计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.010063] [CSTR: 32024.14.csa.010063]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于机密计算的大语言模型安全推理方案①

崔　越[1], 冯　伟[2], 秦　宇[2], 马鸿展[2], 冯登国[2]

[1](中国科学院大学, 北京 100049)
[2](中国科学院 软件研究所, 北京 100190)
通信作者: 冯　伟, E-mail: fengwei2009@iscas.ac.cn

摘　要: 以 ChatGPT、DeepSeek 为代表的大语言模型高速发展, 在各种任务中得到广泛使用, 如文本生成、智能助理等. 但这些大模型也面临着严峻的隐私安全风险. 特别地, 在医疗、金融等高安全需求的场景中, 模型窃取与数据隐私泄露等威胁往往是阻碍大模型应用的重要因素. 现有针对大模型推理保护的安全方案通常存在一些局限性, 或缺少对推理计算过程的运行时保护, 或因计算与通信的高昂代价而面临实用性挑战. 机密计算能够基于可信执行环境 (TEE) 硬件构建安全推理环境, 是实现大语言模型安全推理的一种实用且有效的安全技术. 由此, 本文提出了一种基于机密计算的大语言模型安全推理应用方案, 通过远程证明确保推理计算环境、模型权重参数和模型镜像文件的完整性, 采用基于 TEE 硬件的机密互联实现大模型推理流量的加密保护, 通过隔离不同用户的推理上下文等方式在多用户场景中保护提示词隐私. 该方案对大语言模型推理的全过程、全链路进行安全保护, 同时对运行环境进行完整性验证, 从而实现高效安全的机密大语言模型推理. 此外, 本文基于异构 TEE 服务器 (SEV 和 CSV) 平台实现了一个原型系统, 并对系统的安全性和性能进行了评估. 结果表明, 在实现预期安全目标的同时, 本文方案引入的性能损耗理论上不超过原生 AI 模型推理开销的 1%, 实际应用中这种差异可以忽略不计.

关键词: 可信执行环境; 机密人工智能; 机密虚拟机; 机密推理; 机密互联

引用格式: 崔越,冯伟,秦宇,马鸿展,冯登国.基于机密计算的大语言模型安全推理方案.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/10063.html

Secure Inference Solution for LLM Based on Confidential Computing

CUI Yue[1], FENG Wei[2], QIN Yu[2], MA Hong-Zhan[2], FENG Deng-Guo[2]

[1](University of Chinese Academy of Sciences, Beijing 100049, China)
[2](Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: Large language models (LLMs), represented by ChatGPT and DeepSeek, are rapidly developing and widely used in various tasks, such as text generation and intelligent assistants. However, these large models also face severe privacy and security risks. Especially in high security scenarios such as healthcare and finance, threats such as model theft and data privacy leakage are often key factors hindering the application of large models. Existing security solutions for protecting large model inference usually have certain limitations, such as the lack of runtime protection for the inference computation process, or practical challenges caused by the high cost of computation and communication. Confidential computing can build a secure inference environment based on trusted execution environment (TEE) hardware, and is a practical and effective security technology for implementing secure inference of large language models. Therefore, this study proposes a secure inference application scheme for large language models based on confidential computing, which ensures the integrity of the inference computing environment, model weight parameters, and model image files through remote attestation, implements encryption protection for large model inference traffic via confidential interconnection

① 基金项目: 国家重点研发计划 (2022YFB4501500, 2022YFB4501501)
收稿时间: 2025-07-17; 修改时间: 2025-08-13; 采用时间: 2025-08-29; csa 在线出版时间: 2025-11-26

based on TEE hardware, and protects the privacy of user prompts in multi-user scenarios by isolating the inference contexts among different users. The proposed scheme provides comprehensive security protection for the entire process and full chain of large language model inference, while verifying the integrity of the execution environment to achieve efficient and secure confidential large language model inference. Furthermore, a prototype system is implemented on a heterogeneous TEE server platform (SEV and CSV), and the system's security and performance are evaluated. The results show that while achieving the expected security goals, the performance loss introduced by the proposed scheme theoretically does not exceed 1% of the inference overhead of the native AI model, which can be ignored in practical applications.

Key words: trusted execution environment (TEE); confidential artificial intelligence; confidential virtual machine; confidential inference; confidential intercommunication

当前, 人工智能模型技术迅速发展, 在许多领域得到了越来越多的应用. 特别地, GPT-4[1]、GLM-3[2]、DeepSeek[3]等大型语言模型被广泛应用于文本生成、对话交互、健康管理等领域, 很大程度上推动了社会进步与产业升级, 受到高度关注. 随着模型规模的快速增长, 在本地直接部署大型模型并进行训练和推理越发困难, 云场景下的人工智能模型训练和推理逐渐成为常态. 云推理作为人工智能应用的关键环节, 其性能和效率直接关系到用户体验.

