基于分层混合架构的可验证密态计算方法研究

杜涛，邓劲生*，陈旭灿
（军事科学院，北京 100080）

摘要：针对密态计算中安全与效率权衡僵化、验证机制与密码学原语紧耦合、跨范式互操作缺失的核心挑战，提出一种分层混合式可验证密态计算（layered hybrid verifiable secure computation，LH-VSC）模型。该模型采用纵向分层、横向混合架构。纵向通过数据接入、密态计算、可验证计算、信任管理、结果输出 5 层实现功能解耦；横向由动态调度器基于任务属性与系统状态智能切换全同态加密与安全多方计算路径，实现安全与效率按需优化。本文设计轻量级可验证安全多方计算协议，基于信息论安全消息认证码与聚合证明将验证复杂度降至 $O(1)$；本文构建全同态加密（fully homomorphic encryption，FHE）计算轨迹精简证明机制，通过语义感知分解与递归证明聚合，将零知识简洁非交互式知识论证（zero-knowledge succinct non-interactive argument of knowledge，zkSNARK）证明生成开销从 $O(M\cdot D)$ 降至 $O(M+D)$（M 为子步骤数，D 为最大子电路深度）。该机制实现了安全性、效率与灵活性的协同优化，为隐私保护计算基础设施提供了新范式。

关键词：密态计算；分层混合架构；可验证计算；动态调度；全同态加密；安全多方计算
中图分类号：TP393 **文献标志码：**A
DOI: 10.20172/j.issn.2097-3136.260608

Research on verifiable encrypted computing method based on layered hybrid architecture

Du tao, Deng Jinsheng*, Chen Xucan
（Academy of military science, Beijing 100080, China）

Abstract: A layered hybrid verifiable secure computation model (LH-VSC) was proposed to address the core challenges in encrypted state computing, including the rigid trade-off between security and efficiency, the tight coupling of verification mechanisms with cryptographic primitives, and the lack of cross-paradigm interoperability. This model adopts a vertically layered and horizontally hybrid architecture. Vertically, it realizes functional decoupling via five layers: data access, encrypted computation, verifiable computation, trust management, and result output. Horizontally, a dynamic scheduler intelligently switches between fully homomorphic encryption (FHE) and secure multi-party computation (MPC) paths based on task attributes and system state, enabling on-demand optimization of security and efficiency. A lightweight verifiable secure multi-party computation protocol was designed, leveraging information-theoretically secure message authentication codes and aggregated proofs to reduce verification complexity to $O(1)$. A concise proof mechanism for FHE computation traces was constructed, combining semantic-aware decomposition and recursive proof aggregation to reduce the zero-knowledge succinct non-interactive argument of knowledge (zkSNARK) proof generation overhead from $O(M\cdot D)$ to $O(M+D)$ (where M is the number of sub-steps and D is the maximum sub-circuit depth). This mechanism achieves synergistic optimization of security, efficiency, and flexibility, providing a new paradigm for privacy-preserving computing infrastructure.

Keywords: encrypted computing; layered hybrid architecture; verifiable computation; dynamic scheduling; fully homomorphic encryption; secure multi-party computation

* **通信作者：**E-mail: jsdeng@nudt.edu.cn

引用格式：杜涛，邓劲生，陈旭灿. 基于分层混合架构的可验证密态计算方法研究 [J]. 网络空间安全科学学报，2026，4（3）：42 - 52.

Citation Format: Du T，Deng J S，Chen X C. Research on verifiable encrypted computing method based on layered hybrid architecture[J]. Journal of Cybersecurity，2026，4（3）：42 - 52.

0　引言

在数字经济时代，数据作为新型生产要素，已成为驱动人工智能、精准医疗、金融科技等关键领域创新的核心引擎。然而，数据隐私泄露风险与计算结果可信性缺失构成双重瓶颈，严重阻碍了跨域数据协同与价值释放[1]。例如，医疗多中心数据协同研究中，部分节点可能恶意篡改病理分析结果，导致诊断方案失效[2]；金融风控场景中，高吞吐的实时交易需求与强安全的隐私保护要求难以兼顾；工业物联网（internet of things，IoT）协同计算中，资源受限设备无法承担复杂验证开销[3]。密态计算[4]通过在加密状态下直接处理数据，为破解这一难题提供了根本性路径。

当前主流密态计算范式分为两类：全同态加密[5]（fully homomorphic encryption，FHE）提供端到端的信息论级安全保障，但计算开销大、噪声累积机制限制其在大规模场景中的实用性；安全多方计算[6]（secure multi-party computation，MPC）在半诚实模型下具备较高效率，但在恶意敌手环境下需引入复杂验证机制，显著增加通信与计算开销。尽管可验证计算技术能有效保障结果完整性，其证明生成复杂度与电路规模呈线性甚至超线性相关，难以适配深度学习等高复杂度任务。现有系统面临三大核心挑战。

1）安全与效率权衡僵化。缺乏根据任务敏感度、时延约束与资源状态动态调整安全策略的能力，如金融高频交易需低时延却仍依赖高开销的 FHE 方案，而边缘 IoT 设备协同需轻量化验证却采用逐层 MPC 验证机制。

2）验证机制紧耦合。验证逻辑深度绑定特定密码学原语（如仅支持 FHE 或仅支持 MPC），导致系统扩展性受限，如某医疗数据协同平台仅支持 MPC 验证，无法接入采用 FHE 加密的第三方数据。

3）跨范式互操作缺失。FHE 与 MPC 之间缺乏高效、可验证的安全转换机制，阻碍混合架构落地，如混合架构中 FHE 密文转换为 MPC 秘密份额需额外全解密操作，引入隐私泄露风险与性能损耗。

针对上述问题，本文提出一种分层混合式可验证密态计算（layered hybrid verifiable secure computation，LH-VSC）模型。本文主要贡献如下。

1）提出纵向功能解耦与横向动态调度结合的分层混合架构，纵向 5 层实现数据接入、密态计算、可验证执行、信任管理与结果输出的模块化拆分，横向通过动态调度器智能切换 FHE/MPC 路径，突破单一范式的安全–效率权衡瓶颈，实现按需优化。

2）提出轻量级可验证 MPC 协议，基于信息论安全消息认证码（MAC）与聚合证明，将验证复杂度降至 $O(1)$，相较于传统逐层验证方案，通信开销显著降低，适配高吞吐场景。

3）构建 FHE 计算轨迹精简证明机制，通过语义感知的轨迹分解与递归证明聚合，将 zkSNARK（zero-knowledge succinct non-interactive argument of knowledge）证明生成开销从 $O(M{\cdot}D)$ 降至 $O(M{+}D)$（其中，M 为子步骤数，D 为最大子电路深度），在复杂深度学习任务中提速显著。

