工作通常的做法是利用公钥全同态加密或安全多方计算技术实现数据隐私保护；利用 Yao 混淆电路或双线性配对技术实现计算结果的正确性验证. 由于密码原语的计算开销和通信开销巨大，均不适用于基因序列分析系统中存储、计算资源受限的本地用户的客观性能需求. 为了解决上述挑战性问题，提出了一个轻量级多用户可验证隐私保护基因序列分析方案. 首先，构造了一种高效的可验证多密钥同态数据封装机制 VMK-HDEM，该方案支持在密文域上对 L 个不同输入实例的打包计算，用户端 Sen_i 公钥加密的使用次数复杂度为 $O(L)$，即与其数据集的大小 n_i 无关，大大降低了资源受限的本地用户的计算开销；在可验证性方面，云服务器生成的密文计算结果正确性验证证据大小复杂度为 $O(deg_F)$（其中 deg_F 代表外包计算函数的阶），与用户数据集大小 n_i 无关. 然后，基于所构造的新型密码原语 VMK-HDEM，设计了一个轻量级高效的可验证隐私保护基因序列分析方案 LWPPGS，有效保护了用户基因数据集的隐私和基因序列分析结果的隐私，并高效验证其分析结果的正确性. 最后，通过形式化的安全性证明和实验仿真结果表明了所构造方案 VMK-HDEM 和 LWPPGS 的安全性和实用性.

关键词　多密钥全同态数据封装机制；可验证；基因序列分析；隐私保护；高效性

中图法分类号　TP391

随着云计算和大数据的蓬勃发展，数据要素安全正成为隐私计算、区块链等领域的有力保障，成为车联网、电子医疗等各类新兴网络服务的重要基石. 基因组测序技术自从 2003 年人类基因组计划宣告成功起，到今天已经获得了突飞猛进的发展. 现在，人们已经可以用低廉的成本获取非常详尽的基因组数据，并将其广泛应用于卫生保健、生物医学研究、司法鉴定以及直接面向消费者的基因检测等各个领域[1]. 在基因组序列的比对和处理分析中，将未知序列同已知序列进行相似性比较是一项非常重要的研究，其中序列比对运用某种特定的数学模型或算法，找出 2 个或多个序列之间的最大匹配碱基数或残基数，其反映了序列之间的相似程度. 全基因组关联分析（genome-wide association study, GWAS）作为热门的遗传分析方法，在人类和动植物复杂性状遗传研究中已取得初步成果. 它以连锁不平衡为基础，通过识别数百个或数千个个体的定位群体中高密度的分子标记，一般是从上万个甚至上百万个单核苷酸多态性（single nucletide polymorohism, SNP）标记数据库中筛选出与复杂性状表现型变异相关联的分子标记，进而分析这些分子标记对表现型的遗传效应. 然而，由于每个个体所具有的 SNP 数量巨大且基因序列比对函数运算复杂，所以存储、计算资源受限的本地用户通常将基因测试任务外包给资源充足的云服务器完成.

然而云服务器通常工作在不可信环境中. 半可信的（semi-honest）云服务器在诚实执行协议的同时会通过与用户的交互，从协议的副本中最大程度地窃取用户数据隐私；恶意的云服务器则会通过任意行为破坏协议的正确运行，返回用户错误的计算结果. 由于个体基因序列信息和基因序列分析结果与用户隐私休戚相关，且不正确的基因序列分析结果会严重影响疾病预测、诊断及相关科研工作顺利开展. 此外，基因序列分析通常要通过将多个用户的基因序列信息与带有特定病灶特征的基因序列片段进行比对实现，因此，如何构建轻量级可验证多用户隐私保护基因序列分析方案成为亟待解决的重要研究课题.

国内外现有工作大多利用公钥全同态加密、安全多方计算或差分隐私技术实现[2-8]. 多密钥公钥全同态加密可以实现在不同密钥加密的密文数据上进行密态计算. López-Alt 等人[2] 利用公钥全同态加密算法[3] 设计了多用户安全外包计算方案. Chen 等人[4] 基于 BFV 方案和 CKKS 方案[5]，给出了对应的多密钥全同态加密方案. 然而，上述方案要求数据拥有者对其数据集中的每一个数据进行公钥全同态加密，因此，其公钥加密的使用次数复杂度 $O(n_i)$ 与各用户 Sen_i 数据集大小呈线性相关，其中 n_i 表示用户数据集的大小. 此外，该方案的密文大小随着电路中乘法运算深度的增加而迅速膨胀. 因此无法满足基因序列分析任务中存储、计算、通信资源受限用户的低功

耗性能需求.

为解决上述问题, Zhou 等人[9]和曹珍富等人[10]通过减少公钥加密的使用次数提出了高效的安全外包计算新理论与新方法, 在支持用户数据密态计算的同时将本地用户Sen_i的公钥加密使用次数复杂度由$O(n_i)$降低到$O(1)$. 然而该方案仅支持单用户(单密钥)的安全外包计算, 但在基因分析系统中, 多个用户的同一 SNP 位点的基因信息需要在多个不同密钥下加密以确保不同用户之间的基因数据隐私保护, 并且加密后的密文还需在云端进行外包计算得到密态分析结果. 此外, 该方案还无法在恶意敌手模型下对云服务器给出的计算结果进行高效的正确性验证, 难以支持恶意云服务器输出错误的基因序列分析结果. 最后, 该方案还无法支持在密文域上对多个输入实例进行外包函数批量计算, 其计算开销与通信开销仍有进一步提升空间. 另一方面, 基于秘密共享等技术实现安全外包计算协议和基于数据扰动等技术实现的差分隐私保护协议, 用于多用户隐私保护基因序列分析任务, 也会对用户的通信复杂度和计算结果的精度带来不同程度的影响[10].

针对多用户场景的隐私保护基因序列密态分析, 近年来国内外也有一些研究结果[11-14]. Chen 等人[11]基于 SNP 序列数据库, 提出了采用汉明码和哈希图来保证数据隐私性与完整性的方案; 基于半诚实的云服务器, Hasan 等人[12]提出了大大减少计算开销的高效可扩展的安全共享和计算基因组数据的方法. 然而, 文献 [11–14] 工作大多利用计算、通信开销巨大的公钥全同态加密技术实现, 使得资源受限的本地用户无法承受巨大的计算和通信的复杂性, 不适用于基因序列分析系统这类大数据与复杂函数的计算场景. 且上述基因序列统计分析方案也未考虑在恶意敌手模型下为用户提供计算结果正确性的高效可验证机制.

为了解决上述挑战性问题, 本文提出了一个高效的可验证多密钥同态数据封装机制 VMK-HDEM, 该方案支持多用户、多数据的协同打包与密态计算, 且能在恶意敌手模型下保证外包计算结果的正确性. 并在此基础上, 设计了一个轻量级多用户可验证隐私保护基因序列分析方案. 本文的主要贡献有 3 点:

1)设计了高效的可验证多密钥同态数据封装机制 VMK-HDEM. 相较于国内外大多数可验证外包计算协议, 该方案不使用公钥全同态加密, 而是依据混合加密的原则, 利用离线状态下常数次(与用户数据集大小无关)任意 CPA 安全的公钥加密算法加密对称加密密钥(会话密钥), 在线状态下设计仅包含简单加法、乘法运算的且具有被动攻击下不可区分安全性的对称全同态加密算法加密数据. 在合作不合谋的双服务器模型下, 利用密文重随机化技术和重加密技术, 把多个不同密钥加密的输入数据转换成统一密钥加密的数据, 从而将多密钥问题转换为单密钥问题. 此外, 该方案还利用多密钥同态消息验证码实现恶意敌手模型下, 计算结果的接收方能高效验证云服务器密文计算结果的正确性. 云服务器可在密文域上对多个外包函数输入实例进行打包计算和批量生成计算结果正确性验证证据.

2)基于本文设计的可验证多密钥同态数据封装机制 VMK-HDEM, 提出了多用户、多数据协同的轻量级可验证隐私保护基因序列分析方案(LWPPGS). 方案. 该方案实现了对大规模多用户源基因密文数据的批量计算和处理, 以及在密文域上用于病例对照差异的等位基因卡方检验分析, 从而对功能机制进行有效评估. 最终, 将密文域上的基因组序列分析结果返回给用户, 仅拥有授权用户可高效验证基因序列分析结果的正确性并解密得到明文的分析结果.

3)形式化的安全性证明和仿真实验结果表明本文提出的 VMK-HDEM 机制和 LWPPGS 具备 CPA 安全性, 且在数据拥有者、云服务器与接收方端的计算、通信效率方面均得到显著提升. 在密文域上对 L 个不同输入实例进行外包函数打包计算时, 用户端Sen_i公钥加密的使用次数复杂度为 $O(L)$, 即与其数据集的大小n_i无关, 大大降低了资源受限的本地用户计算开销; 在可验证性方面, 云服务器生成的密文计算结果正确性验证证据大小复杂度为$O(deg_F)$(其中deg_F代表外包计算函数的阶), 与用户数据集大小n_i无关.

1 相关工作

本节主要针对国内外可验证隐私保护基因序列分析工作中的代表性关键密码技术, 即全同态加密和可验证计算进行阐述.

1)全同态加密

全同态加密(fully homomorphic encryption, FHE)的概念由 Rivest, Adleman 和 Dertouzos[15]在 1978 年提出. 全同态加密支持在没有对密文输入数据进行解密的情况下, 即在密文域上进行任意函数F的计算. 授权用户对函数F在密文域上计算后得到的密文计算结果解密后, 可得到与直接在明文域上计算函数F相同的结果[16-17]. 因此, 可以利用全同态加密将数据

加密后上传至云服务器，云服务器在密文域上计算得到的结果解密后与直接通过明文计算得到的结果相同. 通过这种方式，存储、计算资源受限的本地用户可租用云服务器丰富的计算资源进行外包计算，同时保护数据隐私. 2009 年 Gentry[18] 提出了第一个可支持任意函数计算的全同态加密方案，该方案基于多项式环上的理想格，但方案非常复杂且效率低下，实际应用困难. 2009 年 Dijk 等人 [19] 对上述基于理想格的方案进行改进，基于 AGCD 困难问题，提出了一种整数上的同态加密方案，即 DGHV 方案，但存在公钥数目过大和长度过长的问题. Brakerski 等人[20] 基于错误学习（learning with errors, LWE），避开了理想格问题和稀疏子集问题，提出了采用重线性化技术来解决噪声增长问题的同态加密方案，即 BV 方案. 随后，Brakerski 等人[21] 又基于环上的错误学习（RLWE）提出了层次型同态加密方案——BGV，其中提到的密钥转换和模转换技术对之后的研究影响颇深. 之后，Gentry 等人[22] 基于矩阵思想和 LWE 安全假设提出了 GSW 全同态加密方案. 其同态加法和同态乘法就是矩阵上的加法和矩阵上的乘法，并且没有用到密钥转换算法，但该方案需要较大的密钥尺寸和计算开销，这使得它在实际应用中需要更高的硬件资源和计算能力. Cheon 等人[5] 提出了一个近似计算同态加密算法，即 CKKS 方案. 其具体构造基于 BGV 方案，但也可以基于其他类型的全同态加密方案. CKKS 方案是目前对实数和复数进行近似同态计算最有效的方法. 为了支持更深的乘法电路，Lee 等人[23] 和 Liu 等人[24] 分别提出了具有较小评估密钥的 FHEW 高效自举方案和多项式模数下支持打包的批量自举方案. 在多密钥全同态加密方面，López-Alt 等人[2] 提出了一个多密钥外包计算方案，但该方案的解密开销与用户数相关. Chen 等人[4] 基于 BFV 方案和 CKKS 方案[5]，给出了对应的多密钥全同态加密方案. 该方案提出了一种针对打包密文生成重线性化密钥的新方法. Kwak 等人[16] 提出了一种实用的多密钥 TFHE 同态加密方案. 但文献 [2, 4, 16] 方案所需计算开销和通信开销仍十分巨大，难以适应基于大规模数据集与复杂函数的多用户隐私保护基因序列分析，成为资源受限的本地用户的性能瓶颈. 近年来，Zhou 等人[25] 基于任意单向陷门置换提出了一种高效的多用户安全外包计算方案，但未考虑恶意敌手模型下的密文计算结果正确性验证问题.

2）可验证计算

外包计算的可验证性是指客户端希望将在输入 x 上的函数 F 的计算委托给恶意敌手模型下的云服务器完成. 如果服务器回复 y，客户端的目标是用比计算函数 F 花费更少的开销验证 $y=F(x)$ 的正确性. Gennaro 等人[26] 引入了可验证计算的概念，并设计了一种基于混淆电路的非交互式可验证计算方案. Chung 等人[27] 利用全同态加密方案构造了一个使用较小公钥的非交互式可验证计算方案. 此外，Parno 等人[28] 设计了一种基于 CP-ABE 的公开可验证计算方案. Papamanthou 等人[29] 提出了一种云环境下新的动态计算验证模型. 为了支持多用户系统，Choi 等人[30] 提出了一种使用代理无关传输方案的多用户非交互式可验证计算方案. 为了进一步实现多用户环境下的高效公开可验证，近年来，Baum 等人[31] 提出了一种支持可识别中止和高效公开可验证的常数轮安全多方计算协议. 上述边缘计算结果的正确性可验证方案主要通过 Yao 的混淆电路（garbled circuit）或双线性配对实现，计算开销仍较大，无法满足可验证基因序列分析任务中低功耗的本地用户性能需求. 在多服务器模型下，基于公开可验证的秘密共享技术，给出了可验证的外包多项式计算方案[32-33]. 但该方案无法保护用户的数据隐私和计算结果隐私，且需要用户间交互，带来较大的通信开销. 此外，文献 [22–33] 方案每进行一次计算，云服务器就需要生成一个验证证据且用户需要进行一次计算结果正确性验证，为计算结果接收方带来的验证计算开销与通信开销也极大.