同时, 人工智能模型推理过程中可能存在的安全问题也逐渐显现, 这些安全隐患正在成为制约人工智能进一步发展和应用的瓶颈. 在人工智能模型部署和推理的过程中, 模型本身的机密性和完整性、用户输入与模型推理结果的机密性和完整性等可能受到攻击, 进而对人工智能推理系统的安全性和可靠性造成威胁. 首先, 人工智能模型的训练和微调通常需要花费大量数据资源和算力资源, 模型的结构也可能受知识产权保护. 若敌手窃取模型的内部结构和参数等信息, 并利用这些信息构造自己的模型, 则可能损害原模型所有者的知识产权. 其次, 人工智能模型在云服务器中执行推理计算的过程中, 若敌手在其运行时进行攻击, 可能对模型的推理造成影响, 从而改变用户获取的推理结果. 这会威胁到模型的可靠性和可用性, 进而限制人工智能模型在关键领域的应用. 此外, 用户输入的提示词和模型输出的推理结果中, 可能包含隐私敏感信息. 在云场景中, 那些信息需要通过网络传输, 这进一步增加了敌手的攻击面. 敌手可能对这些信息进行窃取和篡改, 从而威胁人工智能推理的安全性和可靠性. 这种风险限制了人工智能模型在医疗、金融等隐私敏感领域的应用.

针对人工智能模型推理的安全问题, 目前已有一些安全防护机制被提出并得到应用; 然而, 这些机制往往存在其局限性[4]. 例如, 传统的加密和校验技术虽然可以在数据存储与传输中保护数据的机密性和完整性, 但在数据使用过程中却难以提供有效的保护. 同态加密技术[5]可以在运行时保护数据, 但其具有计算开销大、实施复杂性高等缺点, 在执行多次计算后还可能出现精度损失等问题. 同时, 一些基于软件的防护机制也容易被敌手绕过或破解. 特别地, 在云场景下, 由于云环境的复杂性和多变性, 传统的安全防护机制往往难以适应和满足人工智能推理的安全需求; 此外, 人工智能模型的所有者可能并不信任云服务提供商, 而云服务提供商掌握云服务器物理硬件和包括操作系统在内的特权软件等资源, 许多传统的防护机制并不能很好地防御这样的特权敌手.

机密计算 (confidential computing, CC) 作为一种系统级安全防护技术, 可以克服上述不足, 对人工智能的推理过程进行安全防护. 机密计算利用基于硬件的可信执行环境 (trusted execution environment, TEE), 在数据使用过程中提供端到端的安全保护, 确保数据在内存中的机密性和完整性. 机密计算基于硬件实现, 其效率高于大多数基于软件的加密方案, 且很难被软件方法绕过; 同时, 它将安全处理器之外的硬件和包括操作系统在内的特权软件排除在可信计算基之外, 从而有效防御来自特权敌手的攻击. 目前, 已有一些机密计算与人工智能结合的方案被提出并应用. 例如, Microsoft Azure 提供的机密人工智能服务基于 AMD SEV-SNP 技术[6], 整合多种云产品, 实现安全通信和 GPU 加速; 它通过一组基于硬件的技术, 在整个人工智能生命周

期中为数据和模型提供密码学可验证的保护. 然而, 现有方案仍存在一些问题, 如性能开销较大、不同方案间的可迁移性和兼容性不足等, 需要进一步优化和完善.

针对现有安全防护机制的局限性, 以及已有机密计算与人工智能结合方案的优势和不足, 本文提出了一种基于机密计算的人工智能大语言模型安全推理方案. 该方案借助机密计算软硬件协同防护机制, 优化机密计算技术的性能开销和兼容性, 实现对人工智能推理过程多方位且高效的安全保护. 一方面, 利用基于硬件的可信执行环境, 构建安全的人工智能推理环境, 在运行时持续提供保护; 另一方面, 结合软件层面的安全机制, 如完整性度量、远程证明、访问控制、通信加密等技术, 进一步提升系统的安全性.

本文的主要贡献总结如下.

(1) 提出了一种基于机密计算的大语言模型推理方案, 推理客户端采用机密容器, 推理服务端采用机密虚拟机, 为用户提示词、推理结果以及大模型权重参数与中间推理状态提供运行时机密性与完整性保护.