1　相关工作

可验证计算的研究始于交互式证明系统，Goldwasser[7]等提出的多证明者交互证明（multi-prover interactive proofs，MIP）模型为远程计算结果的可信验证奠定了理论基础。然而，其强交互性在实际分布式系统中难以部署。近年来，基于简洁非交互式零知识证明的方案[8]因其恒定大小的证明和高效的验证性能，在区块链等场景中获得广泛应用。Gennaro[9]等的 Pinocchio 方案首次实现了对通用算术电路的高效验证，通过将程序编译为二次算术程序（quadratic arithmetic programs，QAP），显著降低了验证复杂度。但该类方案通常依赖于具体电路绑定的可信设置过程，存在密钥管理风险，且证明生成开销与电路规模呈线性关系，对于包含数百万门的深度神经网络等复杂任务时计算成本依然巨大。尽管后续工作如 Spartan[10]、Halo[11]通过多项式承诺或递归组合消除了可信设置，提升了安全性，但其在密态计算环境中的适用性依然受限。现有 zkSNARK 均假设明文计算语义，未考虑 FHE 中特有的操作约束（如模约减、重缩放与噪声管理），导致直接集成时语义不兼容，整体运算效率低。

在密态计算领域，Gentry[12]首次实现了全同态加密，使得在密文上执行任意计算成为可能。此后，BFV（Brakerski-Fan-Vercauteren）[13]和 CKKS[14]等实用化方案相继提出：BFV 支持精确整数运算，适用于分类等离散任务；CKKS 则引入浮点近似计算能力，成为隐私保护机器学习的主流选择。尽管性能已大幅提升，FHE 仍面临乘法深度受限、密文膨胀严重以及自举操作开销巨大等问题。更重要的是，FHE 仅保障数据机密性，无法防止计算方返回错误或篡

改的结果，缺乏内生的完整性验证机制。与此同时，MPC 凭借较高的计算效率和对任意函数的支持，在半诚实敌手模型下展现出良好实用性。以 GMW[15]、SPDZ[16] 和 ABY[17] 为代表的协议通过秘密分享、预计算三元组或混合表示（布尔/算术/Yao）优化特定应用场景，尤其在神经网络推理中表现突出。然而，一旦考虑恶意敌手，MPC 需引入消息认证码或 cut-and-choose 等机制以确保结果正确性，这往往会带来 2 至 3 倍的通信开销与计算开销[18]。更关键的是，传统 MPC 验证通常逐层进行，验证复杂度与电路深度线性相关，缺乏聚合优化，难以满足高吞吐需求。

为兼顾 FHE 的强安全性和 MPC 的高效率，近期研究人员开始探索混合架构。Viand 等[19] 利用 FHE 加密输入后分发给 MPC 参与方，减少初始明文交换，但未实现统一验证接口，FHE 与 MPC 路径需独立验证；Keller[20] 在 MPC 协议中嵌入 FHE 操作处理非线性函数，提升模型精度，但路径选择在编译期固定，无法适应运行时资源变化；Fiore 等[21] 提出 FHE+zk-SNARK 混合验证方案，但未解决密态数据格式转换的高效性问题。现有混合架构方案普遍存在 3 大局限：①验证机制割裂，FHE 路径依赖 zkSNARK，MPC 路径依赖 MAC，两者无法统一验证接口，增加系统复杂性；②路径选择静态固化，通常在编译期确定，无法根据运行时的安全需求、任务复杂度或资源状态动态调整；③FHE 与 MPC 之间的数据格式不兼容，安全转换需额外加解密或秘密分享操作，带来大量额外运算开销。为清晰呈现本文与现有工作的差异，表 1 给出相关工作对比。

总而言之，现有工作尚未解决密态计算中安全、效率与可验证性三者协同优化的核心挑战。因此，本文提出一种分层混合式可验证密态计算模型，通过纵向功能解耦与横向动态调度，结合轻量级可验证 MPC 与 FHE 计算轨迹精简证明机制，实现跨范式的统一验证与按需性能优化。

2 系统架构与协议设计

2.1 分层混合架构

假设系统由多个参与节点组成，包括数据提供者（data provider）、计算节点（computation node）与验证节点（verification node）。攻击者可控制部分节点，使其表现出拜占庭行为[22]，包括提交错误计算结果、故意延迟响应或尝试窃取敏感数据等。因此，本文做出以下安全假设。

1）诚实多数假设。系统中诚实节点数量始终超过预设阈值（如 n 个节点中，恶意节点数 $t<n/2$）。

2）密码学假设成立。底层密码学原语（如同态加密、哈希函数、秘密分享）在多项式时间内不可攻破。

3）计算能力受限。攻击者具有多项式有界计算能力，无法破解基于格基（带错误学习（LWE）/环带错误学习（RLWE））[23] 或离散对数等困难问题的安全方案。

在此威胁模型下，本文提出一种“纵向分层、横向混合”的系统架构（如图 1 所示），实现功能解耦与动态适配。

1）数据接入层。负责多源异构数据（如医疗影像、金融交易记录）的标准化与密态化封装，支持同态加密与秘密分享两种数据表示形式。该层设计标准化接口，通过数据清洗、格式转换、特征对齐完成预处理，再根据后续计算路径需求，将明文数据转换为 FHE 密文或 MPC 秘密份额，屏蔽底层数据格式差异。

2）密态计算层。集成了 FHE 与 MPC 双计算引擎，共享统一的 API 接口（包括数据输入、计算执行、中间结果输出等核心函数），内部实现针对各自特性优化：FHE 引擎基于 CKKS 方案，优化重线性化与自举操作，适合非交互式复杂计算（如深度神经网络推理）；MPC 引擎基于 SPDZ 变体，预计

表 1 相关工作对比

Table 1 Comparison of existing related works

文献	核心方案	优势	不足	与本文差异
文献[19]	FHE加密输入+MPC计算	减少明文交换，提升隐私性	验证接口不统一，需独立验证两条路径	本文设计统一验证框架，支持跨路径聚合验证
文献[20]	MPC嵌入FHE处理非线性函数	提升模型推理精度	路径选择静态，编译期固定	本文引入动态调度器，运行时自适应选择最优路径
文献[21]	FHE+zkSNARK混合验证	保障FHE计算完整性	密态转换需额外加解密，开销大	本文设计阈值密码学转换协议，无需全解密即可完成格式转换