不同于国内外目前的大多数多用户可验证外包计算方案[26-36]，本文基于混合加密原则，通过减少公钥加密使用次数，针对多个输入实例设计批量函数计算和批量生成验证证据新方法，大大降低了资源受限用户端和云服务器端的计算开销与通信开销. 构建了轻量级多用户可验证全同态数据封装机制及高效的可验证隐私保护基因序列分析方案.

2 前置知识

定理 1. 中国剩余定理（Chinese remainder theorem, CRT）[9]. 假设正整数 $m_1, m_2, \cdots, m_n$ 两两互素，则同余方程组：

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ \quad\vdots \\ x \equiv a_n \pmod{m_n} \end{cases},$$

有整数解，并且在模 $M = m_1 * m_2 * \cdots * m_n$ 的情况下解

是唯一的，为

$$x \equiv \left(a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} + \cdots + a_n M_n M_n^{-1}\right) \bmod M,$$

其中 $M_i = \dfrac{M}{m_i}$，M_i^{-1} 为 M_i 模 m_i 的逆元.

定理 2. 标记程序(labeled program)[37]. 一个标记的程序 P 为一个元组 $(f, l_1, \cdots, l_n)$，其中 $f: M^n \to M$ 是一个 n 元变量的函数(如一个电路)，$l_i \in \{0,1\}^*$ 是 f 第 i 个输入的标签. 标记程序可由下面方式组成：给定 $P_1, \cdots, P_t$ 和一个函数 g：$M^t \to M$，所组成的程序 P^* 是通过计算 g 在 $P_1, \cdots, P_t$ 上的输出得到的，表示为 $P^* = g(P_1, \cdots, P_t)$. P^* 的标记输入都是 $P_1, \cdots, P_t$ 不同标记的输入(所有具有相同标记的输入被分到同一组，然后作为 P^* 的一类输入). 令 $f_{id}: M \to M$ 为恒等函数，$l \in \{0,1\}^*$ 为任意标签. 记 $I_l = (f_{id}, l)$ 为带标签 l 的标识程序. 值得注意的是，一个程序 $P = (f, l_1, \cdots, l_n)$ 可以被表示为由 n 个恒等程序 $P = f\left(I_{l_1}, \cdots, I_{l_n}\right)$ 组成.

3 方案模型

本节首先给出可验证多密钥同态数据封装机制(VMK-HDEM)的算法定义与安全模型，包括数据隐私安全模型和可验证性安全模型. 并在此基础上，给出建立在 VMK-HDEM 方案基础上构造的多用户、多数据协同的轻量级可验证隐私保护基因序列分析方案(LWPPGS)的安全目标.

3.1 VMK-HDEM 系统模型

VMK-HDEM 的系统模型如图 1 所示.

1)假定每个数据拥有者 $Sen_i (i = 1,2,\cdots,n_s)$ 的数据集大小是 n_i，在一次函数计算中，Sen_i 有 L 条隐私输入数据实例 $m_{i,l} (l = 1,2,\cdots,L)$，其中 L 是参与外包函数计算的输入数据实例数且 $L \leqslant n_i$，并将这些输入实例用各自的密钥进行加密后形成密文 C_{Sen_i} 以及密文输入的认证码 σ_i，上传给云平台；

2)不失一般性的，在一对合作但不共谋的云服务器 SER 和密码服务器提供商 CSP 场景下，在密文域上计算多元多项式函数 $F\left(x_1, x_2, \cdots, x_{n_s}\right) = \sum_{j=1}^{K} a_j \prod_{i=1}^{n_s} x_i^{t_{i,j}}$，记函数的阶为 deg_F，得到函数的密文计算结果 C_F，以及云服务器 SER 利用外包函数 F 和 σ_i 生成的密文计算结果正确性验证证据 σ；

3)云服务器将密文计算结果 C_F 和计算结果验证证据 σ 返回给接收方 REC；

4)授权的接收方 REC 验证外包计算结果的正确性，验证通过后进一步解密函数计算结果.

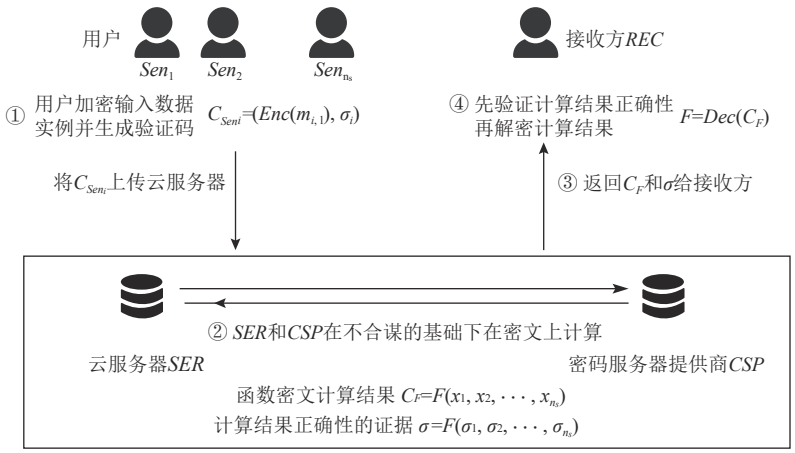


Fig. 1 Network architecture of VMK-HDEM

图 1 VMK-HDEM 的网络架构

3.2 VMK-HDEM 方案算法定义

本文提出的VMK-HDEM方案包含 5 个算法，分别是 $VMK\text{-}HDEM.Setup$，$VMK\text{-}HDEM.KeyGen$，$VMK\text{-}HDEM.Enc$，$VMK\text{-}HDEM.Eval$，$VMK\text{-}HDEM.Dec$. 下面分别给出这 5 个算法的输入输出定义：

1)$VMK\text{-}HDEM.Setup(1^\lambda) \to (PPR, sk_{PKE,SER}, sk_{PKE,CSP}, sk_{PKE,REC})$. 输入安全参数 λ，生成云服务器 SER、密码服务提供商 CSP 和授权接受方 REC 的公私钥 $\left(pk_{PKE,SER}, sk_{PKE,SER}\right)$，$\left(pk_{PKE,CSP}, sk_{PKE,CSP}\right)$ 和 $\left(pk_{PKE,REC}, sk_{PKE,REC}\right)$，输出公共参数 $PPR = (\lambda, pk_{PKE,SER}, pk_{PKE,CSP}, pk_{PKE,REC}, ID, T)$ 其中 ID 是数据拥有者的身份标识符 i 的集合，T 是标签 τ 的标签空间.

2)$VMK\text{-}HDEM.KeyGen(PPR) \to (pbk_i, pvk_i, pbk_{CSP}, pvk_{CSP}, k_i)$. 输入公开参数 PPR，输出数据拥有者 Sen_i 的临时公钥 pbk_i 和临时私钥 pvk_i、密码服务提供商 CSP 的临时公钥 pbk_{CSP} 和临时私钥 pvk_{CSP}，以及由数

据拥有者Sen_i秘密保存的标签生成与验证密钥k_i, k_i包括一个伪随机函数密钥$k_{\text{pse},i}$以及一个用户生成和验证标签的种子x_i.

3) $VMK\text{-}HDEM.Enc(PPR, pvk_i, pbk_i, \{m_{i,l}\}(i=1,2,\cdots, n_s; l=1,2,\cdots,L), k_i) \to C_{Sen_i}$. 输入公开参数$PPR$、数据拥有者$Sen_i$的临时公私钥$pbk_i$和$pvk_i$、消息$\{m_{i,l}\}$和标签生成验证密钥$k_i$, 输出密文$C_{Sen_i}$.

4) $VMK\text{-}HDEM.Eval(PPR, F, sk_{PKE,SER}, sk_{PKE,CSP}, pbk_{CSP}, pvk_{CSP}, C_{Sen_i}(i=1,2,\cdots,n_s)) \to C_{REC,SER}$. 输入公开参数$PPR$、多元多项式函数$F$、云服务器$SER$的私钥$sk_{PKE,SER}$、密码服务提供商$CSP$的私钥$sk_{PKE,CSP}$、密码服务提供商$CSP$的临时公私钥对$pbk_{CSP}$和$pvk_{CSP}$以及密文$C_{Sen_i}$, 输出密文$C_{REC,SER}$.

5) $VMK\text{-}HDEM.Dec(PPR, sk_{PKE,REC}, C_{REC,SER}, F) \to F(m_{1,l}, m_{2,l}, \cdots, m_{n_s,l})$或$\perp$. 输入公开参数$PPR$、授权接受方$REC$的私钥$sk_{PKE,REC}$、密文$C_{REC,SER}$以及函数$F$, 输出$L$个函数的运算结果$F\left(m_{1,l}, m_{2,l}, \cdots, m_{n_s,l}\right)$,其中$l=1,2,\cdots,L$; 或者输出$\perp$, 即当验证失败时无法输出正确的函数计算结果.

3.3 VMK-HDEM 方案安全模型

3.3.1 VMK-HDEM 方案数据隐私安全模型

首先我们给出VMK-HDEM方案的数据输入隐私和输出隐私.

1)输入隐私. 该协议的输入隐私为诚实数据拥有者的本地数据集中的输入数据安全, 能有效抵抗被俘获的数据拥有者、2个合作不合谋的服务器之一和被俘获的数据接收方发起的合谋攻击.

2)输出隐私. 该协议的输出隐私为外包函数计算结果仅能由授权的接收方正确解密获得. 具体而言, 外包函数的计算结果数据安全能有效抵抗被俘获的数据拥有者与2个合作不合谋的服务器之一发起的合谋攻击.

下面形式化定义VMK-HDEM方案的数据隐私安全模型. 为描述简洁, 我们以输入数据隐私安全模型为例给出其形式化安全模型和定义, 输出隐私安全模型可同样定义. 给定一个诚实数据拥有者$Sen_{i'}$的输入数据$m_{i',l}(i' \in ID; l=1,2,\cdots,L)$, $Sen_{i'}$上传它的密文输入数据给云服务器SER进行隐私保护的外包函数计算. 对于VMK-HDEM方案, 通过一个模拟者B和概率多项式时间能力的敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$之间的游戏$DP\text{-}CPA_{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda)$来形式化定义数据隐私安全模型, 下面我们给出该游戏的各个阶段:

1)初始化阶段. 输入安全参数1^λ, 模拟者B运行$VMK\text{-}HDEM.Setup(1^\lambda)$来生成云服务器$SER$、密码服务提供商$CSP$和授权接受方$REC$的公私钥$\left(pk_{PKE,SER}, sk_{PKE,SER}\right)$, $\left(pk_{PKE,CSP}, sk_{PKE,CSP}\right)$和$\left(pk_{PKE,REC}, sk_{PKE,REC}\right)$. 并生成公共参数$PPR=(\lambda, pk_{PKE,SER}, pk_{PKE,CSP}, pk_{PKE,REC}, ID, T)$. 接着运行$VMK\text{-}HDEM.KeyGen(PPR)$生成数据拥有者$Sen_i(i=1,2,\cdots,n_s)$的临时公私钥$pbk_i$和$pvk_i$、$CSP$的临时公私钥$pbk_{CSP}$和$pvk_{CSP}$以及由数据拥有者$Sen_i$秘密保存的用于生成和验证标签的密钥$k_i$. 最后将$(PPR, \{pbk_i\}, pbk_{CSP})$都返回给敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$.

2)查询腐化谕言机O^{Cor}. 定义初始腐化集合$L_{\text{corr}}=\varnothing$, 敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$可以俘获除诚实数据拥有者$Sen_{i'}$以外的数据拥有者$Sen_j(j=1,2,\cdots,n_s$且$j \neq i')$、授权接收方以及云服务器$SER$和密码服务提供商$CSP$中的一个. 敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$可以利用$O^{\text{Cor}}$查询$O^{\text{Cor}}(j)$, 获得对应的被腐化数据拥有者的私钥$pvk_j$, 模拟者$B$更新$L_{\text{corr}} \leftarrow L_{\text{corr}} \cup \{j\}$; 还可以查询$O^{\text{Cor}}(SER)$或$O^{\text{Cor}}(CSP)$以及$O^{\text{Cor}}(REC)$分别获得$sk_{PKE,SER}$或$(sk_{PKE,CSP}, pvk_{CSP})$以及$sk_{PKE,REC}$、模拟者$B$相应的更新腐化集合.

3)挑战阶段. 敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$选择一对挑战明文消息$m_{i',l,0}, m_{i',l,1}(l=1,2,\cdots,L)$, 且$|m_{i',l,0}|=|m_{i',l,1}|$, 并将这对挑战明文发送给模拟者$B$. B随机选择$b \in_R \{0,1\}$, 输出挑战密文$C_{i'}^* = VMK\text{-}HDEM.Enc(PPR, pvk_{i'}, pbk_{i'}, m_{i',l,b}, k_{i'})$给敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$.

4)猜测阶段. 敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$收到挑战密文$C_{i'}^*$后进行猜测, 输出$b' \in \{0,1\}$. 如果$b'=b$, 则说敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$成功赢得了游戏$DP\text{-}CPA_{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda)$; 反之则失败.

定义 1. 数据隐私安全. 在上述游戏$DP\text{-}CPA_{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda)$中, 我们定义概率多项式时间能力的敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$获胜的优势为$Adv_{DP\text{-}CPA}^{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda) = \left|Pr[b'=b] - \frac{1}{2}\right|$. 那么, 所提出的VMK-HDEM方案能实现数据隐私的CPA(选择明文攻击)安全性, 当且仅当对于任意概率多项式时间能力的敌手$\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$, 上述优势都满足不等式$Adv_{CPA}^{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda) \leqslant \epsilon(\lambda)$, 其中$\epsilon(\lambda)$为$\lambda$上的可忽略函数.