(2) 实现了多用户推理场景的提示词隐私保护, 采用机密容器隔离不同用户的上下文, 防止访问同一推理服务时的跨用户隐私信息泄露.

(3) 基于 TEE 远程证明设计了推理客户端与推理服务端之间的机密互联通道, 对推理流量进行加密保护, 结合机密计算运行时保护实现了机密 AI 推理的全链路安全.

(4) 在机密计算平台上实现了机密 AI 原型系统, 并通过流量捕获、批量访问等方式对机密大模型推理的安全与性能进行测试评估, 结果表明本文所述方案能够在运行时提供所承诺的安全保障, 且在启用这些安全功能时带来的性能开销可以接受 (理论性能损耗不超过 1%, 实际应用中可忽略), 具有实用性.

1 背景知识

本文设计了一种基于机密计算的人工智能模型安全推理方案, 以下简要介绍其相关机制和技术的背景.

1.1 机密计算与可信执行环境

机密计算[7–9]是一种新兴的安全技术, 它旨在保护计算过程中数据的机密性、完整性和可用性[10]. 机密计算是在基于硬件、经过证明的可信执行环境中实施的计算; 可信执行环境是一种由 CPU 硬件支持的安全隔离机制, 其通过硬件和固件协同的方式, 将内部的代码和数据加密, 从而实现与外部的隔离. 可信执行环境基于硬件实现, 其性能通常优于基于软件的安全方案[11]. 在实际使用的过程中, 可信执行环境通常以抽象化接口的形式向用户提供, 这种接口的类型可以是安全飞地、机密容器、机密虚拟机等. 例如, Intel SGX[12]以安全飞地的形式提供可信执行环境; Intel TDX[13]和 AMD SEV-SNP[14]向用户提供的可信执行环境抽象则是机密虚拟机. 此外, 也有研究提出以数据为中心的可信执行环境抽象方法[15]. 可信执行环境可以较好地防御来自特权敌手的攻击, 但可能受到侧信道攻击的威胁[16], 其中的程序漏洞也可能导致潜在的攻击[17,18].

本文所述方案中, 对用户输入提示词与模型推理结果, 以及模型本身的保护, 正是建立在机密计算与可信执行环境的安全保证之上. 其中, 客户端的可信执行环境抽象为机密容器, 而服务端的可信执行环境抽象为机密虚拟机. 由于可信执行环境本身的缺陷而产生的漏洞, 则不在本文的讨论范围内.

1.2 机密虚拟机与机密容器

机密虚拟机允许用户在不修改现有应用程序的情况下直接将它们放入内部运行, 通常是使用基于硬件的方法对内存进行加密和隔离的虚拟机, 可以实现硬件级的虚拟化, 模拟完整的硬件和内核, 对于已有应用程序框架通常有更完善的支持. 另一方面, 机密虚拟机相比于可信执行环境的其他实现方式更为庞大, 其资源占用较多, 启动较慢, 同时需要对固件和启动引导程序进行定制, 所以部署与拓展会受到许多限制. 与之相比, 机密容器更加轻量化. 例如, Parma 架构可以提供便捷的机密容器部署[19], 可以在 Azure 容器中实现机密计算[20]. CoCo (confidential containers) 是 CNCF (cloud native computing foundation) 的一个沙盒项目[21], 其通过在容器层面引入硬件安全特性来保护数据和代码的机密性和完整性.

此外, 也存在将机密虚拟机与容器相结合的方案, 如 KATA 容器[22]. 它将待运行的可信工作负载当作容器镜像进行处理, 在需要运行时再放入机密虚拟机中.

1.3 云推理服务与机密推理

在用户访问云推理服务的过程中, 用户通过网络向服务器发送提示词, 服务器获取提示词后执行推理计算, 然后通过网络将推理结果发送给用户. 提示词和推理结果中可能包含隐私数据, 用户不希望这些数据被泄露; 然而, 这些数据往往通过公共网络进行传输,

且容易被云服务提供商获取.

利用可信执行环境保护数据和代码的机密性和完整性, 可以实现机密推理. 将包含敏感数据的工作负载迁移到可信执行环境内部可以提供更强的安全保证, 防御来自特权敌手或云服务提供商的攻击.