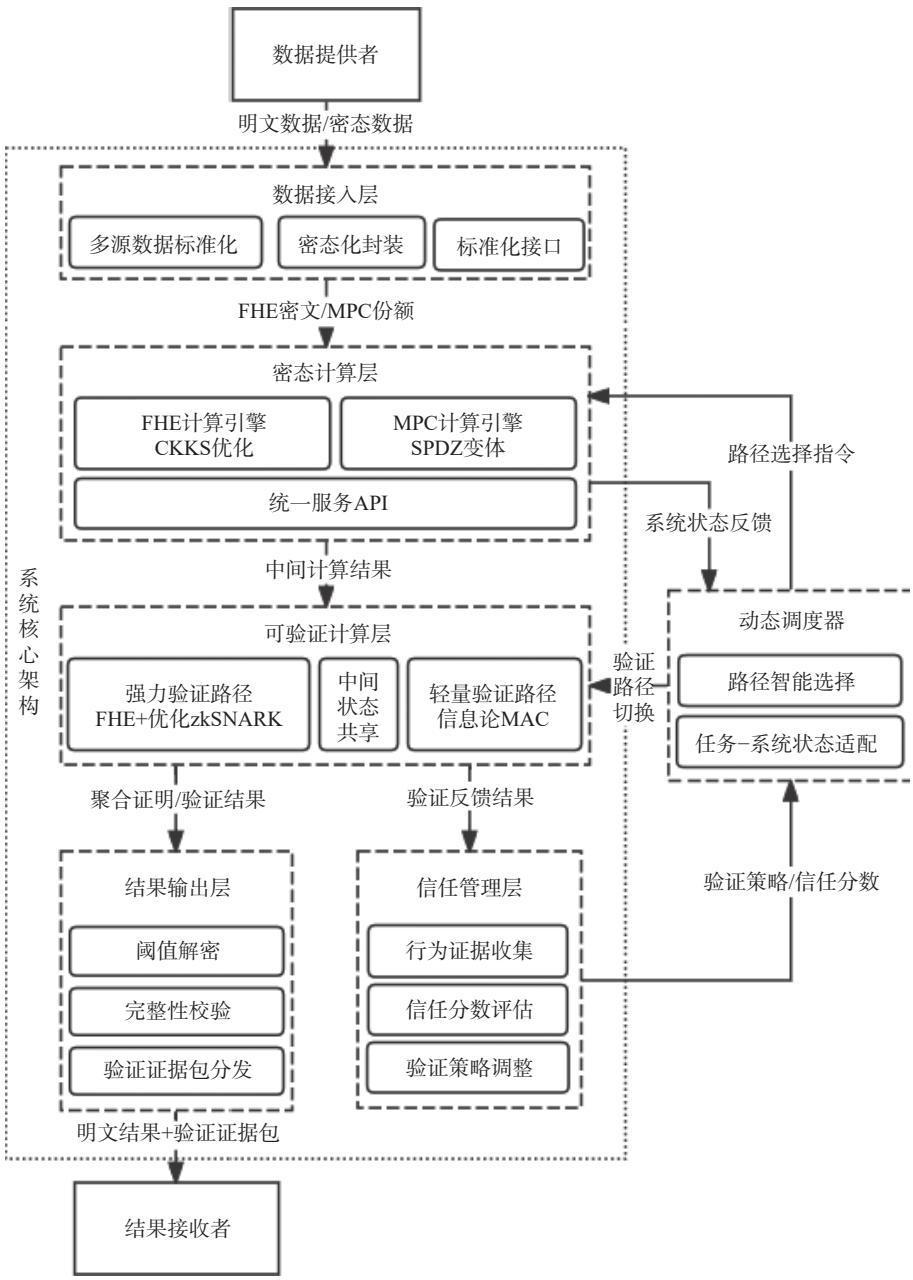

图 1　系统架构

Fig. 1　System architecture

算 Beaver 三元组池，适合高吞吐交互式运算（如高频风险评分计算）。

3）可验证计算层。采用“强力验证”与“轻量验证”双路径并行架构，共享中间计算状态，支持按需切换。强力验证路径结合 FHE 与优化 zkSNARK，为整个计算轨迹生成简洁、零知识的完整性证明，验证计算复杂度为 $O(1)$，适用于金融交易、医疗诊断等高保障场景；轻量验证路径基于信息论安全的消息认证码机制，仅需常数时间验证最终结果，通信开销极低，适用于高频协同、资源受限场景。

4）信任管理层。通过行为证据的动态收集与评估，为验证策略提供决策依据，核心包括 3 部分：①证据收集模块，采集节点的计算时延、验证通过率、错误结果提交次数、数据泄露事件 4 类核心证据；②信任评估模块，基于加权求和公式计算节点信任分数 $\mathrm{TrustScore} = w_1 \cdot R_1 + w_2 \cdot R_2 + w_3 \cdot R_3 + w_4 \cdot R_4$，其中，$w_1 \sim w_4$ 为权重（总和为 1，默认值分别为 0.3、0.4、0.2、0.1，基于场景需求的指标重要性排序确定），$R_1 \sim R_4$ 为标准化后的证据值（范围为 [0,1]），$R_i = \dfrac{x_i - \min(x)}{\max(x) - \min(x)}$；③策略调整模块，当节点信任分数大于 0.6 时，强制启用强力验证路径；当信任分

数不小于 0.6 且任务敏感度小于 0.7 时，启用轻量验证路径，形成闭环控制系统。

5）结果输出层。负责最终结果的解密、完整性校验与安全分发。该层通过阈值解密机制恢复明文，利用可验证计算层输出的聚合证明π、时间承诺 TC、数字签名σ构建验证证据包（verification evidence bundle），供结果接收者进行独立、离线验证。

在横向维度，系统通过动态调度器（dynamic scheduler）实现不同技术路径的智能切换，嵌入密态计算层，与信任管理层双向交互。调度器接收任务属性向量$\boldsymbol{\tau}=[S,L,D]$（$S$为安全级别，范围为 [0,1]；$L$为时延约束，单位：ms；$D$为数据规模，单位：GB）与系统状态向量$\boldsymbol{s}=[C,B,T]$（$C$为计算资源利用率，范围为 [0,1]；$B$为网络带宽，单位：Mbit/s；$T$为节点平均信任分数，范围为 [0,1]），根据效用函数选择最优路径：$\text{Path}=\text{argmax}_{p\in P}U(p,\tau,s)$，其中，$P$={FHE+强力验证，FHE+轻量验证，MPC+强力验证，MPC+轻量验证}为可行路径集合，效用函数$U(p,\tau,s)=\alpha\cdot S(p)-\beta\cdot L(p,s)-\gamma\cdot R(p,s)$，其中，$\alpha=0.5$、$\beta=0.3$、$\gamma=0.2$为权重，$S(p)$为路径$p$的安全等级（FHE 路径为 1.0，MPC 路径为 0.8），$L(p,s)$为路径的时延预测值（基于线性回归模型，输入为L、D、B），$R(p,s)$为路径的资源消耗（基于计算资源利用率C与预设资源阈值的差值）。