3.3.2 VMK-HDEM 方案可验证性安全模型

我们首先给出VMK-HDEM方案的可验证性目标. 在VMK-HDEM方案中, 多元多项式函数$F\left(x_1, x_2, \cdots, x_{n_s}\right)$的每一个输入数据$x_i$都有一个验证码$\sigma_i$. 我们的可验证性安全模型保证了敌手在不知道相关密钥的情况下, 只能获得合法用户已生成的验证码$\sigma_i(i=1,2,\cdots,n_s)$, 或合法用户已生成的验证码上的函数计算结果$\sigma\left(\sigma = F\left(\sigma_1, \cdots, \sigma_{n_s}\right)\right)$. 直观地说, 我们的可验证性安全模型保证了当云服务器计算得出错误

的函数结果时,授权的接收方接受此结果的概率是可忽略的.

下面我们形式化的定义 VMK-HDEM 方案的可验证性(不可伪造性)安全模型. 对于 VMK-HDEM 方案,我们通过一个模拟者 B 和敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 之间的实验 $HomUF\text{-}CMA_{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)$ 来形式化定义可验证性安全模型,下面我们介绍该实验的 5 个阶段:

1)初始化. 模拟者 B 运行 $VMK\text{-}HDEM.Setup(1^{\lambda})$ 来生成云服务商 SER、密码服务提供商 CSP 和接受方 REC 的公私钥 $\left(pk_{PKE,SER}, sk_{PKE,SER}\right)$, $\left(pk_{PKE,CSP}, sk_{PKE,CSP}\right)$ 和 $\left(pk_{PKE,REC}, sk_{PKE,REC}\right)$. 并生成公共参数 $PPR=(\lambda, pk_{PKE,SER}, pk_{PKE,CSP}, pk_{PKE,REC}, ID, T)$($ID$ 是数据拥有者的身份标识符 i 的集合,T 是标签 τ 的标签空间). 并将 PPR 发送给敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$.

2)查询认证码生成谕言机 $O^{\text{Mac-Gen}}$. 敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 可以适应性向模拟者提交表单 (Δ, l', m) 进行查询. 其中 Δ 是一个数据集的标识符; $l'=(i,\tau)$, 其中 $i\in ID$, $\tau\in T$; $m\subseteq M$, 是消息总空间 M 中的一个子集. 模拟者 B 针对敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 不同的查询,回答如下:

① 如果 (Δ, l', m) 中的 Δ 代表的数据集是第一次被查询,模拟者 B 初始化一个空的列表 $L_{\Delta}=\varnothing$ 并且继续执行②.

② 如果 (Δ, l', m) 中的 l' 的身份标识符 i 是第一次被查询,模拟者 B 运行密钥生成算法生成用于生成和验证认证标签的密钥 $k_i \xleftarrow{\$} VMK\text{-}HDEM.KeyGen(PPR)$(此处\$代表密钥生成算法 $VMK\text{-}HDEM.KeyGen$ 是一个概率性算法,下文出现的\$符号含义相同),除了生成 k_i,还生成数据拥有者 Sen_i 的临时公私钥 (pbk_i, pvk_i) 和密码服务提供商 CSP 的临时公私钥对 (pbk_{CSP}, pvk_{CSP}). 并且 k_i 隐式地与数据拥有者的身份标识符 i 绑定. 模拟者 B 继续执行③.

③ 如果 (Δ, l', m) 满足 $(l', m)\notin L_{\Delta}$, 模拟者 B 运行加密算法得到认证码 $\sigma_{l'} \xleftarrow{\$} VMK\text{-}HDEM.Enc\left(PPR, pvk_i, pbk_i, m, k_i\right)$(注意这里只运行 $VMK\text{-}HDEM.Enc$ 中生成认证码的部分),并返回认证码 $\sigma_{l'}$ 给敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$, 并更新集合 $L_{\Delta}\leftarrow L_{\Delta}\cup\{l', m\}$.

④ 如果 (Δ, l', m) 满足 $(l', \cdot)\in L_{\Delta}$(意味着敌手已经对于某个集合 m' $(m\subseteq m')$ 进行过查询 (Δ, l', m')),那么 B 忽略此次查询.

3) 查询认证码验证谕言机 $O^{\text{Mac-ver}}$. 敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 可以通过查询验证谕言机,即敌手可以提交一个查询 (P, Δ, m, σ). 其中,P 是一个标记程序,可以记为 $P=\left(F, l'_1, l'_2, \cdots, l'_{n_s}\right)$, F 是一个有 n_s 个变量的函数,l'_i 是 F 第 i 个输入的标签; Δ 代表某个数据集; m 是消息空间的一个消息; σ 是敌手想要验证的认证码. 模拟者 B 计算 $VMK\text{-}HDEM.Dec(P, \Delta, \{k_i\}_{i\in P}, m, \sigma)$(注意这里 B 只运行 $VMK\text{-}HDEM.Dec$ 中的验证认证码部分),其中 $\{k_i\}_{i\in P}$ 是指所有和程序 P 中涉及到的数据拥有者的标识符 i 绑定的密钥的集合. 若验证成功,代表验证码 σ 是消息 m 对应的合法认证码,模拟者 B 输出 1; 反之,代表验证失败,模拟者 B 输出 0. 模拟者 B 将结果返回给敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$.

4)查询腐化谕言机 $O^{\text{Mac-Cor}}$. 敌手可以通过查询腐化谕言机来获取被腐化的数据拥有者的私钥. 具体地说,在游戏最初,模拟者 B 初始化一个为空的腐化用户集合 $L_{\text{corr}}=\varnothing$. 在游戏过程中,敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 可以适应性地使用任意数据拥有者的标识符 i 查询其密钥. 如果 $i\notin L_{\text{corr}}$, 那么腐化的谕言机将密钥 k_i 返回给敌手(如果 i 是第一次被查询则运行 $VMK\text{-}HDEM.KeyGen$ 来生成密钥 k_i),然后更新列表 $L_{\text{corr}}\leftarrow L_{\text{corr}}\cup\{i\}$. 如果 $i\in L_{\text{corr}}$, 那么腐化的谕言机返回之前已生成的密钥 k_i.

5)伪造阶段. 在游戏的最后,敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 输出一个元组 $(P^*, \Delta^*, m^*, \sigma^*)$. 如果 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 输出的元组是一个成功伪造(根据定义 2),游戏将输出 1,否则输出 0.

定义 2. 成功伪造. 已知实验 $HomUF\text{-}CMA_{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)$ 中 $(P^*, \Delta^*, m^*, \sigma^*)$ 是敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 在实验结束时返回的元组,其中 $P^*=\left(F^*, l_1^*, \cdots, l_t^*\right)$, Δ^* 是某个数据集的标识符, $m^*\in M$, σ^* 是一个认证码. 如果对于任何 $i\in P^*$ 且 $i\notin L_{\text{corr}}$(即 P^* 中没有被腐化的用户), $VMK\text{-}HDEM.Dec(P^*, \Delta^*, \{k_i\}_{i\in P^*}, m^*, \sigma^*)$(注意这里只运行解密算法中的验证认证码的部分)的运算结果都输出 1,并且下列 3 种条件中任一条件能够被满足,则敌手输出的 $(P^*, \Delta^*, m^*, \sigma^*)$ 是一个成功的伪造.

1)条件 1. L_{Δ^*} 在游戏过程中没有被初始化(即数据集 Δ^* 没有被查询过).

2)条件 2. 对于所有的 $i\in[t]$, 存在 $\exists(l_i^*, m_i)\in L_{\Delta^*}$, 但 $m^*\neq F^*(m_1, m_2, \cdots, m_t)$. 即敌手不能在已经正确生成了认证码的消息的函数计算结果上做伪造.

3)条件 3. 存在一个标签 l^* 满足 $(l^*, \cdot)\notin L_{\Delta^*}$, 即敌手从未用 l^* 进行过查询.

定义 3. 可验证性安全. 在上述游戏 $HomUF\text{-}CMA_{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)$ 中,我们定义概率多项式时间能力的敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$ 获胜的优势为 $Adv_{HomUF\text{-}CMA}^{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)=Pr[HomUF-CMA_{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)=1]$. 那么,所提出的 VMK-

HDEM方案能实现可验证性安全(不可伪造性)，当且仅当对于任意概率多项式时间能力的敌手$\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$，上述优势都满足不等式$Adv_{HomUF\text{-}CMA}^{\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}}(\lambda)\leqslant\epsilon(\lambda)$，其中$\epsilon(\lambda)$为$\lambda$上的可忽略函数.

3.4 LWPPGS 方案安全目标

3.4.1 LWPPGS 方案数据隐私安全目标

LWPPGS 的数据隐私安全目标定义有：

1)输入隐私. LWPPGS 方案的输入隐私是指多个用户的输入基因序列信息，即对应一段基因序列的碱基对排序. 用户将基因序列数据加密上传到服务器，并确保 2 个合作不合谋的服务器之一无法与其他被俘获的用户合谋获取诚实用户的基因信息.

2)输出隐私. LWPPGS 方案的输出隐私是指用户的基因序列的分析结果. 分析结果仅能由授权的接收方正确解密获得. 具体而言，计算基因序列的外包函数的计算结果的数据隐私安全能够有效抵抗被俘获的数据拥有者与 2 个合作不合谋的服务器之一发起的共谋攻击.

由于 LWPPGS 方案是基于可验证的多密钥同态数据封装机制 VMK-HDEM 构建的，因此其数据隐私安全可以规约到 VMK-HDEM 的数据隐私安全上.

3.4.2 LWPPGS 方案可验证性安全目标

LWPPGS 的可验证性安全目标是指当云服务器计算得出错误的基因序列分析结果时，用户接受该结果的概率是可忽略的. 与 LWPPGS 方案的数据隐私安全模型相同，由于 LWPPGS 方案是基于 VMK-HDEM 方案构建的，因此其可验证性安全可以规约到 VMK-HDEM 的可验证性安全上.

4 可验证的多密钥同态数据封装机制

本节中，我们基于混合加密原则，提出了一个可验证的多密钥同态数据封装机制 VMK-HDEM. 利用一个任意的CPA安全的公钥加密算法PKE作为黑盒(其包括初始化算法$PKE.Setup$、密钥生成算法$PKE.KeyGen$、加密算法$PKE.Enc$和解密算法$PKE.Dec$)用于加密对称密钥；并设计了一个具有被动攻击下不可区分安全性的对称同态加密算法加密数据. 同时，构造了一个多密钥同态消息认证码来共同组成 VMK-HDEM. 方案的可验证性是指接收方能有效验证密文计算结果的正确性，防止恶意云服务器输入错误计算结果. 为了实现可验证性，每一个数据拥有者(发送方)Sen_i将对自己的带有标签l_i的密文输入数据进行计算以生成一个认证码σ_i，然后云服务器将生成密文计算结果正确性的证据$\sigma=F(\sigma_1,\cdots,\sigma_{n_s})$. VMK-HDEM 方案中用户外包计算的函数为多元多项式形式的函数. 在下面的方案描述过程中，为方便起见，我们令n_s(数据拥有者个数)为函数的变元个数，令$F\left(x_1,x_2,\cdots,x_{n_s}\right)=\sum_{j=1}^{K}a_j\prod_{i=1}^{n_s}x_i^{t_{i,j}}$，即函数计算的$n_s$个自变量分别来自$n_s$个$Sen_i$. 值得一提的是，每个发送方$Sen_i$的数据集大小是$n_i$，所构造的 VMK-HDEM 方案支持对$L$个不同输入实例的批量函数计算. 即$Sen_i$每次从其数据集中选择$L$个输入数据进行加密并打包($L\leqslant n_i$)，再将打包密文输入发送给云服务器，使其能在$L$组已打包的$n_s$维密文输入数据上进行一次函数运算，批量得到$L$个函数密文计算结果以及一个计算结果正确性验证证据$\sigma$. 最后，接收方可以使用$\sigma$对$L$个函数密文计算结果的正确性进行批量验证，并解密分别得到这L次函数计算的明文计算结果.

4.1 VMK-HDEM 协议的符号描述

本文中 VMK-HDEM 的符号描述如表 1 所示.

Table 1 Table for Describing VMK-HDEM Notations

表 1 VMK-HDEM 符号描述表

符号	描述
$m_{i,l}$	数据拥有者Sen_i的第l个数据
$F\left(x_1,x_2,\cdots,x_{n_s}\right)$	多用户外包计算的多元多项式
a_j	多项式F第j项$Item_j$的系数
K	多项式F的项数
$t_{i,j}$	变元x_i在多项式F第j项$Item_j$中的幂指数
deg_j	多项式F第j项$Item_j$的次数
deg_F	多项式F的次数，即$deg_F=\max\left(deg_1,deg_2,\cdots,deg_K\right)$
$F_{k_{\text{pse}}}$	带密钥k_{pse}的伪随机函数：$\{0,1\}^*\times\{0,1\}^{2\lambda}\to\{0,1\}^{2\lambda}$
ℓ_i	Sen_i的密文C_i的公开标签
σ_i	Sen_i的密文C_i的认证码

4.2 VMK-HDEM 方案

本节详细介绍 VMK-HDEM 方案的具体步骤.

1)$VMK\text{-}HDEM.Setup(1^\lambda)\to(PPR,sk_{PKE,SER},sk_{PKE,CSP},sk_{PKE,REC})$. 输入安全参数$\lambda$，任选一个 CPA 安全的公钥加密算法 PKE，运行$PKE.Setup(1^\lambda)$和$PKE.KeyGen(1^\lambda)$；输出 3 对公私钥，分别为云服务商 SER、密码服务提供商 CSP 和授权接受方 REC 的公私钥对$(pk_{PKE,SER},sk_{PKE,SER})$，$(pk_{PKE,CSP},sk_{PKE,CSP})$和$(pk_{PKE,REC},sk_{PKE,REC})$. 私钥$SK=(sk_{PKE,SER},sk_{PKE,CSP},sk_{PKE,REC})$分别由云服务器 SER，密码服务提供商 CSP，授权接受方 REC 各自秘密保存. 数据拥有者的身份标识符i的集合$ID=[n_s]$是一个整数界限$n_s\in\mathbb{N}$. 标签τ的标签空间T由任意二进

制字符串组成，即$T=\{0,1\}^*$. 最后参数生成算法输出公开参数$PPR=(\lambda,pk_{PKE,SER},pk_{PKE,CSP},pk_{PKE,REC},ID,T)$.