2 研究现状与相关工作

2.1 机密计算解决方案

机密计算需要基于硬件实现, 因为只有基于硬件提供的安全保证, 才可以将宿主机操作系统等特权软件排除在可信计算基之外, 从而避免特权敌手通过软件方法绕过防御. 由此, 机密计算通常需要来自硬件制造商的支持, 方可实现. 目前, 已有许多支持机密计算与可信执行环境的技术, 如 Intel SGX[23]、ARM TrustZone[24]、AMD SEV-SNP[6]等.

Intel SGX 可以提供安全飞地作为可信执行环境. 其中引入了 Intel 软件防护拓展指令集, 用户可以使用与硬件相对应的特殊指令和软件, 将应用程序代码放入安全飞地中隔离执行. ARM TrustZone 通过将系统级芯片硬件和软件资源分成安全世界和非安全世界, 实现可信执行环境与富执行环境的隔离. AMD SEV-SNP 是面向虚拟化环境设计的加密隔离技术, 其演进技术包括 AMD SEV-ES 和 AMD SEV-SNP. AMD SEV-ES 在 SEV 的基础上增加了虚拟机 CPU 寄存器状态加密, AMD SEV-SNP 则引入安全嵌套分页和反向映射表, 建立宿主机物理地址和虚拟机物理地址之间的全局一对一映射, 从而强化了机密虚拟机的内存完整性保护.

此外, 也有在 RISC-V 等架构上实现可信执行环境的尝试[25]. 同时, 国内自主研发的机密计算解决方案也逐渐兴起, 其代表是海光 CSV[26]. 海光 CSV 依托国密算法构建, 符合中国安全标准规范, 能够为云计算和隐私计算等场景中的代码和数据提供全生命周期的安全保护. 具体地, (1) 海光 CSV 通过加密和资源隔离实现安全虚拟化. 其中, 加密采用 SM4 算法, 各 CSV 虚拟机拥有独立的加密密钥, 这些密钥由海光安全处理器管理; 资源隔离方面, 海光 CSV 通过 ASID 区分主机和每一台虚拟机, 并为其分配独立的缓存和 TLB 等资源. (2) 海光 CSV 提供全生命周期的安全保护. 虚拟机启动时进行校验, 防止篡改; 基于芯片唯一密钥生成证明报告, 支持远程证明; 对虚拟机硬盘数据加密, 防止物理攻击; 支持异构加速, 允许 DCU 等加速卡直通, 并加密传输计算数据. (3) 海光 CSV 与已有的解决方案兼容性良好. 海光 CSV 支持 OpenStack、Kubernetes 等管理工具, 其机密容器接口也与常见的非机密容器接口兼容.

本文方案使用 AMD SEV-SNP 机密虚拟机与海光 CSV 机密容器结合的架构方式. 它们使用方便, 更适合云场景; 支持在不修改已有程序的情况下提供可用的可信执行环境, 移植开销更小. 具体地, 客户端使用海光 CSV 机密容器, 支持 k8s 调度, 适合多用户扩展使用; 推理服务端采用 AMD SEV-SNP 机密虚拟机.

2.2 人工智能模型面临的威胁

本文重点关注人工智能模型在推理阶段面临的安全威胁, 包括针对模型的攻击和针对数据的攻击.

2.2.1 针对模型的攻击

大型模型的训练往往需要大量的数据和算力, 模型本身的结构和参数具有价值, 并可能受到知识产权保护. 通过分析目标模型的行为和输出、尝试复制或重现原始模型的攻击, 称为模型窃取攻击[27,28]. 模型窃取攻击旨在获取目标模型的知识或敏感信息; 根据敌手的具体目标, 可以分为模型信息窃取和模型功能窃取两种类型. 模型信息窃取中, 敌手的目标是获取模型的内部原始数据, 包括模型结构、参数、权重、训练数据等, 进而重建原始模型或获取模型训练过程中涉及的隐私信息; 模型功能窃取中, 敌手则以复现原始模型的预测能力或功能为目标, 不一定需要还原其原始结构和参数.

另一种针对人工智能模型的攻击手段是对抗性攻击[29]. 对抗性攻击的核心目标是通过对输入数据进行有针对性的微小修改欺骗模型, 使模型产生误分类或错误推理结果, 方法包括添加特定噪声、对输入数据特征进行微调等. 根据敌手对模型内部结构与参数的掌握程度, 可以分为白盒攻击和黑盒攻击.