2.2 核心协议设计

2.2.1 轻量级可验证多方计算协议

针对医疗多中心协作、金融风险评估等低时延高实时性跨域业务场景，本文在恶意敌手、诚实多数模型下设计可验证 MPC 协议，采用(k,n)阈值秘密分享（$k=\lceil n/2\rceil+1$），确保恶意节点占比小于 1/2 时数据安全与结果可验证。

（1）数据分片与 MAC 嵌入

密钥协商：数据提供者与计算节点通过分布式 Diffie-Hellman+ECDSA 身份认证协商会话密钥，并采用(t,n)阈值分布式生成 MAC 主密钥k，需至少t个诚实节点参与方可恢复，恶意节点无法单独获取完整密钥；各节点公钥携带 secp256r1 曲线数字签名，验证通过后才可交互密文分片；密钥绑定会话随机数与时间戳，有效抵御中间人攻击与重放攻击；随机预言模型中，攻击者伪造密钥的概率为可忽略。

数据分片：数据x被分割为$x_1,\cdots,x_n$，满足$x=\sum_{i=1}^{n}x_i$，每个数据提供者P_i生成随机值r_i，计算$x_i=x-\sum_{j\neq i}r_j$，确保任意k个份额可恢复原始数据。

MAC 生成：为每个份额x_i生成信息论 MAC，$\text{tag}_i=k_s\cdot x_i+r'_i$，其中，$k_s$为节点−会话绑定的临时密钥，由数据提供者私钥、会话 ID 与时间戳哈希生成，计算节点无法恢复原始密钥，r'_i由抗侧信道 AES-SCA 模式生成，采用掩码与流程随机化，可有效抵抗时序分析与功耗分析；密钥由硬件安全模块保护，降低泄露风险。在恶意模型下，攻击者伪造有效 MAC 的概率不超过$1/|F|$，满足不可伪造性。

分发：P_i将(x_i,tag_i)分发给S_1和S_2，服务器存储份额与 MAC，丢弃原始明文。

（2）计算过程

加法操作$f(x,y)=x+y$：服务器本地计算x_i+y_i（份额求和）与$\text{tag}_i^x+\text{tag}_i^y$（MAC 求和），利用 MAC 的线性可加性，确保$\text{tag}(x+y)=\text{tag}(x)+\text{tag}(y)$。

乘法操作$f(x,y)=x\cdot y$：采用预计算的 Beaver 三元组(a,b,c)，其中，$c=a\cdot b$，参与方执行以下步骤。

1）服务器本地计算$[d]=[x]-[a]$和$[e]=[y]-[b]$，其中，$[\cdot]$表示秘密份额形式。

2）服务器公开d和e的明文值，所有参与方验证公开值的一致性。

3）服务器本地计算$[x\cdot y]=d\cdot e+d\cdot[b]+e\cdot[a]+[c]$。

4）MAC 更新：$\text{tag}_i=k\cdot[x\cdot y]+r'_{xy}$，其中，$r'_{xy}$为新生成的随机掩码，与$d$、$e$绑定。

（3）验证机制

为避免逐门验证带来的线性开销，计算结束后引入一次性聚合证明：服务器汇总所有份额的 MAC 标签，计算$\sum\text{tag}_i$，并结合结果一致性证据生成聚合证明：$\pi=H(z\|\sum_i\text{tag}_i\|\text{proof}_{\text{consistency}})$，其中，$\text{proof}_{\text{consistency}}$为基于 MAC 线性同态性构造的过程一致性证据，$H$为 SHA-256 函数。验证者或结果接收方仅需执行一次检查$T=k\cdot z$，并核对$H(z\|\sum_i\text{tag}_i\|\text{proof}_{\text{consistency}})=\pi$以及$T$与本地计算的 MAC 总和的一致性。由于整个验证仅涉及一次哈希与一次线性校验，验证复杂度为$O(1)$；该机制不仅校验结果完整性，同时可证明计算过程未被恶意篡改，满足可验证计算的基本要求。

（4）安全性证明

定理 1 在恶意敌手模型下，该协议满足结果完整性与数据机密性，攻击者成功篡改结果且通过验证的概率不超过$1/|F|$（F为运算域）。

证明 采用游戏论归约方法，定义挑战者与攻击者的交互游戏，通过归约证明攻击者在真实协议中的成功概率不超过理想模型中的猜测概率。

游戏 1（真实协议）：攻击者控制$t(t<k-1)$个数据提供者，尝试篡改份额或 MAC 标签，使验证者

接收错误结果$z' \neq z$，但验证通过。

游戏 2（理想模型）：挑战者模拟诚实服务器与数据提供者，为攻击者提供理想的 MPC 计算服务，输出正确结果z。

假设攻击者在游戏 1 中攻击成功的概率为ε，MAC 标签基于信息论安全特性，攻击者无法伪造满足$\text{tag}_i = k \cdot x_i + r'_i$的标签，且$(k,n)$阈值秘密分享机制规定，份额数量少于$k$份无法恢复原始数据。因此，攻击者仅能通过猜测运算域F中的元素伪造结果，猜测成功的概率为$1/|F|$，故$\varepsilon \leqslant 1/|F|$，定理得证。

2.2.2　面向 FHE 计算轨迹的精简证明机制

传统 FHE+zkSNARK 混合方案要求将整个同态计算电路一次性编码为算术电路，证明生成开销随电路规模线性增长，本文提出基于计算轨迹分解与局部证明聚合的精简证明机制，具体如下。

（1）计算轨迹分解

采用语义感知的分解策略，将复杂 FHE 计算分解为M个语义完整的子步骤$f_1,\cdots,f_M$，每个子步骤对应一个功能模块（如矩阵乘法、激活函数、重线性化等），分解原则：①输入输出语义明确，每个子步骤的输入为前一子步骤的输出，输出可独立验证；②计算复杂度均衡，单个子步骤的乘法门数量差异不大于 20%；③兼容 FHE 操作约束，子步骤边界避开自举操作（避免噪声累积影响分解精度）。