2）$VMK\text{-}HDEM.KeyGen(PPR)\rightarrow(pbk_i,pvk_i,pbk_{CSP},pvk_{CSP},k_i)$. 每个数据拥有者$Sen_i(i=1,2,\cdots,n_s)$随机选择$2L+1$个大素数$p_{i,1},p_{i,2},\cdots,p_{i,L},s_i,q_{i,1},q_{i,2},\cdots,q_{i,L}$. 令$q_{i,\min}=\mathrm{Min}\{q_{i,l}\}$,消息空间为$Z^*_{q_{i,\min}}$, 满足$|p_{i,l}|=|q_{i,l}|=|s_i|=\lambda(l=1,2,\cdots,L)$. 计算$N_i=\prod_{l=1}^{L}p_{i,l}$. 计算$T_i=s_i\prod_{l=1}^{L}p_{i,l}$, 令$N_i=p_{i,l}\cdot M_{i,l}(l=1,2,\cdots,L)$,其中$M_{i,l}\cdot M'_{i,l}\equiv 1\pmod{p_{i,l}}$. 数据拥有者$Sen_i$的临时公钥和私钥分别为$pbk_i=(T_i,\{q_{i,l}\})$, $pvk_i=(\{p_{i,l}\},s_i,N_i)(l=1,2,\cdots,L)$.$CSP$随机选择$2L+1$个大素数$p_1,p_2,\cdots,p_L,s,q_1,q_2,\cdots,q_L$, 满足$|p_1|=|p_2|=\cdots=|p_L|=|q_1|=\cdots=|q_L|=|s|=\lambda$. 计算$N=\prod_{l=1}^{L}p_l$. 令$N=p_l\cdot Q_l(l=1,2,\cdots,L),Q_l\cdot Q'_l\equiv 1(\mathrm{mod}\ p_l),(l=1,2,\cdots,L)$. 计算$T=s\prod_{l=1}^{L}p_l$, CSP的临时公私钥为$pbk_{CSP}=T$, $pvk_{CSP}=(\{p_l\},\{q_l\},s,N)$. 接着通过一个伪随机函数生成器生成伪随机函数密钥k_{pse}, 以及带密钥的伪随机函数$F_{k_{\mathrm{pse}}}:\{0,1\}^*\times\{0,1\}^{2\lambda}\rightarrow\{0,1\}^{2\lambda}$. 最后$Sen_i$选择一个随机数$x_i\in Z^*_{T_i}$, 输出用于生成和验证认证标签的密钥$k_i=(k_{\mathrm{pse}},x_i)$, k_i由数据拥有者Sen_i秘密保存.

3）$VMK\text{-}HDEM.Enc(PPR,pvk_i,pbk_i,\{m_{i,l}\}(i=1,2,\cdots,n_s;l=1,2,\cdots,L),k_i)\rightarrow C_{Sen_i}$. 首先定义认证码空间，每个认证码$\sigma$包含一对$(I,y)$, 其中集合$I\subseteq ID$, 多项式$y$位于$n_s$元多项式环$F_{q_{1,\min},q_{2,\min},\cdots,q_{n_s,\min}}[X_1,X_2,\cdots,X_{n_s}]$中. 每个拥有数据$m_{i,l}$的数据拥有者$Sen_i$计算$m_{i,l,p_{i,l}}=m_{i,l}\ (\mathrm{mod}\ p_{i,l})$, $(l=1,2,\cdots,\mathrm{L})$, 然后随机选择$r_{i,0}$, $r_{i,1}\in\{0,1\}^{2\lambda}$, $k_{i,l}\in\{0,1\}^{\lambda}$, 其中$r_i\in Z^*_T$ 并且计算

$$C_{i,SER}=PKE.Enc_{pk_{PKE,SER}}(r_{i,0}),$$

L是偶数时：$C^1_{i,CSP,l}=PKE.Enc_{pk_{PKE,CSP}}(p_{i,l}\|p_{i,l+1})$

$$(l=1,3,5,\cdots,L-1);$$

L是奇数时：$C^1_{i,CSP,l}=PKE.Enc_{pk_{PKE,CSP}}(p_{i,l}\|p_{i,l+1})$

$$(l=1,3,5,\cdots,L-2),$$

$$C^1_{i,CSP,l}=PKE.Enc_{pk_{PKE,CSP}}(p_{i,l}\|0^{\lambda})(l=L).$$

数据拥有者Sen_i可以在离线状态下完成上述操作.

对给定的L, 对密文进行加密：

$$C_i=r_{i,0}\left(\sum_{l=1}^{L}M^{-1}_{i,l}M_{i,l}\left(m_{i,l}+k_{i,l}q_{i,l}\right)_{p_{i,l}}\right)+r_{i,1}N_i\bmod T_i, \tag{1}$$

其中$M^{-1}_{i,l}M_{i,l}\equiv 1\pmod{p_{i,l}}$. 接着$Sen_i$首先为密文$C_i$生成标签$l_{i'}=(i,\tau_i)\in ID\times T$, 其中$i$为$Sen_i$的身份标识符, 标签$\tau_i$通过连接它们的性质和索引产生，比如$\tau_i=$“输入”$\|i.Sen_i$将标签$l_{i'}$公开. 接着为密文$C_i$生成一个认证码$\sigma_i=(I_i,y_i)$,其中$I_i=i$, y_i是一个以X_i为变量的一次多项式. Sen_i首先将标签值代入密钥生成阶段生成的伪随机函数计算得到$r_{\tau_i}=F_{k_{\mathrm{pse}}}(\tau_i)$. 接着令$y_{0,i}=C_i$; $y_{1,i}=\dfrac{r_{\tau_i}-C_i}{x_i}$($x_i$是$KeyGen$算法中生成的认证标签密钥$k_i$的一部分), 再利用$(y_{0,i},y_{1,i})$来构建多项式$y_i(X_i)=y_{1,i}X_i+y_{0,i}$. 注意当自变量$X_i=0$时，多项式$y_i(X_i)=C_i$; 当自变量取$x_i$时, $y_i(X_i)=r_{\tau_i}$.

接着Sen_i用接收方REC的公钥加密认证标签密钥k_i：

$$C_{k_{\mathrm{pse}}}=PKE.Enc_{pk_{PKE,REC}}(k_{\mathrm{pse}}),$$

$$C_{x_i}=PKE.Enc_{pk_{PKE,REC}}(x_i).$$

令$C_{k_i}=(C_{k_{\mathrm{pse}}},C_{x_i})$, 数据拥有者$Sen_i$将密文$C_{Sen_i}=(C_{i,SER},\{C^1_{i,CSP,l}\}(l=1,2,\cdots,L),C_i,\sigma_i,C_{k_i})$发送给云服务器$SER$.

4）$VMK\text{-}HDEM.Eval(PPR,F,sk_{PKE,SER},sk_{PKE,CSP},pbk_{CSP},pvk_{CSP},C_{Sen_i}(i=1,2,\cdots,n_s))\rightarrow C_{REC,SER}$. 云服务器$SER$首先用其私钥$sk_{PKE,SER}$解密得$r_{i,0}=PKE.Dec_{sk_{PKE,SER}}(C_{i,SER})$, 然后发送$C_i$和$C^1_{i,CSP,l}(l=1,2,\cdots,L)$给$CSP$.

CSP首先用其私钥$sk_{PKE,CSP}$解密得$p_{i,l}=PKE.Dec_{sk_{PKE,CSP}}(C^1_{i,CSP,l})$. 接着随机选择$r_{i,CSP}\in\{0,1\}^{2\lambda}$, $k^{CSP}_{i,l}\in\{0,1\}^{\lambda}$, 并利用自己的临时公私钥进行重加密：

$$C'_{i,l}=(C_i\bmod p_{i,l})\bmod q_{i,l}=r_{i,0}m_{i,l}\bmod q_{i,l}, \tag{2}$$

$$C''_{i,l}=(C'_{i,l}+k^{CSP}_{i,l}q_l)\bmod p_l,(l=1,2,\cdots,L), \tag{3}$$

$$C''_i=\left(\sum_{l=1}^{L}Q_lQ'_lC''_{i,l}+r_{i,CSP}N\right)\bmod T. \tag{4}$$

最后，CSP将$C_{CSP}=(C''_i)$发送给SER. 不失一般性地，我们令外包多元多项式F的第j项$Item_j=a_j\prod_{i=1}^{n_s}x_i^{t_{i,j}}$ $(j=1,2,\cdots,K)$的次数为deg_j. 接收到C_{CSP}后，SER随机选取$r\in\{0,1\}^{2\lambda}$, 满足$r\in Z^*_T$, 然后计算：

$$C''_{i,SER}=r^{-1}_{i,0}C''_i,$$

$$C''_{i,SER}=rC''_{i,SER}.$$

接着计算：

$$C_{Item_j}=r^{deg_F-deg_j}a_j\prod_{i=1}^{n_s}(C''_{i,SER})^{t_{i,j}}, \tag{5}$$

$$C^{\mathrm{bld}}_F=\sum_{j=1}^{K}C_{Item_j}, \tag{6}$$

$$Y_{Item_j}=a_j\prod_{i=1}^{n_s}(y_i)^{t_{i,j}}, \tag{7}$$

$$Y_F^{\text{bld}}=\sum_{j=1}^{K}Y_{Item_j},\tag{8}$$

其中，Y_{Item_j}和Y_F^{bld}是n_s元的多元多项式，且Y_F^{bld}的阶为deg_F.

接着SER发送C_F^{bld}和Y_F^{bld}给CSP. CSP接收之后，计算：

$$C_{F,l}^{CSP,1}=\left(C_F^{\text{bld}}\bmod p_l\right)\bmod q_l,$$

$$C_{F,l}^{CSP,2}=PKE.Enc_{pk_{PKE,REC}}\left(C_{F,l}^{CSP,1}\right).\tag{9}$$

接着CSP发送$C_{F,l}^{CSP,2}$给SER，最后SER首先计算：

$$C_{REC,SER}=PKE.Enc_{pk_{PKE,REC}}(r).$$

SER再根据接收的n_s个认证码$\{\sigma_i:=(I_i,y_i)\}_i$，生成最终的关于密文域上函数计算结果正确性验证证据$\sigma=(I,y)$，其中$I=\bigcup_{i=1}^{n_s}I_i$，$y=Y_F^{\text{bld}}$.

最后SER将$C_{REC,SER}=(\{C_{F,l}^{CSP,2}\}(l=1,2,\cdots,L),C_{REC,SER},C_F^{\text{bld}},C_{k_i})$和认证码$\sigma$发给授权接收方$REC$.

5) $VMK\text{-}HDEM.Dec(PPR,l_{i'}\,(i=1,2,\cdots,n_s),sk_{PKE,REC},C_{REC,SER},\sigma,F)\to F_l\left(m_{1,l},m_{2,l},\cdots,m_{n_s,l}\right)$或$\perp$. 接收方首先需要验证密文域上外包计算结果的正确性，再对计算结果进行解密.

令$P=\left(F,l'_1,\cdots,l'_{n_s}\right)$，其中$F$为上述云服务器$SER$上的外包计算函数，$l_{i'}$为数据拥有者$Sen_i$公开的标签，且$l_{i'}=(i,\tau_i)$. 接着$REC$用自己的私钥解密得到用于验证函数计算结果正确性证据的密钥k_i，包括x_i和伪随机函数密钥k_{pse}：

$$x_i=PKE.Dec_{sk_{PKE,REC}}\left(C_{x_i}\right),$$

$$k_{\text{pse}}=PKE.Dec_{sk_{PKE,REC}}\left(C_{k_{\text{pse}}}\right).$$

再用得到的伪随机函数密钥计算：

$$r_{\tau_i}=F_{k_{\text{pse}}}(\tau_i).$$

又有$\sigma=(I,y)$，令$I=\left\{\bar{I}_1,\bar{I}_2,\cdots,\bar{I}_{n_s}\right\},\bar{I}_v\neq\bar{I}_u(v,u\in\{1,2,\cdots,n_S\}$且$v\neq u)$. REC接着验证下列3个等式，当且仅当3个等式都成立时，REC接收SER的计算结果并对其解密；反之验证失败，REC输出$\perp$.

$$\left\{\bar{I}_1,\bar{I}_2,\cdots,\bar{I}_{n_s}\right\}=\{i:i\in P\},\tag{10}$$

$$y(0,0,\cdots,0)=C_F^{\text{bld}},\tag{11}$$

$$y\left(x_1,x_2,\cdots,x_{n_s}\right)=F\left(r_{\tau_1},\cdots,r_{\tau_{n_s}}\right).\tag{12}$$

若验证成功，REC用自己私钥解密函数的计算结果：

$$r=PKE.Dec_{sk_{PKE,REC}}\left(C_{REC,SER}\right),$$

$$C_{F,l}^{CSP,1}=PKE.Dec_{sk_{PKE,REC}}\left(C_{F,l}^{CSP,2}\right)(l=1,2,\cdots,L).$$

最后得到L个函数计算结果：

$$F\left(m_{1,l},m_{2,l},\cdots,m_{n_s,l}\right)_l=r^{-deg_F}C_{F,l}^{CSP,1}(l=1,2,\cdots,L).\tag{13}$$

4.3 轻量级高效可验证隐私保护基因序列分析协议

在本节中，我们基于5.2节中提出的VMK-HDEM，设计了LWPPGS. 该方案能在密文上对用户的基因数据进行统计分析，实现用于病例对照差异的等位基因卡方检验.

4.3.1 等位基因卡方检验方案

我们首先介绍明文域上的等位基因卡方检验方案的计算方法. 等位基因卡方检验通常需要与多个SNP进行遗传关联测试. 以一对等位基因A和a为例，A为主要等位基因，a为次要等位基因. 一个SNP的数据可以以2种形式表示：一种是按基因型计数（例如aa, Aa和AA），另一种是根据等位基因A和a的疾病状态计数的列联表. 等位基因情况（A, a）的列联表如表2所示.