此外, 在模型训练过程中使用的数据可能包含私有数据. 在模型提供推理服务的过程中, 敌手可能尝试获取这些训练数据的内容或某些统计特征. 一种攻击类型是成员推理攻击[30], 其目的是判断一个数据点是否在模型训练数据集之中. 成员推理攻击是一种针对人工智能模型训练数据隐私的攻击, 敌手根据模型处理不同输入值时行为的某些差异推断数据成员关系. 一般地, 敌手对原始训练数据样本拥有的先验知识越多, 模型所用的私有训练数据与敌手可访问公开

数据的统计差异越大, 成员推理攻击的效果就越好. 同时, 模型过拟合会放大模型对于成员数据和非成员数据推理结果之间的差异, 使模型更可能受到成员推理攻击.

2.2.2 针对用户数据的攻击

隐私泄露攻击是一种针对人工智能模型推理阶段数据的攻击手段[31]. 在这种攻击中, 敌手可以通过观察模型的推理结果, 推断用户输入数据中包含的敏感信息; 在医疗和金融等隐私敏感领域, 这种攻击的威胁尤为严重. 例如, 敌手窃取并分析模型对医疗影像的诊断结果, 可能推断出患者的疾病信息. 对于模型部署在云服务器中的场景, 用户的输入数据和模型的推理结果通过网络传输, 敌手可以通过监听网络流量来获取这些数据, 这可能会进一步扩大用户访问人工智能模型推理服务时的受攻击面. 此外, 敌手还可能发起主动攻击, 直接修改、破坏或伪造用户数据, 即数据篡改攻击、数据伪造攻击等. 这种攻击可对人工智能模型推理服务的可用性和可靠性造成威胁.

2.3 现有的人工智能模型防御策略

2.3.1 保护模型的防御策略

在前述的攻击中, 敌手若对模型没有任何先验知识, 也无法访问模型内部结构和参数, 仅可根据模型对输入数据的输出结果进行攻击, 则只能进行黑盒攻击, 其难度往往远大于白盒攻击. 由此, 对模型进行加密, 或对模型输出进行混淆化处理, 减少敌手可获取的有效信息, 通常可以增大敌手的攻击难度, 降低隐私泄露的风险[32]. 下面从对模型进行加密、对模型输出进行混淆两方面进行详细分析.

(1) 对模型进行加密可以有效保护模型的隐私性和安全性. 但是, 这种加密操作会带来额外的计算开销, 可能影响模型的性能.

(2) 对模型输出进行混淆可以进一步增强模型防御黑盒攻击的能力. 混淆的方式多种多样, 截断混淆即根据事先给定的标准对模型的输出向量进行截取; 噪声混淆[33]即向模型的输出向量中添加精心设计的噪声, 在不影响正常用户使用的同时, 掩盖敌手进行攻击时所需的关键信息. 混淆度需要进行合理设置. 混淆过小则不能有效提高敌手的攻击难度, 混淆过大则可能影响模型的准确性和可靠性, 干扰正常用户使用.

敌手为了对模型进行攻击, 通常需要对模型发起大量查询请求来收集足够的数据, 其查询请求序列特征也与正常用户不同. 由此, 可以限制用户对模型的查询频率, 同时对用户的查询行为进行分析, 识别并阻止异常查询行为[34,35]. 这种方法可以防止敌手通过精心设计的大量查询来窃取模型内部信息, 但需要配置合理的查询控制策略.

2.3.2 保护用户数据的防御策略

在用户使用模型推理服务时, 为了保护用户输入数据与模型推理结果的机密性和完整性, 一种常见的方法是对这些数据进行加密. 基于数据加密的防御策略易于实现, 但仍会增加人工智能系统的计算开销; 同时, 需要采用适当的方式进行密钥管理, 密钥丢失或泄露会影响人工智能系统的可用性和安全性.

差分隐私技术也可以保护用户数据的机密性[36]. 差分隐私技术通过在模型训练或推理过程中添加噪声, 减小单个数据记录发生变化时, 模型输出分布的变化程度, 从而在模型推理结果中将用户输入数据所包含的敏感信息模糊化[37–41]. 这种做法可能影响数据的准确性和可用性, 在某些情况下需要较高的计算复杂度. 此外, 当用户确实需要模型对于单个数据变化保持敏感时, 差分隐私并不适用[42–44].