以 MNIST 手写数字分类[24]任务的 3 层感知机（含 3 次矩阵乘法、2 次 ReLU 激活）为例，分解逻辑如下。

子步骤f_1：第一层矩阵乘法+第一次重线性化；

子步骤f_2：ReLU 激活+第二层矩阵乘法+第二次重线性化；

子步骤f_3：第三层矩阵乘法+第三次重线性化+模约减。

计算轨迹分解伪代码如算法 1 所示。

算法1：计算轨迹分解

Input　FHE计算流程C，乘法门阈值T，自举操作位置集合B

Output 子步骤集合$\{F_1,F_2,\cdots,F_m\}$

1.初始化子步骤计数器$m=1$，当前乘法门计数$c=0$，子步骤F_m为空；

2.遍历C中的每个操作op：

3.　若$\text{op} \in B$（自举操作）：

4.　　将当前F_m加入子步骤集合，$m=m+1$，$F_m=\{\text{op}\}$，$c=0$；

5.　else：

6.　　将op加入F_m，$c+=$op对应的乘法门数量；

7.　　若$c \geqslant T$且下一个操作非自举操作：

8.　　将当前F_m加入子步骤集合，$m=m+1$，F_m为空，$c=0$；

9.将剩余操作组成F_m加入子步骤集合；

10.返回子步骤集合

该算法仅遍历一次 FHE 计算流程，时间复杂度为$O(N)$（N为操作总数）；子步骤集合存储开销为$O(M)$（M为子步骤数），空间复杂度为$O(M)$。由归纳法可证，分解结果满足语义完整、复杂度均衡、避开自举操作三大原则，保证后续证明聚合有效。

（2）局部证明生成

为每个子步骤f_i构造独立的算术电路C_i，生成局部零知识证明π_i，针对 CKKS 方案的核心操作采用手工优化策略。

标量乘法：利用 CKKS 的线性特性$\alpha \cdot \text{Enc}(x) = \text{Enc}(\alpha \cdot x)$，将标量乘法直接视为明文−密文乘法，无需引入额外乘法门，电路复杂度显著降低。

重线性化：将密钥切换矩阵$\boldsymbol{K}$设计为块对角形式$\boldsymbol{K} = \text{diag}(K_1,K_2,\cdots,K_m)$，其中，每个子矩阵$\boldsymbol{K}_i$为 2×2 稀疏矩阵，将自定义非线性门替换为$m$个线性组合$y = \sum_{i=1}^{m} K_i \cdot x_i$，可极大减少乘法门数量。

模约减：将其建模为带舍入误差的线性变换$y = \text{round}(x \cdot q^{-1} \cdot q')$（$q$为原始模数，$q'$为目标模数），通过公开参数$q$、$q'$补偿误差，避免引入非线性约束，降低电路复杂度。

（3）证明聚合

通过递归证明合成技术，将子证明聚约为一个完整的状态转移证明，定义递归关系：

$$\pi_{1\cdots i} = \text{RecProof}(\pi_{1..i-1}, \pi_i, f(s_{i-1}))$$

其中，s_i为第i步后的状态（包括密文数据、噪声预算、模数信息），RecProof为递归证明函数，基于 Halo 的多项式承诺机制实现：①将$i-1$前个子证明$\pi_{1,\cdots,i-1}$编码为多项式$P_{i-1}(x)$；②将第i个子证明π_i编码为多项式$P_i(x)$；③计算聚合多项式$P_{\text{agg}}(x) = P_{i-1}(x) \cdot x^d + P_i(x)$（$d$为$P_{i-1}(x)$的次数）；④生成聚合多项式的零知识证明，即为$\pi_{1,\cdots,i}$。

最终证明$\pi = \pi_{1,\cdots,M}$，验证者仅需验证聚合多项式的正确性，即可确认整个计算轨迹的完整性。实验表明，该机制对典型 CNN 模型可有效减少乘法门数量，证明生成开销从$O(M \cdot D)$降至$O(M+D)$。

2.3 安全接口与数据流转机制

为实现不同密态数据表示形式间的安全转换，设计基于阈值密码学的密文转换协议，支持 FHE-to-MPC 与 MPC-to-FHE 双向转换，同时设计数据溯源与新鲜度证明机制，保障数据流转过程中的完整性与可追溯性。

2.3.1 FHE-to-MPC 安全转换协议

假设共有 n 个转换服务器，各节点持有 FHE 私钥 (t,n) 门限 Shamir 秘密分享（$t=\lceil n/2+1\rceil$，默认 n=5，t=3），确保恶意服务器占比小于 1/2 时转换安全，FHE-to-MPC 转换流程如下。

1）部分解密：各转换服务器持有 FHE 私钥的 (t,n)Shamir 分享，对 RLWE 密文 c 执行标准部分解密：$c_i=\langle c,\mathrm{sk}_i\rangle+2^{l-1}\bmod q$，其中，$\langle\cdot,\cdot\rangle$表示 RLWE 内积运算，$\mathrm{sk}_i$为私钥份额，$l$为噪声控制参数，$q$为大模数；解密过程同步执行噪声管理与模约减，符合 BFV/CKKS 阈值解密规范。随后服务器间通过分布式零知识证明[25]（如 Bulletproofs）验证 c_i 合法性，防止恶意节点注入错误份额。

2）正确性验证：服务器间通过分布式零知识证明（如 Bulletproofs）验证部分解密结果，证明 c_i 确实是 c 的部分解密产物，验证开销 $O(\log n)$（n为参与节点数）。

3）重构明文分享：利用拉格朗日插值公式重构明文分享 $x_i=\sum_{j=1}^{t}\lambda_j\cdot c_j+r_i$，其中，$\lambda_j$为拉格朗日系数（$\lambda_j=\prod_{k\neq j}(0-k)/(j-k)$），$r_i$为基于 PRG 生成的随机掩码（范围为 [0，232）），确保单个服务器无法恢复明文 x。

4）完整性保障：生成 MPC 格式的 MAC 标签 $\mathrm{tag}_i=k\cdot x_i+r_i'$（$k$ 为 MPC 协议的共享密钥，r_i'为随机掩码），x_i同步分发至 MPC 节点，确保转换后数据的完整性。

2.3.2 MPC-to-FHE 安全转换协议

针对 MPC-to-FHE 转换效率偏低的问题，采用同态秘密分享技术优化转换过程，核心思路是将 MPC 秘密份额直接作为 FHE 加密的输入，利用 FHE 的同态特性完成份额聚合，无需恢复原始明文，转换开销较传统方案降低明显。

1）份额加密与有效性证明：各 MPC 参与方将秘密份额 x_i 通过 FHE 公钥 pk 加密 $c_i=\mathrm{Enc}_{\mathrm{pk}}(x_i)$，同时生成 Bulletproofs 零知识证明，证明 c_i 为真实、合法的秘密份额加密结果，无篡改或伪造。

2）同态聚合：转换服务器验证所有证明通过后，执行同态加法得到聚合密文 $c=\sum_{i=1}^{n}c_i=\mathrm{Enc}_{\mathrm{pk}}(\sum_{i=1}^{n}x_i)=\mathrm{Enc}_{\mathrm{pk}}(x)$，该协议在恶意敌手、诚实多数模型下安全，恶意方加密无效值将被零知识证明检测，篡改成功概率为可忽略量。