Table 2 Sample Contingency Tables Required for Chi-square Tests of Alleles

表2 等位基因的卡方检验所需示例列联表

等位基因	a	A	总计
对照组病例数	n_{00}	n_{01}	$2r_0$
实验组病例数	n_{10}	n_{11}	$2r_1$
等位基因计数	c_0	c_1	$2N$

表2中n_{00}是携带次要等位基因a的对照病例数，n_{01}是主要等位基因A的对照病例数，n_{10}是携带次要等位基因a的实验组病例数，n_{11}是携带主要等位基因A的实验组病例数；r_0是对照组病例数，r_1是实验组病例数；c_0表示次要等位基因的计数，c_1表示主要等位基因的计数，N是总病例数. y_i表示第i个个体的疾病状态，$y_i=1$表示个体患有疾病，反之，$y_i=0$；$s_i\in\{0,1,2\}$对应于个体i的等位基因变异数，其中，$n_{11}=y^{\mathrm{T}}\cdot s$，$r_1=2y^{\mathrm{T}}\cdot y$，$c_1=\sum_{i=0}^{N}s_i$.

通过计算卡方值χ^2，可确定预期频率和观察到的频率之间是否存在显著差异，从而计算关联强度OR：

$$\chi^2=\frac{2N(2n_{11}N-2c_1r_1)^2}{c_1(2N-c_1)2r_1\cdot(2N-2r_1)},$$

$$OR=\frac{n_{11}(n_{11}-2r_1-c_1+2N)}{(c_1-n_{11})(2r_1-n_{11})}.$$

值得一提的是，我们可以通过使用矢量化公式同时计算M个SNP的χ^2值和OR值. 此时，$u_b=\{1\}_{i=1}^{b},b=M,N;n_{11}=y^{\mathrm{T}}\cdot\boldsymbol{S},2r_1=2(y^{\mathrm{T}}\cdot y)u_M,c_1:=u_N^{\mathrm{T}}\cdot\boldsymbol{S}$，标量$2N$用矩阵$\boldsymbol{d}=2Nu_M$替代. 其中$\boldsymbol{S}\in\{0,1,2\}^{N\times M}$表示SNP矩阵，

$S_{i,j}$对应第i个个体的第j个 SNP.

并且根据χ^2和OR的表达式, 我们可以按照分子分母分离出 4 个式子, 分别计算出 4 个表达式的结果即可得到卡方值χ^2和关联强度OR的值:

$$Item_{j,\mathrm{num}} = 2N(2n_{11}N - 2c_1r_1)^2,$$

$$Item_{j,\mathrm{den}} = c_1(2N - c_1)2r_1 \cdot (2N - 2r_1),$$

$$Item_{j,\mathrm{ornum}} = n_{11}(n_{11} - 2r_1 - c_1 + 2N),$$

$$Item_{j,\mathrm{orden}} = (c_1 - n_{11})(2r_1 - n_{11}).$$

如图 2 所示, 数据拥有者Sen_i输入多个 SNP 位点的基因型数据及相关疾病的表现型后, 云服务器在密文域上对多个数据拥有者输入的数据进行卡方检验运算, 将结果返回给接收方 *REC*, *REC* 根据计算结果生成相关医学诊断, 在得到数据拥有者Sen_i的请求申请后, 将医学诊断发送给数据拥有者.

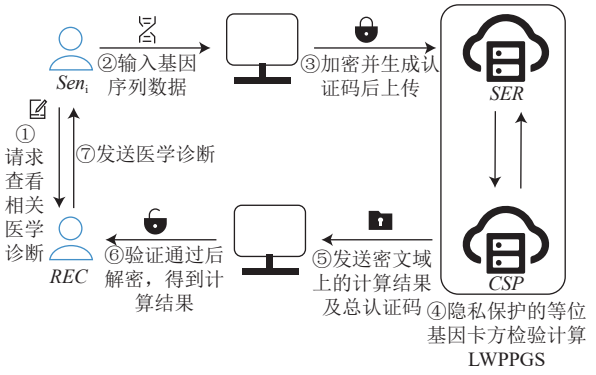


Fig. 2 Functional implementation architecture of LWPPAS

图 2 LWPPGS 功能实现架构

4.3.2 隐私保护的等位基因卡方检验方案(LWPPGS)

接下来我们在上述基因序列分析计算方法(等位基因卡方检验)的基础上, 基于可验证多密钥同态数据封装机制*VMK-HDEM*, 构造一个轻量级可验证多用户隐私保护等位基因卡方检验方案 LWPPGS. 该方案包含 5 个算法*LWPPGS.Setup*, *LWPPGS.KeyGen*, *LWPPGS.Enc*, *LWPPGS.Eval*和 *LWPPGS.Dec*, 我们假定共有n_u个用户: $\{U_i\}(i = 1,2,\cdots,n_u)$; 每个用户$U_i$有$M$个 SNP: $\{s_{i,j}\}(j = 1,2,\cdots,M)$; $\{y_{i,j_0}\}(j_0 = 1)$代表用户U_i的疾病状态; 令 5.3.1 节提出的可用于计算卡方值χ^2和关联强度OR的 4 个函数表达式为$R_j = (Item_{j,\mathrm{num}}, Item_{j,\mathrm{den}}, Item_{j,\mathrm{ornum}}, Item_{j,\mathrm{orden}})$.

本文提出的 LWPPGS 方案的符号描述如表 3 所示.

下面我们详细介绍 LWPPGS 方案的各个算法:

1) $LWPPGS.Setup(1^\lambda)$. 调用可验证多密钥同态数据封装机制中的$VMK\text{-}HDEM.Setup(1^\lambda)$算法完成初始化, 输入安全参数$\lambda$, 生成 3 对公私钥$(pk_{PKE,SER}, sk_{PKE,SER})$, $(pk_{PKE,CSP}, sk_{PKE,CSP})$和$(pk_{PKE,REC}, sk_{PKE,REC})$, 分别分发给云服务商*SER*, 加密服务提供商*CSP*, 授权接受用户*REC*. 并输出公开参数$PPR = (\lambda, pk_{PKE,SER}, pk_{PKE,CSP}, pk_{PKE,REC}, ID, T)$. ID是用户的身份标识符i的集合, T是标签τ的标签空间.

Table 3 Table for Describing LWPPGS Notations

表 3 LWPPGS 符号描述表

符号	描述
U_i	第i个用户
$s_{i,j}$	用户U_i的第j个 SNP 位点数值
n_{11}	携带主要等位基因的实验组病例数
r_0	对照组病例数
r_1	实验组病例数
c_1	主要等位基因的计数
N	总病例数

2) $LWPPGS.KeyGen(PPR)$. 调用可验证多密钥同态数据封装机制中的$VMK\text{-}HDEM.KeyGen(PPR)$算法, 以公共参数$PPR$为输入, 为每个用户$U_i$随机选择 2$L$+1 个大素数$p_{i,1}, p_{i,2}, \cdots, p_{i,L}, s_i, q_{i,1}, \cdots, q_{i,L}$, 计算$N_i = \prod_{l=1}^{L} p_{i,l}$, $T_i = s_i \prod_{l=1}^{L} p_{i,l}$, 设置用户$U_i$的临时公钥和临时私钥分别为$pbk_i = (T_i, \{q_{i,l}\})$, $pvk_i = (\{p_{i,l}\}, s_i, N_i)(l = 1,2,\cdots,L)$. *CSP* 也随机选择 2$L$+1 个大素数$p_1, p_2, \cdots, p_L, s, q_1, q_2, \cdots, q_L$, 计算$N = \prod_{l=1}^{L} p_l$, $T = s\prod_{l=1}^{L} p_l$, *CSP* 的临时公私钥为$pbk_{CSP} = T$, $pvk_{CSP} = (\{p_l\}, \{q_l\}, s, N)$. 用户$U_i$通过一个伪随机函数生成器生成伪随机函数私钥$k_{\mathrm{pse}}$, 并选择一个随机数$x_i$, 生成用于生成和验证认证码标签的密钥$k_i = (k_{\mathrm{pse}}, x_i)$, 并由$U_i$秘密保存.

3) $LWPPGS.Enc(PPR, pbk_i, pvk_i, \{s_{i,j}\}(i = 1,2,\cdots,n_u; j = 1,2,\cdots,M), \{y_{i,j_0}\}(i = 1,2,\cdots,n_u; j_0 = 1), k_i)$. 该算法输入公共参数、用户$U_i$的临时公私钥对、用户的疾病状态$y_{i,j_0}(j_0 = 1)$和$M$个 SNP 上的$s_{i,j}(j = 1,2,\cdots,M)$, 其中$y_{i,j_0} = \{0,1\}$; $s_{i,j} = \{0,1,2\}$. 然后调用算法$VMK\text{-}HDEM.Enc\left(PPR, pbk_i, pvk_i, \{s_{i,j}\}(i = 1,2,\cdots,n_u, j = 1,2,\cdots,M), k_i\right)$和算法$VMK\text{-}HDEM.Enc(PPR, pbk_i, pvk_i, \{y_{i,j_0}\}(i = 1,2,\cdots, n_u; j_0 = 1), k_i)$, 分别对用户$U_i$的对应 SNP 位点上的等位基因数$\{s_{i,j}\}$以及是否患病$\{y_{i,j_0}\}$进行加密, 得到加密后的数据$C_{s_i}$, C_{y_i}和认证码σ_{s_i}, σ_{y_i}再上传至服务器进行外包计算.

4) $LWPPGS.Eval(PPR, R_j, sk_{PKE,SER}, sk_{PKE,CSP}, pbk_{CSP}, pvk_{CSP}, C_{\mathrm{Sen}_i}(i = 1,2,\cdots,n_u))$. 该算法输入公共参数$PPR$、与卡方值$\chi^2$和关联强度$OR$有关的函数表达式$R_j$、计

算服务器 *SER* 和 *CSP* 的私钥、*CSP* 的临时公私钥对和用户上传的密文数据C_{Sen_i}，且其中

$$C_{Sen_i}=\left(C_{y_i},C_{s_i}\right);$$

$$R_j=\left(Item_{j,\mathrm{num}},Item_{j,\mathrm{den}},Item_{j,\mathrm{ornum}},Item_{j,\mathrm{orden}}\right).$$

然后*LWPPGS.Eval*算法调用*VMK-HDEM.Eval*$(PPR,R_j,sk_{PKE,SER},sk_{PKE,CSP},pbk_{CSP},pvk_{CSP},C_{Sen_i}(i=1,2,\cdots,n_u))$算法计算$R_j$中的函数表达式

$$C_{Item_{j,\mathrm{num}}}=d\times(n_{11}\times d-c_1\times 2r_1)^2,$$

$$C_{Item_{j,\mathrm{den}}}=c_1\times(d-c_1)\times 2r_1\times(d-2r_1),$$

$$C_{Item_{j,\mathrm{ornum}}}=n_{11}\times(n_{11}-2r_1-c_1+d),$$

$$C_{Item_{j,\mathrm{orden}}}=(c_1-n_{11})\times(2r_1-n_{11}),$$

其中，$\boldsymbol{u}_b=\{1\}_{i=1}^{b}\in\mathbb{R}^b;b=M,N;$

$$n_{11}:=\boldsymbol{y}^{\mathrm{T}}\boldsymbol{S};c_1:=\boldsymbol{u}_N^{\mathrm{T}}\boldsymbol{S}.$$

$\boldsymbol{y}^{\mathrm{T}}=(C_{y_{1,j_0}},C_{y_{2,j_0}},\cdots,C_{y_{n_u,j_0}})$是单位行向量，

$\boldsymbol{S}=(C_{s_{1,j}},C_{s_{2,j}},\cdots,C_{s_{n_u,j}})$是单位行向量.

计算完成后得到密文域上的n_u个用户在同一 SNP 位点的卡方检验值计算所需的所有分子分母值.

最后服务器将上述 L 组密文计算结果：

$$C_{R_j}=\left(C_{Item_{j,\mathrm{num}}},C_{Item_{j,\mathrm{den}}},C_{Item_{j,\mathrm{ornum}}},C_{Item_{j,\mathrm{orden}}}\right)(j=1,2,\cdots,L)$$

和上述计算过程中生成的验证码σ，以及其他密文发送给授权的接收方.

5) *LWPPGS.Dec*$(PPR,sk_{PKE,REC},C_{R_j},\sigma)$. 该算法输入公共参数*PPR*、授权接收者的私钥$sk_{PKE,REC}$和密文域上的推荐结果C_{R_j}，调用算法*VMK-HDEM.Dec*$(PPR,sk_{PKE,REC},C_{R_j},\sigma)$，用户利用认证标签密钥验证$\sigma$，若验证失败则拒绝接受*SER*的计算结果；若验证成功，则利用自己的私钥进行解密，获得明文域上卡方检验所需要的计算数据$R_j=\left(Item_{j,\mathrm{num}},Item_{j,\mathrm{den}},Item_{j,\mathrm{ornum}},Item_{j,\mathrm{orden}}\right),(j=1,2,\cdots,L)$. 最后计算得到$L$个卡方值$\chi_j^2$以及$L$个关联强度$OR$的值：

$$\chi_j^2=\frac{Item_{j,\mathrm{num}}}{Item_{j,\mathrm{den}}},$$

$$OR_j=\frac{Item_{j,\mathrm{ornum}}}{Item_{j,\mathrm{orden}}}(j=1,2,\cdots,L).$$

5 安全性证明

5.1 VMK-HDEM 方案的安全性证明

5.1.1 VMK-HDEM 方案的数据隐私安全性证明

本文提出的 VMK-HDEM 基于混合加密的思想设计实现. 具体而言，使用了 CPA 安全的任意公钥加密算法 *PKE* 加密对称密钥；利用仅包含简单加法、乘法运算的对称全同态加密算法加密数据. 依据文献 [38] 中有关混合加密安全性的定理 11 和定理 12，只需证明我们的对称全同态加密方案具有被动攻击下的不可区分安全性，便可证明混合加密方案，即本文构造的 VMK-HDEM 的数据隐私是 CPA 安全的.