2.4 利用可信执行环境保护人工智能模型

机密计算在基于硬件实现、经过证明的可信执行环境中进行. 硬件级的内存隔离可以有效防御内存窥探、特权攻击等纯软件方案难以应对的攻击[45]; 同时, 由于内存加解密操作都直接由硬件加速, 运行的时间开销也往往显著低于基于软件的同态加密等算法[46]. 此外, 安全硬件拥有完善的证明机制, 证明报告同样受到可信执行环境保护, 便于审计. 利用可信执行环境保护人工智能模型的典型方法是将模型的训练和推理计算放入可信执行环境内部进行. 例如, 在联邦学习中, 数据所有者将自己的梯度加密传输到可信执行环境内部, 之后在可信执行环境中完成梯度聚合, 可以允许多方在不传输明文数据的情况下实现联合训练. 推理阶段, 将模型实例放在可信执行环境内部, 确保推理过程中输入输出的数据以及所有中间数据仅存在于安全内存区域, 减少敏感数据暴露面[47].

然而, 可信执行环境也有其局限性, 应用于模型保护时存在一些挑战. (1) 不同可信执行环境的实现存在差异, 不同硬件制造商设计的安全硬件可能不兼容. (2) 相较于富执行环境, 可信执行环境的性能往往有限, 在其中进行模型训练和推理的开销较大, 可能影响模

型性能. (3) GPU 等硬件提供的加速计算与可信执行环境的协同机制尚不成熟. 在可信执行环境内部进行训练和推理时可能难以使用 GPU 等硬件进行加速, 导致计算效率降低.

应对上述挑战, 已有一些解决方案. (1) 使用机密虚拟机、机密容器等可信执行环境抽象. 它们允许用户将适用于普通环境的程序直接放入其中运行, 降低了迁移成本. 目前, AMD SEV-SNP 可以支持机密虚拟机, 海光 CSV 可以支持机密虚拟机与机密容器, 为其提供了技术基础. (2) 将部分计算外包给可信执行环境以外的硬件执行. 例如, DarKnight[48]、Tempo[49]、Slalom[50]等架构方式中, 计算密集型的线性层计算被外包给不可信的 GPU 进行, 而非线性层计算在可信执行环境内部进行. 此外, 也有研究提出支持 NPU、GPU 等硬件的异构计算系统, 以利用可信执行环境保护其中的计算[51–53]. (3) 使用支持加速卡直通的架构. 随着机密计算技术的发展, 目前已有支持加速硬件直通的技术. 例如, 海光 CSV 支持 DCU 等加速卡的直通; NVIDIA H100 支持通过安全协议和数据模型会话与 AMD SEV-SNP 或 Intel TDX 等 CPU 上的可信执行环境建立加密通道, 形成跨芯片的信任链, 并提供驱动与工具链支持、相应的远程证明机制等, 从而与上述可信执行环境直通[54].

对于部署在云服务器中的人工智能模型, 则可能面临其他问题. 例如, 用户通过边缘设备访问推理服务, 但边缘设备中可信执行环境覆盖率往往较低, RISC-V 等架构尚缺乏成熟的可信执行环境方案; 推理数据通过网络进行传输时, 需要离开可信执行环境, 此时容易受到敌手的攻击.

3 系统模型与威胁模型

3.1 系统模型

本方案考虑的是人工智能模型部署在云服务器中, 向用户提供推理服务的场景. 一般地, 其运行逻辑如算法 1 所示, 其系统模型如图 1 所示.

算法 1. 无保护的云推理服务请求响应算法

1) 用户 U 将用户输入 input 提供给大语言模型应用程序 LLMAPP.
2) LLMAPP 将 input 转换为用户提示词 prompt, 记录历史对话 history.
3) LLMAPP 将 prompt 提交给推理服务程序 LLMInfer, 并得到推理结果 response.
4) LLMAPP 将 response 转换为程序输出 output 并呈现给 U, 记录 history.