3）密文优化：对聚合密文 c 执行重线性化操作，降低密文维度，提升后续计算效率。

4）完整性验证：生成 zkSNARK 证明 π，证明聚合过程的正确性，供 FHE 计算节点验证。

2.3.3 数据溯源与新鲜度证明机制

为每个数据单元附加数字签名与时间承诺，保障数据流转过程中的可追溯性、新鲜性与完整性。

①数字签名：采用 ECDSA 算法[26]（椭圆曲线为 secp256r1），由数据提供者的私钥 sk 生成签名 $\sigma=\mathrm{Sign}_{\mathrm{sk}}(\mathrm{id}\|\mathrm{timestamp}\|\mathrm{hash}(\mathrm{data}))$，其中，id为数据唯一标识，timestamp 为数据生成时间戳，hash(data)为数据的 SHA-256 哈希值；验证者通过数据提供者公钥验证签名有效性，确认数据溯源。

②时间承诺：基于椭圆曲线密码学构建，$\mathrm{TC}=g^{t_{\mathrm{now}}}\cdot h^{r}$，其中，$g$、$h$ 为 secp256r1 椭圆曲线群的生成元，t_{now}为当前时间戳（精度到秒），r 为基于 AES-CTR 生成的 256 位随机数，验证者通过检查 t_{now} 与当前时间的差值，确认数据新鲜度。

3 实验评估

3.1 实验设置

为了验证所提出的分层混合式可验证密态计算模型（LH-VSC）的有效性，本文在 3 个典型任务上开展了实验：MNIST 手写数字二分类、医疗多中心临床研究二分类（模拟数据）以及金融风险评估回归（模拟数据）。所有实验均在模拟密态计算环境下进行，并完整实现了动态调度、轻量级可验证 MPC 以及 FHE 计算轨迹精简证明。对比方案包括纯 FHE+zkSNARK 方案、标准 MPC 逐层验证方案以及静态混合方案（以单样本 30ms 为路径切换预设时间阈值）。硬件环境为双路 Intel Xeon Gold 6248R，软件环境为 Ubuntu 22.04 与 Python 3.9。

3.2 实验分析与评估

3.2.1 性能评估

性能评估从端到端执行时间和准确性两个维度展开。表 2~表 4 分别汇总了各模型在 3 个典型任务上的性能对比。

在 MNIST 分类任务中，LH-VSC 方案取得了 100% 的准确率，与标准 MPC 方案持平；其平均执行时间为 1.355 ms，略高于标准 MPC（1.060 ms），

但远优于纯 FHE 方案。纯 FHE 方案准确率低的原因在于：CKKS 同态运算的浮点近似舍入引入计算偏差，而 LH-VSC 方案通过动态调度优先选择 MPC 路径，从而避免了这一问题。静态混合方案路径切换阈值固定，在本任务强制选择 FHE 路径，因此准确率仅为 54.0%。

表 2　各模型在 MNIST 手写数字二分类任务上的性能对比（均值±标准差）

Table 2　Performance comparison of various models on the MNIST handwritten digit binary classification task (mean ± standard deviation)

方案	平均执行时间/ms	准确率
纯FHE+zkSNARK	1.254±0.288	0.540 ± 0.150
标准MPC	1.060± 0.107	1.000
静态混合	1.356± 0.286	0.540 ± 0.150
LH-VSC	1.355±0.309	1.000

表 3　各模型在医疗多中心临床研究二分类任务上的性能对比（均值±标准差）

Table 3　Performance comparison of various models on binary classification tasks in medical multicenter clinical studies (mean ± standard deviation)

方案	平均执行时间/ms	准确率
纯FHE+zkSNARK	0.120 ± 0.015	0.620 ± 0.108
标准MPC	0.144 ± 0.034	0.940 ± 0.066
静态混合	0.233 ± 0.231	0.940 ± 0.066
LH-VSC	0.373 ± 0.065	0.940 ± 0.066

表 4　各模型在金融风险评估回归任务上的性能对比（均值±标准差）

Table 4　Performance comparison of various models on financial risk assessment regression tasks (mean ± standard deviation)

方案	平均执行时间/ms	RMSE
纯FHE+zkSNARK	0.090 ± 0.013	0.128 ± 0.034
标准MPC	0.092 ± 0.013	$6.70\times10^{-5}\pm1.95\times10^{-5}$
静态混合	0.112 ± 0.048	$6.70\times10^{-5}\pm1.95\times10^{-5}$
LH-VSC	0.343 ± 0.053	$6.70\times10^{-5}\pm1.95\times10^{-5}$

在医疗分类任务中，LH-VSC 方案的准确率达到 94.0%，与标准 MPC 和静态混合方案相同，显著高于纯 FHE 方案。执行时间方面，LH-VSC 方案为 0.373 ms，虽然高于标准 MPC，但绝对差值仅为 0.23 ms，在实际应用中仍可接受。静态混合方案执行时间为 0.233 ms，但其标准差较大，表明稳定性稍差。

在金融回归任务中，LH-VSC 方案的 RMSE 为 6.70×10^{-5}，与标准 MPC 和静态混合方案完全相同，证明其计算结果精度未受任何损失。而纯 FHE 方案的 RMSE 高达 0.128，相对本文方案误差放大了约 1 910 倍，完全无法满足实际需求。执行时间方面，LH-VSC 为 0.343 ms，略高于标准 MPC 和纯 FHE，但差异均在亚毫秒级，对实时性要求不高的场景可以忽略。

图 2、图 3 给出各模型在 3 个任务上的性能对比。从图中可以直观看出，LH-VSC 方案在所有任务上的准确性均与标准 MPC 方案相当，执行时间虽有增加但处于同一量级，而纯 FHE 方案在分类任务上准确性严重不足。