在给出具体的安全性证明之前，我们先给出 2 个困难问题AGCD和D-AGCD的定义.

定义 4. 近似最大公约数问题(AGCD问题). (ρ,η,γ)−近似最大公约数问题是指：对于一个η比特的素数u，和一个γ比特的整数$M=u\cdot q_0$，其中q_0是在$\mathbb{Z}\cap[0,2^{\gamma}/u)$中的随机整数. 给出多项式个数的样本$D_{\gamma,\rho}(u):=$ {选择$q\leftarrow Z,r\leftarrow\mathbb{Z}\cap(0,2^{\rho})$,输出$x=q\cdot u+r(\mathrm{mod}\ M)$},求$u$.

定义 5. 判定性的近似最大公约数问题. (D-AGCD问题). (ρ,η,γ)−判定性的近似最大公约数问题是指：对于一个η比特的素数u和一个γ比特的整数$M=u\cdot q_0$，以及多项式数量样本$D_{\gamma,\rho}(u)$，判断$z=x+r'\cdot b(\mathrm{mod}\ M)$中的$b$，$b\in\{0,1\}$且$x\leftarrow D(u)$且$r'\leftarrow Z$.

下面，我们给出 VMK-HDEM 方案的数据隐私安全性定理.

定理 3. 数据隐私. 假设$\mathcal{A}_{\mathrm{dp}}^{\text{VMK-HDEM}}$是一个概率多项式时间的敌手，且能够以不可忽略的优势概率ε'攻破 VMK-HDEM 方案中对称全同态加密算法在被动攻击下的不可区分安全性. 那么存在一个模拟者B，可以通过与敌手$\mathcal{A}_{\mathrm{dp}}^{\text{VMK-HDEM}}$的交互，以不可忽略的优势概率$\epsilon_{\text{D-AGCD}}$解决 D-AGCD 困难问题，即

$$\varepsilon_{\text{D-AGCD}}\geqslant\varepsilon'.$$

证明. 我们首先给定一个诚实数据拥有者$Sen_{i'}$和其输入数据$\{m_{i',l}\}(i'\in ID;l=1,2,\cdots,L)$，$Sen_{i'}$上传它的密文输入数据给云服务器*SER*进行隐私保护的外包函数计算，再假定存在一个概率多项式时间能力的敌手$\mathcal{A}_{\mathrm{dp}}^{\text{VMK-HDEM}}$能以不可忽略的优势概率成功攻破 VMK-HDEM 方案中加密数据部分的对称全同态加密算法在被动攻击下的不可区分安全性，我们进而证明存在一个多项式时间的模拟者B能够通过下述实验$DP\text{-}CPA_{\mathcal{A}_{\mathrm{dp}}^{\text{VMK-HDEM}}}(\lambda)$利用$\mathcal{A}_{\mathrm{dp}}^{\text{VMK-HDEM}}$的能力解决 D-AGCD问题.

1)初始化阶段. 系统运行*VMK-HDEM.Setup*(1^{λ})算法，为*SER*, *CSP*和*REC*各生成一对公私钥$\left(pk_{PKE,SER},sk_{PKE,SER}\right),\left(pk_{PKE,CSP},sk_{PKE,CSP}\right),\left(pk_{PKE,REC},sk_{PKE,REC}\right)$. 并输出公开参数$PPR=(\lambda,pk_{PKE,SER},pk_{PKE,CSP},pk_{PKE,REC},ID,T)$. 然后系统运行*VMK-HDEM.KeyGen*(PPR)算法输出数据拥有者Sen_i的临时公私钥$pbk_i=\left(T_i,\{q_{i,l}\}\right),pvk_i=\left(\{p_{i,l}\},s_i,N_i\right)$和*CSP*的临时公私钥$pbk_{CSP}=T$，$pvk_{CSP}=$

$(\{p_l\},\{q_l\},s,N)$. 模拟者 B将 $(PPR,\{pbk_i\},pbk_{CSP})$返回给敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$.

2)查询腐化谕言机 O^{Cor}. 定义初始腐化集合 $L_{\text{corr}}=\varnothing$, 敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$可以俘获除诚实数据拥有者 $Sen_{i'}$以外的数据拥有者 $Sen_j(j=1,2,\cdots,n_s$且 $j\neq i')$、授权接收方以及云服务器 SER 和密码服务提供商 CSP 中的一个. 敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$可以利用 O^{Cor}查询 $O^{\text{Cor}}(j)$, O^{Cor}返回对应被腐化的数据拥有者的私钥 $pvk_j=\left(\{p_{j,l}\},s_j,N_j\right)$给敌手, 且模拟者 B更新 $L_{\text{corr}}\leftarrow L_{\text{corr}}\cup j$; $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$还可以查询 $O^{\text{Cor}}(SER)$, O^{Cor}返回 $sk_{PKE,SER}$; $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$查询 $O^{\text{Cor}}(CSP)$, O^{Cor}返回 $(sk_{PKE,CSP},pvk_{CSP})$, 其中 $pvk_{CSP}=(\{p_l\},\{q_l\},s,N)$; $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$查询 $O^{\text{Cor}}(REC)$, O^{Cor}返回 $sk_{PKE,REC}$. 且模拟者 B相应地更新腐化集合.

3)挑战阶段. 敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$选择一对挑战明文 $(\{m_{i',l,0}\},\{m_{i',l,1}\})(l=1,2,\cdots,L)$, 满足 $\left|m_{i',l,0}\right|=\left|m_{i',l,1}\right|$, 将它们发送给模拟者 B.B随机选取 $b\in_R\{0,1\}$然后计算

$$C_{i'}=r_{i',0}\left(\sum_{l=1}^{L}M_{i',l}^{-1}M_{i',l}\left(m_{i',l,b}+k_{i',l}q_{i',l}\right)_{p_{i',l}}\right)+r_{i',1}N_{i'}\bmod T_{i'}=q_{i'}\cdot u_{i',h}+\left(m_{i',h,b}+k_{i',h}q_{i',h}\right)_{p_{i',h}}\bmod T_{i'},$$

其中 $r_{i',0},r_{i',1}\in\{0,1\}^{2\lambda}$, $k_{i',l}\in\{0,1\}^{\lambda}\subseteq\mathbb{Z}\cap(0,2^{\rho})(\rho=\lambda)$, $N_{i'}=p_{i',1}\cdot M_{i',1},M_{i',l}^{-1}M_{i',l}\equiv1(\bmod p_{i',l})$,且 $q_{i'}\in\mathbb{Z}_{q_{0,i'}},q_{0,i'}=T_{i'}/u_{i',h}$.

模拟者 B返回 $C_{i'}^{*}\leftarrow C_{i'}+r''b\bmod T_{i'}$作为挑战密文返回给敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$, 其中 $r''\in_R\mathbb{Z}_{T_{i_0}}$. 敌手输出 $b'=\{0,1\}$. 如果 $b'=b$, 则敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$在游戏 $DP\text{-}CPA_{\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}}(\lambda)$ 中获胜, 说明该方案的对称全同态加密算法部分在被动攻击下不具备不可区分性安全; 反之敌手失败, 则说明该方案的对称全同态加密算法部分在被动攻击下具备不可区分性安全.

从上述的模拟规约过程中我们可以观察到: 当 $b=1$并且 r''是在 $\mathbb{Z}_{T_i}$中服从均匀分布随机选取时, $C_{i'}^{*}$在 $\mathbb{Z}_{T_i}$上服从均匀分布. 因此在这种情况下, 模拟者 B解决困难问题 D-AGCD的优势概率是 0, 因为 $C_{i'}^{*}$没有泄露任何 $m_{i',l,b}$的任何信息, 由此敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$猜对 b的概率是 1/2. 相反地, 当 $b=0$时, $C_{i'}^{*}=C_{i'}$, 且 $C_{i'}^{*}\leftarrow D_{\gamma,\lambda}\left(u_{i,h}\right)$, 因此 $C_{i'}^{*}$与 $m_{i',l,b}$的密文是不可区分的.

因此, 给定模拟者 D-AGCD困难问题实例, $z=x+r'\cdot b(\bmod M)$, 判断 $b\in\{0,1\}$. 模拟者 B只需要令 $C_{i'}^{*}=z$, $r''=r',C_{i'}=x$; 将 $C_{i'}^{*}$作为挑战密文返回给敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$. 若敌手 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$能够攻破 VMK-HDEM 方案中对称全同态加密算法在被动攻击下的不可区分安全性, 则 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$输出的 $b'=b$, 模拟者 B以 $\mathcal{A}_{\text{dp}}^{\text{VMK-HDEM}}$的输出 b'作为对 D-AGCD困难问题的解, 即可得

$$\varepsilon_{\text{D-AGCD}}\geqslant\varepsilon'$$

证毕.

5.1.2 VMK-HDEM 方案的可验证性安全性证明

定理 4. 可验证性安全. 如果 5.2 节中 *VMK-HDEM.KeyGen* 算法中定义的函数 $F_{k_{\text{pse}}}$是一个伪随机函数, 那么本文所构造 VMK-HDEM 方案的可验证性安全(不可伪造性)可抵抗在 4.3.2 节中定义的概率多项式时间能力的敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$发起的适应性选择消息攻击.

证明. 由 Fiore[29] 等人给出的 Theorem3可知: 如果函数 F是一个伪随机函数, 那么其在文献 [29] 中给出的多密钥同态消息认证码对于概率多项式实践能力的敌手发起的适应性选择消息攻击具有可验证性安全(不可伪造性). 而在本文构造的 VMK-HDEM方案中, 将文献 [29] 给出的多密钥同态消息认证码技术在对应的密文域上实现, 使得所构造的 VMK-HDEM 方案在恶意敌手模型下保护数据隐私的同时可实现对云服务器密文计算结果的正确性验证. 由此, 定理 4 可由文献 [29] 的 Theorem3中的结论直接得证. 证毕.

5.2 LWPPGS 方案的安全性

1)LWPPGS 方案的数据隐私安全. 由于 LWPPGS 方案的数据隐私安全目标是基于 VMK-HDEM 方案的数据隐私安全模型构建的, 因此当 VMK-HDEM 方案的数据隐私能实现 CPA 安全时, LWPPGS 方案的数据隐私也能实现 CPA 安全.

2)LWPPGS 方案的可验证性安全. 由于 LWPPGS 方案的可验证性安全目标基于 VMK-HDEM 方案的可验证性安全模型构建的, 因此当 VMK-HDEM 方案的可验证性安全可抵抗 4.3.2 节中定义的概率多项式时间能力的敌手 $\mathcal{A}_{\text{ver}}^{\text{VMK-HDEM}}$发起的适应性选择消息攻击时, LWPPGS 方案也可获得相应的可验证性安全.

6 性能分析

本节对 VMK-HDEM 与基于公钥全同态加密 Brakerski's FHE[20] 的 López-Alt 多用户安全外包计算方案[2], 在计算开销和通信开销等方面进行了对比分析, 表明了以 VMK-HDEM 作为密码原语构造的多用户可验证隐私保护基因序列分析方案的高效性与实用性.

我们使用 MIRACL 库[39], 在搭载 2.6 GHz 处理器和 4 GB RAM 的 Linux 平台上运行以研究方案性能. 我们设置安全参数 $\lambda=128$. 在 VMK-HDEM方案和轻量级高效可验证隐私保护基因序列分析方案中, 我们使用消息空间在 $\{0,1\}^{2\lambda}$上具有 CPA 安全的 ElGamal 来实例化 *PKE*. 在对比方案 López-Alt 方案[2] 使用的

Brakerski 的公钥全同态加密算法[3]中，我们设置$|q_B|=512,|p_B|=4\,096$.

我们首先统计了 López-Alt 方案[2]和 VMK-HDEM 方案分别在数据拥有者端SEN、云服务器SER端和接收方REC端的关于数据隐私保护和可验证性的计算开销和通信开销. 表 4 和表 5 分别是 López-Alt 方案[2]和 VMK-HDEM 方案的计算开销对比表和通信开销对比表.

Table 4 Comparison of Computation Cost of López-Alt and VMK-HDEM Schemes

表 4 López-Alt 和 VMK-HDEM 方案的计算开销对比

方案	用于数据隐私保护的计算开销			用于可验证性的计算开销		
	SEN	*SER*	*REC*	*SEN*	*SER*	*REC*
López-Alt	$O(n_i\cdot L)$	$O\left(n_s+K2^{deg_F}\right)$	$O(2^{deg_F})$	$O(L)$	$O(L\cdot Kdeg_F)$	$O\left(L\cdot Kdeg_F\right)$
VMK-HDEM	$O(L)$	$O\left(n_s+Kdeg_F\right)$	$O(1)$	$O(1)$	$O\left(Kdeg_F\right)$	$O\left(Kdeg_F\right)$

注：n_s表示来自n_s个用户的输入数据数，K和deg_F分别表示多元多项式$F(x_1,x_2,\cdots,x_{n_s})$的项数和阶.

Table 5 Comparison of Communication Cost of López-Alt and VMK-HDEM Schemes

表 5 López-Alt 和 VMK-HDEM 方案的通信开销对比

方案	用于数据隐私保护的通信开销			用于可验证性的通信开销		
	SEN	*SER*	*REC*	*SEN*	*SER*	*REC*
López-Alt	$O(L)$	$O\left((n_s+2^{deg_F})\cdot L\right)$	$O\left(2^{deg_F}\right)$	$O(L)$	$O(deg_F\cdot L)$	$O(deg_F\cdot L)$
VMK-HDEM	$O(L)$	$O(n_s\cdot L)$	$O(1)$	$O(1)$	$O(deg_F)$	$O(deg_F)$

表 4 和表 5 中L是函数输入实例个数，n_s是数据拥有者个数，n_i是数据拥有者Sen_i的数据集大小，deg_F是多项式F的次数. 下面我们具体分析 López-Alt 方案[2]和 VMK-HDEM 方案中不同参与方的开销.