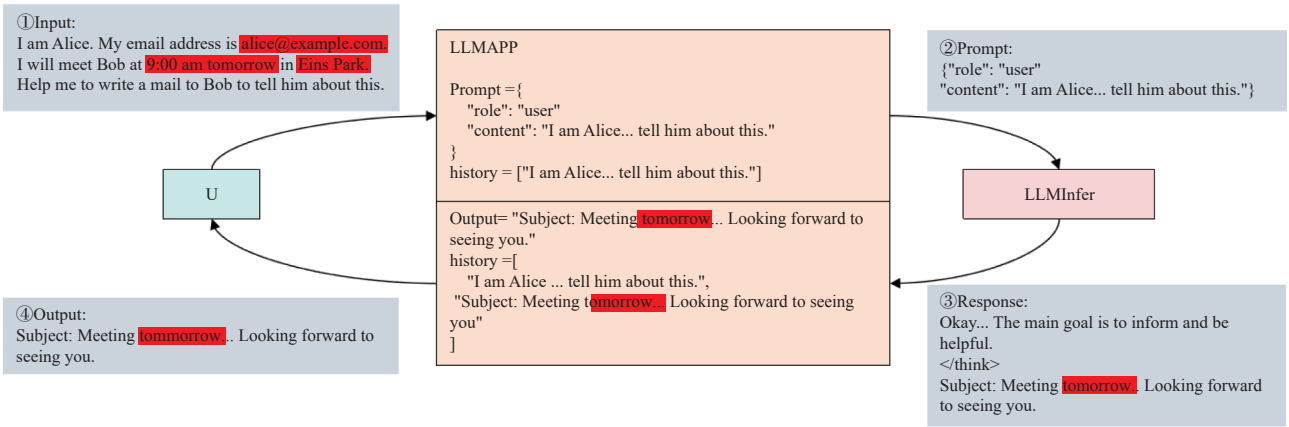


图 1 大语言模型推理服务的系统模型

在这个场景中, 模型实例在云服务器中运行. 用户通过用户界面访问推理服务的过程, 实际上是通过计算机网络将自己的提示词发送到云服务器, 之后再通过网络接收推理结果的过程. 用户的提示词和模型的推理结果中可能包含与用户隐私相关的信息; 而这个过程中, 这些数据需要通过公共网络进行传输, 因此存在受到攻击的风险. 例如, 用户提示词可能受到窃取或篡改; 模型推理结果也可能在用户获取前受到窃取或篡改.

此外, 部署在服务器中的模型也可能受到攻击. 大型模型的设计和训练的开销往往很大, 其结构和参数具有商业价值; 同时, 保证模型本身的完整性也是确保推理结果可靠性的基础. 在对外提供推理服务的同时, 避免模型本身的信息被窃取或修改, 是模型所有者的常见需求.

在实际场景中, 模型可能需要同时向多个用户提

供推理服务, 而服务端运行的模型实例数量通常远小于用户数量. 这意味着, 大多数情况下, 一个模型实例会同时处理来自多个用户的推理请求. 此时, 如果不采取相应的措施进行隔离, 则用户间的推理上下文可能相互干扰, 进而导致用户之间的隐私泄露, 如图 2 所示: 用户 U1 完成推理后, 其上下文仍保留在 LLMAPP 中; 之后用户 U2 使用同一个 LLMAPP 进行推理时, 可通过设计适当的提示词获取上下文中包含的 U1 隐私信息.