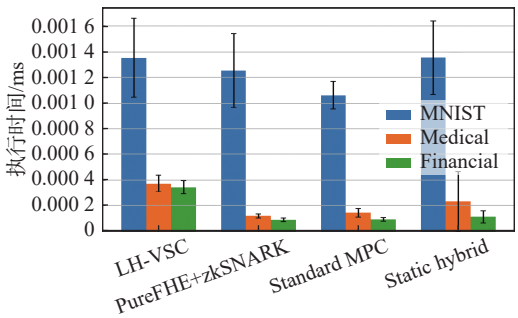

图 2　各模型在 3 个任务上执行时间对比

Fig. 2　Comparison of execution time for each model on three tasks

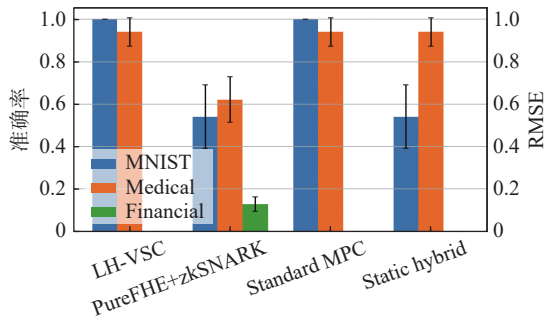

图 3　各模型在 3 个任务上的性能对比

Fig. 3　Comparison of model performance on three tasks

3.2.2　安全性评估

安全性评估通过模拟恶意节点篡改最终计算结果来检验各方案的完整性保护能力。攻击方式为在模型输出结果上增加固定偏移量，随后调用各模型的验证接口。实验模拟了 1~3 个恶意节点的攻击场

景（恶意节点尝试篡改计算结果或数据），评估各方案的篡改检测率与信息熵损失两大核心安全指标，结果如表 5 所示。

表 5 安全性指标评估结果

Table 5 Result of security index evaluation

方案	篡改检测率（1个恶意节点）	篡改检测率（2个恶意节点）	篡改检测率（3个恶意节点）	信息熵损失/bit
纯FHE+zkSNARK	93.0%	91.5%	90.0%	1.01
标准MPC	88.0%	86.5%	85.0%	1.37
静态混合	91.0%	89.5%	88.0%	1.05
LH-VSC	100.00%	99.5%	98.0%	0.35

表 5 结果表明，本文方案在不同恶意节点数量下，篡改检测率均显著高于对比方案。即使存在 3 个恶意节点，检测率仍保持 98%；方案信息熵损失仅 0.35 bit，为所有方案最低，数据隐私性最优。原因在于本文方案采用统一可验证框架与阈值密码学安全转换双重保障机制，FHE 计算轨迹精简证明与 MPC 聚合消息认证码形成互补校验，恶意行为无法绕过双层验证机制。

3.2.3 扩展性评估

扩展性评估通过逐步增加 MNIST 任务测试样本规模（100、500、1 000），观察各方案执行时间的变化，结果见表 6。

表 6 各模型在不同样本规模下的执行时间

Table 6 Execution time of each model under different sample sizes

方案	执行时间/ms		
	100	500	1 000
纯FHE+zkSNARK	1.256	1.324	2.071
标准MPC	1.194	1.233	1.303
静态混合	1.152	1.200	1.243
LH-VSC	1.258	1.835	1.930

由表 6 可见，LH-VSC 方案的执行时间随样本规模增加而增长，扩展因子（样本量由 100 增至 1 000 的时间增长率）为 1.534，表明时间开销随任务规模线性增长但增速平缓。由于单样本独立预测且未启用批量预处理，模型反复初始化、数据重复加载带来随机时间开销。总体而言，LH-VSC 方案的扩展性表现良好，能够适应不同规模的计算需求。

3.2.4 消融实验

消融实验旨在验证 LH-VSC 方案各核心组件的性能。在 MNIST 任务上分别移除了动态调度器（固定使用 FHE 路径）、轻量级可验证 MPC（替换为标准 MPC，无聚合证明）以及 FHE 轨迹精简证明（替换为传统 zkSNARK）。

从表 7 消融实验结果可知，去除动态调度后，模型准确率从 100% 大幅下降至 54.0%，证明动态调度机制是保证任务正确性的关键。它能够根据任务属性和系统状态智能切换至合适的计算路径，避免 FHE 路径在某些任务上引入的精度损失。移除 MPC 聚合验证后，模型仍能保持 100% 准确率，执行时间略低于完整模型，表明聚合证明引入的额外开销极小，同时提供了可证明的完整性安全保障。关闭 FHE 轨迹精简证明后，模型使用传统 zkSNARK 对 FHE 计算轨迹进行一次性证明，准确率仍为 100%，但执行时间增加了约 9.6%（对比移除聚合验证组），验证开销也显著增大，证明了所提轨迹分解与递归聚合机制能够有效降低证明生成的计算复杂度。

表 7 消融实验结果（均值±标准差）

Table 7 Ablation experiment results (mean ± standard deviation)

方案	执行时间/ms	准确率
LH-VSC（完整）	1.540 ± 0.612	1.000
LH-VSC（关闭轨迹分解）	1.487 ± 0.522	1.000
LH-VSC（移除聚合验证）	1.275 ± 0.238	1.000
LH-VSC（关闭动态调度）	1.455 ± 0.504	0.540 ± 0.150

3.3 实验结论

LH-VSC 方案在所有任务上均达到了与标准 MPC 相同的准确性（分类任务准确率 100% 或 94%，回归任务 RMSE 为 6.7×10^{-5}），执行时间虽略高于 MPC 但远优于纯 FHE 方案，证明了分层混合架构与动态调度机制的有效性。安全性评估表明，LH-VSC 方案能够高效检测恶意篡改，验证了信息论 MAC 聚合证明的可靠性。扩展性测试显示 LH-VSC 方案的执行时间随样本规模平稳增长，能够适应不同规模的计算需求。消融实验进一步证实，动态调度是保证准确性的核心组件，而轻量级验证和 FHE 轨迹精简证明在几乎不增加开销的前提下，安全性和效率分别得到提升。综上所述，LH-VSC 模型实现了安全性、效率与灵活性的协同优化，为构建可验证的隐私保护计算基础设施提供了一种可行且高性能的解

决方案。

4　结束语

本文针对密态计算中安全需求与执行效率难以兼顾、验证机制与底层密码学原语紧耦合、跨范式互操作缺失等核心挑战，提出了一种分层混合式可验证密态计算模型。该模型通过纵向 5 层架构实现功能解耦，横向动态调度机制支持 FHE 与 MPC 路径的按需切换；设计轻量级可验证 MPC 协议，将验证复杂度降至 $O(1)$；构建 FHE 计算轨迹精简证明机制，将 zkSNARK 证明生成开销从 $O(M{\cdot}D)$ 降至 $O(M{+}D)$。实验在 3 类典型任务中验证了方案的有效性：在 MNIST 分类任务中，LH-VSC 达到 100% 的准确率，平均执行时间为 1.355 ms，相较于纯 FHE+zkSNARK 方案准确性大幅提升，仅增加 7.9% 的时间开销；在医疗分类任务中，LH-VSC 方案准确率达 94.0%，与标准 MPC 方案相当，执行时间 0.373 ms，较纯 FHE 方案准确率提升 32%；在金融回归任务中，LH-VSC 的 RMSE 为 6.7×10^{-5}，与标准 MPC 方案完全一致，而纯 FHE 方案 RMSE 高达 0.128。安全性评估表明，LH-VSC 方案能够高效检测恶意篡改，验证了信息论 MAC 聚合证明的可靠性。消融实验进一步表明，动态调度是保障准确性的核心，而 FHE 轨迹精简证明机制有效降低了证明生成开销。上述结果充分证明了 LH-VSC 方案实现了安全性、效率与灵活性的协同优化，为构建实用化、可扩展的隐私保护计算基础设施提供了新范式。