6.1 计算开销

6.1.1 数据隐私保护的计算开销

假定每个数据拥有者$Sen_i(i=1,2,\cdots,n_s)$的数据集大小为n_i，最好的情况是在每个外包函数计算实例中，Sen_i只需提供其数据集中的 1 个输入数据，且Sen_i可提供L个输入数据作为L次打包函数计算的输入. 而最坏情况下，每一次函数运算都需要Sen_i输入其数据集中的全部数据作为外包函数计算的输入并批量打包进行L次函数计算，此时本文 VMK-HDEM 方案中数据拥有者的公钥加密算法 *PKE* 使用次数复杂度为$O(L)$；而 López-Alt 的方案[2]的公钥加密使用次数复杂度是$O(n_i\cdot L)$. 这是因为本文的 VMK-HDEM 方案基于混合加密，数据拥有者可在离线阶段用公钥加密算法加密临时对称密钥，用全同态对称加密算法加密数据，因此公钥加密的使用次数复杂度与数据拥有者Sen_i的数据集大小n_i无关；而 López-Alt 方案[2]中，每一次函数运算中数据拥有者需要对其数据集中的每一个数据进行 Brakerski's 公钥全同态加密，因此其公钥加密的使用次数复杂度为$O(n_i\cdot L)$. 显见，VMK-HDEM 方案在数据拥有者端的计算复杂度显著降低. 如图 3 所示，横坐标是数据集的大小n_i，纵坐标是数据拥有者的公钥加密计算开销，在最坏情况下，随着数据集大小n_i的增长，VMK-HDEM 方案中Sen_i的公钥加密的计算开销基本保持不变，而 López-Alt 方案[2]中Sen_i的公钥加密计算开销迅速增加.

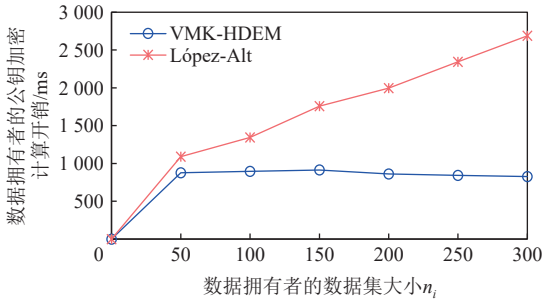


Fig. 3 Analysis of public key computational cost of López-Alt and VMK-HDEM schemes at Sen_i

图 3 López-Alt 与 VMK-HDEM 方案在Sen_i的公钥计算开销分析

在 VMK-HDEM 方案中，云服务器 *SER* 的计算开销为$O\left(n_s+Kdeg_F\right)$，远小于 López-Alt 方案[2]中的$O\left(n_s+K2^{deg_F}\right)$.因为 VMK-HDEM 方案中，云服务器 *SER* 为了在密文域上计算外包函数，除了运行n_s次乘法盲化外，只需要进行$K\cdot(deg_F-1)$次乘法运算；而 López-Alt 的方案[2]在最坏的情况下，即多项式F的每个乘积项$Item_j(j=1,2,\cdots,K)$的次数都为deg_F，需要n_s次乘法盲化和$K\cdot2^{deg_F}$次乘法运算来完成密文域上函数计算. 如图 4 所示，横坐标是Sen_i加密的数据实例个数L，纵坐标是云服务器的计算开销，当多元多项式的项数K和阶deg_F增大时，VMK-HDEM 方案中云服务器 *SER* 的计算开销的增长幅度明显小于 López-Alt 方案[2]中的增长幅度.

在 VMK-HDEM 方案和 López-Alt 的方案[2]中，密码服务器提供商 *CSP* 的计算开销都是$O(n_sL)$(此处n_s代表外包计算函数F的变元个数)，用于在函数计算时对F的n_s个输入数据进行重加密. 而对于接收方 *REC*，由于 López-Alt 的方案中使用 Brakerski 的公钥 FHE[20]，进行解密需要对大小为2^{deg_F}的密文向量和私钥向量进行内积运算；而在 VMK-HDEM 中，接收方 *REC* 的计算开销与外包函数F的次数无关，其计算复

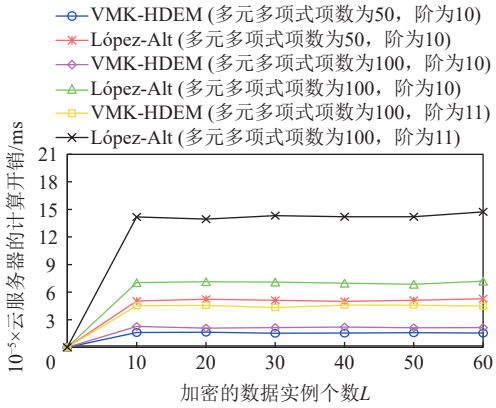


Fig. 4 Analysis of computational cost of López-Alt and VMK-HDEM schemes at *SER*

图 4 López-Alt 与 VMK-HDEM 方案在 *SER* 的计算开销分析

杂度为$O(1)$, 大大低于 López-Alt 中所需的$O\left(2^{deg_F}\right)$. 如图 5 所示, 横坐标是Sen_i加密的数据实例个数L, 纵坐标是 *CSP* 和 *REC* 的计算开销, 随着加密的数据实例个数 L 的增长, VMK-HDEM 方案中 *REC* 的计算开销明显少于 López-Alt 方案中的计算开销; VMK-HDEM 方案中 *CSP* 的计算开销也略少于 López-Alt 方案.

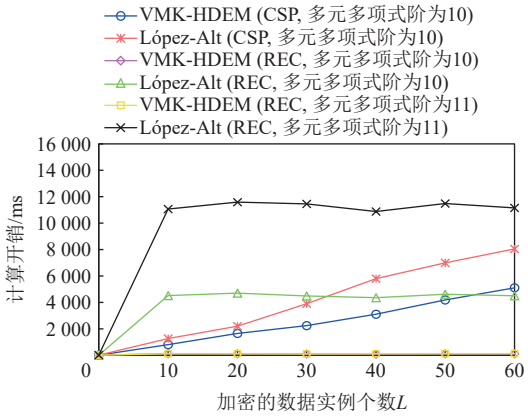


Fig. 5 Analysis of computational cost of López-Alt and VMK-HDEM schemes at *CSP* and *REC*

图 5 López-Alt 与 VMK-HDEM 方案在 *CSP* 和 *REC* 的计算开销分析

6.1.2 可验证性的计算复杂度分析

在 VMK-HDEM 方案中, 数据拥有者Sen_i可以对其在L个函数输入实例中的消息$m_{i,l}(l=1,2,\cdots,L)$进行打包生成一个打包后的密文C_i, 然后在密文C_i上生成一个打包认证码σ_i, 因此计算开销与其函数的输入实例个数L无关, 计算复杂度是$O(1)$; 而在 López-Alt 的方案[2] 中, 因无法对$m_{i,l}$进行打包, 所以需要生成L个认证码$\sigma_{i,l}(l=1,\cdots,L)$, 计算复杂度应为$O(L)$. 如图 6 所示, 横坐标是$Sen_i$加密的数据实例的个数$L$, 纵坐标为$Sen_i$在可验证性上的计算开销, 随着实例个数的增加, VMK-HDEM 方案的数据拥有者的可验证性的计算开销基本不变, 而 López-Alt 方案中的开销在不断增加.

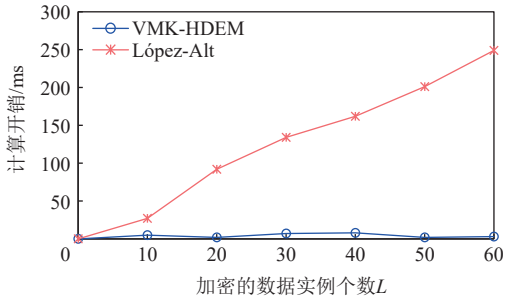


Fig. 6 Analysis of computational cost for verifiability of López-Alt and VMK-HDEM schemes at Sen_i

图 6 López-Alt 与 VMK-HDEM 方案可验证性在Sen_i的计算开销分析

对于云服务器*SER*而言, 在 VMK-HDEM 方案中Sen_i会对其L个输入实例进行打包生成一个验证码$\sigma_i(i=1,2,\cdots,n_s)$, 因此*SER*只需要将$\sigma_i$代入外包计算函数$F$中进行一次函数计算, 生成一个计算结果正确性的证据σ, 计算复杂度是$O\left(Kdeg_F\right)$; 而在 López-Alt 的方案[2] 中, 因无法对$m_{i,l}$进行打包, 所以计算复杂度应为$O(L\cdot Kdeg_F)$. 如图 7 所示, 横坐标是Sen_i加密的数据实例个数L, 纵坐标是*SER*的可验证性的计算开销, 随着实例个数的增加, VMK-HDEM 方案中*SER*的可验证性计算开销基本不变, 而 López-Alt 的方案[2] 中开销在不断增加.

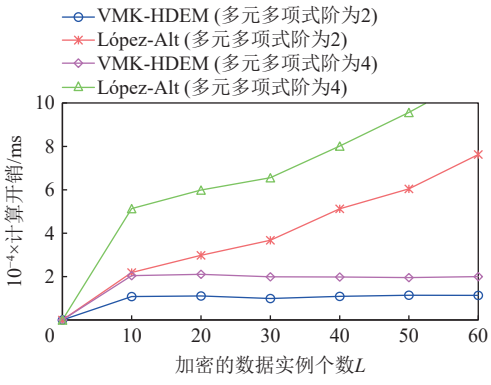


Fig. 7 Analysis of computational cost for verifiability of López-Alt and VMK-HDEM schemes at *SER*

图 7 López-Alt 与 VMK-HDEM 方案可验证性在 *SER* 的计算开销分析

对于接收方*REC*而言, 由于 VMK-HDEM 方案中, 云服务器*SER*对L个输入实例的函数计算结果批量生成一个唯一的用于计算结果正确性验证的证据σ,

因此*REC*只需对此证据进行1次验证就能表明L个函数计算结果的正确性，计算复杂度是$O\left(Kdeg_F\right)$；相对地，López-Alt方案中接收方*REC*的计算复杂度是$O\left(L\cdot Kdeg_F\right)$.

如图8所示，横坐标是Sen_i加密的数据实例个数L，纵坐标是*REC*的可验证性的计算开销，随着实例个数的增加，VMK-HDEM方案中*REC*的可验证性计算开销基本不变，而López-Alt的方案[2]中验证的计算开销基本呈线性增长.

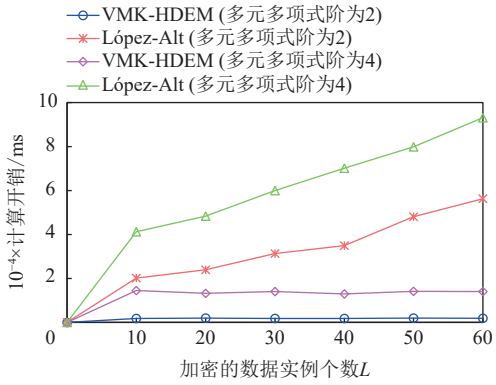


Fig. 8 Analysis of computational cost for verifiability of López-Alt and VMK-HDEM schemes at *REC*

图8 López-Alt与VMK-HDEM方案可验证性在*REC*的计算开销分析

6.2 通信复杂度分析

6.2.1 数据相关的通信复杂度

在本文提出的VMK-HDEM方案中，当$Sen_i(i=1,2,\cdots,n_s)$加密的输入数据实例个数为L时，云服务器*SER*和接收方*REC*的通信复杂度分别为$O(n_s\cdot L)$和$O(1)$；但在基于Brakerski公钥FHE[20]的López-Alt的方案[2]中，由于云服务器每在密文域上执行1次乘法运算，其密文大小就将扩张1倍，所以其云服务器*SER*和接收方*REC*的通信复杂度分别为$O\left((n_s+2^{deg_F})\cdot L\right)$和$O\left(2^{deg_F}\right)$. 如图9所示，横坐标为多元多项式的阶数$deg_F$，纵坐标为云服务器*SER*和接收方*REC*的通信开销. 随着多元多项式阶数的增加，VMK-HDEM方案中*SER*和*REC*的通信开销基本不变，而López-Alt的方案[2]中的通信开销随之迅速增加.

当$Sen_i(i=1,\cdots,n_s)$加密的输入数据实例个数为L时，在VMK-HDEM方案中，由*VMK-HDEM.KeyGen*算法可知，数据拥有者Sen_i的通信开销复杂度是$O(L)$；而López-Alt的方案[2]中Sen_i的通信开销也是$O(L)$. 但在VMK-HDEM方案中，数据拥有者仅需向云服务器上传$L/2$个加密对称密钥的公钥加密算法*PKE*的密文；而在López-Alt的方案[2]中，数据拥有者需要上传

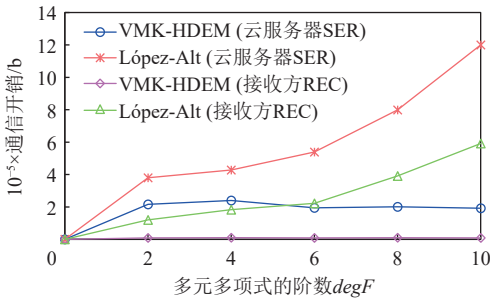


Fig. 9 Analysis of communication cost of *REC* and *SER* in López-Alt and VMK-HDEM schemes

图9 López-Alt与VMK-HDEM方案在*REC*和*SER*的通信开销分析

L个输入数据的公钥全同态加密密文. 如图10所示，横坐标是Sen_i的加密的数据实例个数L，纵坐标是数据拥有者的通信开销，VMK-HDEM方案中数据拥有者的通信开销小于López-Alt的方案中的开销.

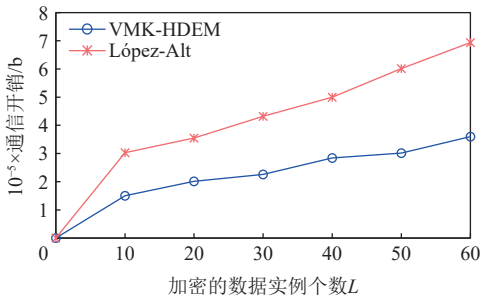


Fig. 10 Analysis of communication cost of López-Alt and VMK-HDEM schemes at *SEN*

图10 López-Alt与VMK-HDEM方案在*SEN*的通信开销分析

6.2.2 可验证性相关的通信复杂度

在VMK-HDEM方案中，假设数据拥有者Sen_i有L个输入实例$m_{i,l}(l=1,2,\cdots,L)$，由7.1.2小节中的分析可知，VMK-HDEM方案中Sen_i只生成1个验证码σ_i，而López-Alt的方案[2]需要生成L个认证码. 因此VMK-HDEM方案中，数据拥有者的有关验证性的通信复杂度是$O(1)$，López-Alt的方案是$O(L)$. 如图11所示，横坐标是数据拥有者加密的数据实例个数L，纵坐标是数据拥有者的通信开销，随着实例个数的增加，VMK-HDEM方案中数据拥有者可验证性的通信开销基本不变，而López-Alt的方案中的开销随之增加.

由7.1.2节的分析可知，在VMK-HDEM方案中，云服务器*SER*会对L个计算结果正确性的证据进行打包生成一个证据σ，并将其发送给接收方*REC*；而López-Alt的方案[2]中，云服务器会计算生成L个计算结果正确性的证据并将其发送给接收方*REC*. 因此

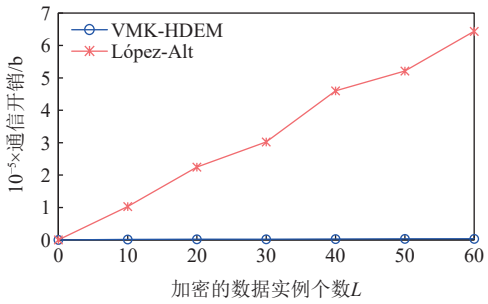


Fig. 11 Analysis of communication cost for verifiability of López-Alt and VMK-HDEM schemes at *SEN*

图 11 López-Alt 与 VMK-HDEM 方案在 *SEN* 的可验证性的通信开销分析

VMK-HDEM 方案中云服务器和接收方有关验证性的通信复杂度都为 $O(deg_F)$; 而 López-Alt 的方案中云服务器和接受方有关验证性的通信复杂度都为 $O(deg_F \cdot L)$. 如图 12 所示, 横坐标为 Sen_i 加密的数据实例个数 L, 纵坐标为云服务器和接收方的可验证性的通信开销. 当 $deg_F = 10$ 时, 随着实例个数的增加, VMK-HDEM 方案中接收方 *REC* 和云服务器 *SER* 的可验证性通信开销基本不变, 而 López-Alt 的方案[2] 中的开销随之增加.

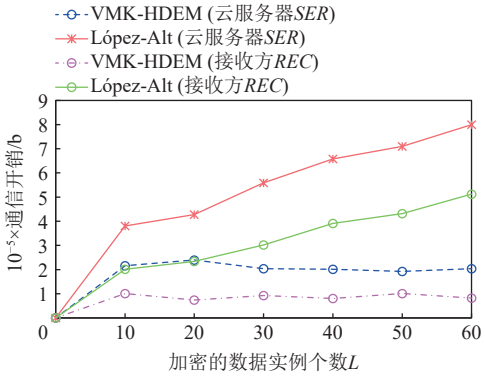


Fig. 12 Analysis of communication cost for verifiability of López-Alt and VMK-HDEM schemes at *SER* and *REC*

图 12 López-Alt 与 VMK-HDEM 方案在 *SER* 和 *REC* 的可验证性的通信开销分析

7 结 论

本文提出了一种高效可验证的多密钥同态数据封装机制 VMK-HDEM, 并基于该新型密码原语设计了一个轻量级的多用户可验证隐私保护基因序列分析协议 LWPPGS. 通过减少公钥加密使用次数, 对数据拥有者的数据进行批量加密; 支持对多个函数输入实例在密文域上进行批量函数计算和批量生成计算结果正确性验证证据. 最后, 安全性分析和实验仿真表明在保护用户基因数据隐私和基因序列分析结果隐私的前提下, 所构造的方案能准确、高效地完成基因序列分析并验证其正确性.

作者贡献声明: 胡云舒提出可验证的多密钥同态数据封装机制 VMK-HDEM 的算法思路和实验方案并撰写论文; 周俊提出可验证隐私保护基因序列分析方案 LWPPGS 的算法思路和实验方案; 曹珍富对本文的密码方案给出总体设计思路和框架; 董晓蕾提出指导意见并修改论文.

参 考 文 献

[1] Kantarcioglu M, Jiang W, Liu Y, et al. A cryptographic approach to securely share and uery genomic sequences[C]//Proc of the 2008 IEEE Transactions Information Technol. Piscataway, NJ: IEEE, 2008: 606–617

[2] López-Alt A, Tromer E, Vaikuntanathan V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption[C]//Proc of the 44th annual ACM Symp on Theory of Computing. New York: ACM, 2012: 1219–1234

[3] Brakerski Z, Vaikuntanathan V. Fully homomorphic encryption from ring-LWE and security for key dependent messages[C]//Advances in Cryptology—CRYPTO 2011(Lecture Notes in Computer Science). Berlin: Springer, 2011: 505–524

[4] Chen H, Dai W, Kim M, et al. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference[C]//Proc of the 2019 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2019: 395–412

[5] Cheno J H, Kim A, Kim M, et al. Homomorphic encryption for arithmetic of approximate numbers[C]//Advances in Cryptology – ASIACRYPT 2017. Cham: Springer, 2017: 409–437

[6] Coron J S, Lepoint J, Tibouchi M. ScaleInvariant fully homomorphic encryption over the integers[C]//Advances in Cryptology-EUROCRYPT 2010. Berlin: Springer, 2014: 311–328

[7] Cheon J H, Kim J, Lee M S, et al, CRT-based fully homomorphic encryption over the integers[J]. Information Sciences, 2015, 310: 149−162

[8] Micciancio D, Vaikuntanathan V. SoK: Learning with errors, circular security, and fully homomorphic encryption[C]//Proc of Public-Key Cryptography – PKC 2024. Cham: Springer Nature Switzerland, 2024: 291−321

[9] Zhou Jun, Kim-Kwang R C, Cao Zhenfu, et al. PVOPM: Verifiable privacy-preserving pattern matching with efficient outsourcing in the malicious setting[C]//Proc of the 2021 IEEE Transactions Dependable Secure Computing. Piscataway, NJ: IEEE, 2021: 2253−2270

[10] Cao Zhenfu, Dong Xiaolei, Zhou Jun, et al. Research advances on big data security and privacy preserving[J]. Journal of Computer

Research and Development, 2016, 53(10): 2137−2151 (in Chinese)
(曹珍富, 董晓蕾, 周俊, 等. 大数据安全与隐私保护研究进展[J]. 计算机研究与发展, 2016, 53(10): 2137−2151)

[11] Chen Jing, Chen Zhiping, Kuang Linai, et al. Security count query and integrity verification based on encrypted genomic data[C]//Proc of the 9th Int Conf on Computer Engineering and Networks. Berlin: Springer, 2021: 647−654

[12] Hasan M Z, Mahdi M S R, Sadat M N, et al. Secure count query on encrypted genomic data[J]. Journal of Biomedical Informatics, 2018, 81: 41−52

[13] Choi S G, Katz J, Kumaresan R, et al. Multi-client non-interactive verifiable computation[C]//Proc of Theory of Cryptography. Berlin: Springer, 2013: 499−518

[14] Feng Shuai, Deng Lunzhi. Identity-based data auditing scheme with privacy protection[J]. Journal of Guizhou Normal University (Natural Sciences), 2023, 41(2): 105–112
(冯帅, 邓伦治. 基于身份的隐私保护数据审计方案[J]. 贵州师范大学学报: 自然科学版, 2023, 41(2): 105–112)

[15] Rivest R L, Adleman L, Dertouzos M L. On data banks and privacy homomorphisms[C]. Foundations of Secure Computation, New York: Academia Press, 1978: 169−179

[16] Kwak H, Min S, Song Y. Towards practical multi-key TFHE: Parallelizable, key-compatible, quasi-linear complexity[C]//Proc of Public-Key Cryptography – PKC 2024. Cham: Springer, 2024: 354−385

[17] Peng Changgen, He Xin, Tan Weijie, et al. Research status and countermeasures of artificial intelligence algorithm security[J]. Journal of Guizhou Normal University (Natural Sciences), 2022, 40(6): 1−16
(彭长根, 何兴, 谭伟杰等. 人工智能算法安全研究现状与对策[J]. 贵州师范大学学报: 自然科学版, 2022, 40(6): 1−16)

[18] Gentry C. Fully homomorphic encryption using ideal lattices[C]// Proc of the 41st Annual ACM Symp on Theory of Computing. New York: ACM, 2009: 169−178

[19] Dijk M V, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers[J]. Lecture Notes in Computer Science, 2010, 2009(4): 24−43

[20] Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE[C]//Proc of IEEE 52nd Annual Symp on Foundations of Computer Science (FOCS2011). Piscataway, NJ: IEEE, 2011: 97−106

[21] Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) Fully homomorphic encryption without bootstrapping[C]//Proc of the 3rd Innovations in Theoretical Computer Science Conf (ITCS'12). New York: ACM, 2012: 309–325

[22] Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]//Advances in Cryptology – CRYPTO 2013. Berlin: Springer, 2013: 75−92

[23] Lee Y, Micciancio D, Kim A, et al. Efficient FHEW bootstrapping with small evaluation keys, and applications to threshold homomorphic encryption[C]//Proc of EUROCRYPT 2023. Berlin: Springer, 2023: 227−256

[24] Liu Fenghao, Wang Han. Batch bootstrapping I: - A new framework for SIMD bootstrapping in polynomial modulus[C]//Proc of EUROCRYPT 2023. Berlin: Springer, 2023: 321−352

[25] Zhou Jun, Cao Zhenfu, Qin Zhan, et al. LPPA: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in VANETs[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 420−434

[26] Gennaro R, Gentry C, Parno B. Non-interactive verifiable computing: Outsourcing computation to untrusted workers[C]//Advances in Cryptology – CRYPTO 2010. Berlin: Springer, 2010: 465−482

[27] Chung K M, Kalai Y, and Vadhan S. Improved delegation of computation using fully homomorphic encryption[C]//Advances in Cryptology – CRYPTO 2010. Berlin: Springer, 2010: 483−501

[28] Parno B, Raykova M, Vaikuntanathan V. How to delegate and verify in public: Verifiable computation from attribute-based encryption[C]// Proc of Theory of Cryptography. Berlin: Springer, 2012: 422–439

[29] Papamanthou C, Shi E, Tamassia R. Signatures of correct computation[C]//Proc of Theory of Cryptography. Berlin: Springer, 2013: 222–242

[30] Choi S G, Katz J, Kumaresan R. et al. Multi-client noninteractive verifiable computation[C]//Proc of Theory of Cryptography. Berlin: Springer, 2013: 499–518

[31] Baum C, Orsini E, Scholl P, et al. Efficient constant-round MPC with identifiable abort and public verifiability[C]//Advances in Cryptology – CRYPTO 2020. Berlin: Springer, 2020: 562−592

[32] Zhang Liangfeng, Wang Huaxiong. Multi-server verifiable computation of low-degree polynomials[C]//Proc of IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2022: 596−613

[33] Chen Xin, Zhang Liangfeng, Publicly Verifiable homomorphic secret sharing for polynomial evaluation[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 4609−4624

[34] Gordon S D, Katz J, Liu F H, et al. Multi-client verifiable computation with stronger security guarantees[C]//Proc of Theory of Cryptography. Berlin: Springer, 2015: 144–168

[35] Manulis M, Nguyen J. Fully homomorphic encryption beyond IND-CCA1 security: Integrity through verifiability[C]//Advances in Cryptology-EUROCRYPT 2024. Cham: Springer, 2024: 63−93

[36] Jing Liu, Liang Fengzhang. Privay-preserving and publicly verifiable matrix multiplication[J]. IEEE Transaction on Service Computing, 2013, 16(3): 2059−2071

[37] Fiore D, Mitrokotsa A, Nizzardo L, et al. Multikey homomorphic authenticators[C]//Advances in Cryptology – ASIACRYPT 2016. Berlin: Springer, 2016: 499−530

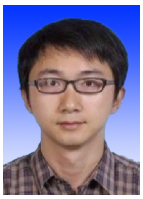
[38] Katz J, Lindell Y. Introduction to Modern Cryptography, Second Edition[M]. Boca Raton FL, USA: CRC Press, 2014

[39] Multiprecision integer and rational arithmetic c/c++ library[OL]. http://www.shamus.ie/



Hu Yunshu, born in 2004. Undergraduate. Her main research interests include fully homomorphic encryption, secure multiparty computation, and verifiable computation.

胡云舒, 2004 年生. 本科生. 主要研究方向为全同态加密、安全多方计算、可验证计算.



Zhou Jun, born in 1982. PhD, professor, PhD supervisor. Member of CCF, IEEE, ACM and IACR. His main research interests include public key cryptography, cloud computing security, and privacy-preserving machine learning.

周　俊, 1982 年生. 博士，教授，博士生导师. CCF、IEEE、ACM 和 IACR 会员. 主要研究方向为公钥密码学、云计算安全、机器学习隐私保护.



Cao Zhenfu, born in 1962. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include public key encryption, digital signature, private computation, and post-quantum cryptography.

曹珍富, 1962 年生. 博士，教授，博士生导师. CCF 高级会员. 主要研究方向为公钥加密、数字签名、隐私计算、后量子密码.



Dong Xiaolei, born in 1971. PhD, professor, PhD supervisor. Her main research interests include number theory, cryptography, and network security.

董晓蕾, 1971 年生. 博士，教授，博士生导师. 主要研究方向为数论、密码学、网络安全.