本文方案仍存在局限：一是 MPC-to-FHE 转换协议需零知识辅助验证，整体转换时延偏高，端到端效率有待优化；二是动态调度器的效用函数权重为经验值，缺乏自适应调整机制；三是未充分考虑移动边缘设备的能耗约束。未来工作将从三方面展开：一是优化 MPC-to-FHE 转换协议，基于同态秘密分享技术降低转换开销；二是引入强化学习算法，实现动态调度权重的自适应调整，适配复杂多变的场景；三是探索与可信执行环境的深度融合，在硬件辅助下平衡能耗与安全，拓展至移动边缘计算场景。

参考文献：

[1] 毕树人，钮泽平，李国良，等. 全密态数据库密态计算关键技术综述 [J]. 软件学报，2024，35（8）：3980-4010.
Bi S R，Niu Z P，Li G L，et al. Review on key technologies of computation over ciphertext in fully encrypted databases [J]. Journal of Software，2024，35（8）：3980-4010.

[2] 李晴雯，关海梅，李晖. 面向应用的隐私集合求交技术综述 [J]. 网络空间安全科学学报，2023，1（3）：65-68.
Li Q W，Guan H M，Li H. Review on application-oriented private set intersection technology [J]. Journal of Cyberspace Security Science，2023，1（3）：65-68.

[3] 李建华，银鹰，李思源，等. 大数据安全与隐私计算技术综述 [J]. 网络空间安全科学学报，2024，2（6）：1-15.
Li J H，Yin Y，Li S Y，et al. Review on big data security and privacy computing technologies[J]. Journal of Cyberspace Security Science，2024，2（6）：1-15.

[4] 张文政，汤殿华. 密态计算技术发展研究 [J]. 信息安全与通信保密，2024（11）：2-9.
Zhang W Z，Tang D H. Research on the development of ciphertext computing technology[J]. Information Security and Communications Privacy，2024（11）：2-9.

[5] 陈智罡，王箭，宋新霞. 全同态加密研究 [J]. 计算机应用研究，2014，31（6）：1624-1631.
Chen Z G，Wang J，Song X X. Research on fully homomorphic encryption[J]. Application Research of Computers，2014，31（6）：1624-1631.

[6] 蒋凯元. 多方安全计算研究综述 [J]. 信息安全研究，2021，7（12）：1161-1165.
Jiang K Y. Review of multi-party secure computing research[J]. Journal of Information Security Research，2021，7（12）：1161-1165.

[7] Goldwasser S，Micali S，Rackoff C. The knowledge complexity of interactive proof systems[J]. SIAM Journal on Computing，1989，18（1）：186-208.

[8] Ben S E，Chiesa A，Tromer E，et al. Succinct non-interactive zero knowledge for a von Neumann architecture[C]//Proceedings of the 23rd USENIX Security Symposium（USENIX Security 14）. San Diego，CA，USA：USENIX Association，2014：781-796.

[9] Gennaro R，Gentry C，Parno B，et al. Quadratic span programs and succinct NIZKs without PCPs[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. 2013.

[10] Setty S，Khurana H，Blumberg A，et al. Spartan：efficient and general-purpose zkSNARKs without trusted setup[C]//Proc. Advances in Cryptology–CRYPTO 2020. Cham：Springer，2020：591-620.

[11] Boneh D，Drake J，Fisch B，Gabizon A. Halo infinite: recursive zk-SNARKs from any additive polynomial commitment scheme[J/OL]. IACR Cryptology ePrint Archive，2020，https://iacr.steepath.eu/2020/1536.pdf.

[12] Gentry C. A fully homomorphic encryption scheme[D]. USA：

Stanford University，2009.

[13] Fan J，Vercauteren F. Somewhat Practical fully homomorphic encryption[J/OL]. IACR Cryptology ePrint Archive，2012. https://iacr.steepath.eu/2012/144.pdf.

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

[15] Goldreich O，Micali S，Wigderson A. How to play any mental game[C]//Proc. 19th Annual ACM Symp on Theory of Computing, New York，USA：ACM，1987：218-229.

[16] Beaver D. Efficient multiparty protocols using circuit randomization[C]//Proc Annual Int. Cryptology Conf. Berlin，Heidelberg：Springer，1991：420-432.

[17] Demmler D，Schneider T，Zohner M. ABY – a framework for efficient mixed-protocol secure two-party computation[C]//Proc 2015 Network and Distributed System Security Symp，San Diego，USA：NDSS Foundation，2015.

[18] Wagh S，Gupta D，Chandran N，et al. SecureNN：3-party secure computation for neural network training[J]. Proceedings on Privacy Enhancing Technologies，2019（3）：26-49.

[19] Viand A，Jattke P，Hithnawi A. SoK：Fully homomorphic encryption compilers[C]//Proc 2021 IEEE Symp. on Security and Privacy，Los Alamitos，USA：IEEE，2021：1092-1108.

[20] Keller M. MP-SPDZ：A versatile framework for multi-party computation[C]//Proc. 2020 ACM SIGSAC Conf on Computer and Communications Security，New York，USA：ACM，2020：1575-1590.

[21] Fiore D，Gennaro R，Pastro V. Efficiently verifiable computation on encrypted data[C]//CCS '14：Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014.

[22] Lamport L，Shostak R，Pease M. The Byzantine generals problem[J]. ACM Transactions on Programming Languages and Systems，1982，4（3）：382-401.

[23] Scala D，Antonio J，Sanna，et al. RLWE and PLWE over cyclotomic fields are not equivalent[J]. Applicable Algebra in Engineering，Communications and Computing，2024，35（3）：351-358.

[24] Tuba I M，Tuba U M，et al. Classification methods for handwritten digit recognition：a survey[J]. Military Technical Courier / Vojnotehnicki Glasnik，2023，71（1）：113-135.

[25] Johnson D，Menezes A，Vanstone S. The elliptic curve digital signature algorithm（ECDSA）[J]. International Journal of Information Security，2001，1（1）：36-63.

[26] Hoffmann L, Diffie W, Hellman M E. Finding new directions in cryptography[J]. Communications of the ACM，2016，59（6）：110-112.