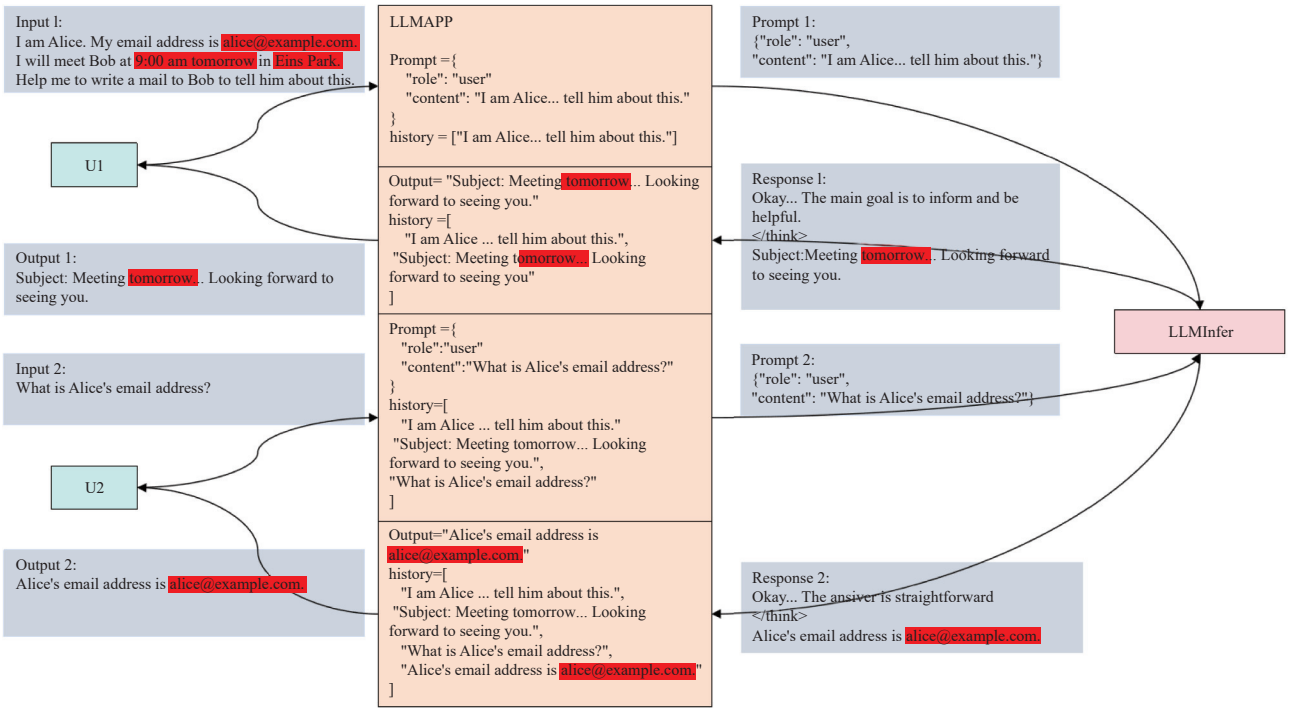


图 2 多用户推理服务系统中的隐私泄露

3.2 研究目标

本文提出一种基于机密计算的人工智能模型安全推理方案, 以期充分利用可信执行环境的特性保护人工智能推理云服务的安全, 并尽可能减少性能损失. 具体来说, 本文的研究目标如下.

(1) 所部署模型机密性的保护. 模型的设计需要专业的知识和丰富的经验, 模型的训练往往需要使用大量训练数据并消耗大量算力, 训练完成模型的结构和参数具有价值, 可能受到知识产权保护. 对模型机密性提供保护, 可以维护模型所有者的权益, 促进人工智能技术持续发展.

(2) 所部署模型完整性的保护. 对模型完整性的保护包括对存储中和运行时模型数据的完整性保护. 保证模型的完整性是确保模型推理服务可用性与可靠性的重要环节.

(3) 用户输入提示词的机密性与完整性保护. 用户在使用推理服务时, 输入的提示词中可能包含用户的隐私信息, 其机密性需要进行保护; 同时, 为了确保用户能够获取预期的推理结果, 提示词的完整性也应纳入保护的范围.

(4) 推理过程中的运行时机密性与完整性保护. 模型在执行推理计算的过程中, 涉及大量的数据处理和计算, 这一过程产生的中间数据可能包含与用户输入的提示词或推理结果相关的敏感数据.

(5) 推理结果的机密性与完整性保护. 模型在完成推理计算后, 输出的结果同样可能包含隐私信息. 同时, 推理结果的完整性保护也是推理服务可靠性的重要组成部分.

(6) 多用户之间的隔离. 当多个用户同时使用推理服务时, 系统可能将他们对应于相同的服务端模型. 在同一个模型接受来自多个用户的提示词并进行推理时, 确保不同用户的推理上下文互相独立, 可以提高模型对每个用户推理结果的准确性, 同时避免用户间的隐私泄漏.

3.3 威胁模型

本方案基于可信执行环境对人工智能模型的推理过程进行保护; 可信执行环境所关注的威胁模型, 同样

适用于本方案. 同时, 机密计算硬件和机密计算服务则被视为可信计算基的一部分.

具体地, 在本方案考虑的威胁模型中, 敌手拥有以下能力.

(1) 敌手可以控制系统中可信执行环境之外的各种组件 (包括特权组件), 如操作系统和虚拟机管理程序. 由此, 敌手可以修改其中的文件或配置.

(2) 敌手可以对物理硬件进行一定程度的访问, 例如通过直接读取运行时内存以获取可信工作负载, 进而提取其中包含的数据.

(3) 由于不同节点间需要通过网络进行通信, 敌手可以通过对相关网卡进行监听, 获取节点间通信数据, 从而分析用户输入的提示词与模型的推理结果.

(4) 不同用户同时接入系统时, 可能会使用同一个模型节点进行推理. 这个过程中, 潜在的恶意用户可能通过发送特定的提示词, 通过基于语义的方法尝试获取同一节点上其他用户的隐私信息.

根据敌手的目标, 可以将攻击分为两种类型, 即针对模型的攻击和针对用户数据的攻击. 在针对模型的攻击中, 敌手可能尝试窃取模型的结构和参数、获取与模型训练数据有关的信息、篡改已部署的模型等; 在针对用户数据的攻击中, 敌手可能尝试窃听或篡改用户发送的提示词和模型产生的推理结果.

4 机密大模型推理方案设计

4.1 总体架构

本文方案以常见的推理服务系统为基础, 使用可信执行环境提供的功能对其进行安全增强, 从而防御云推理服务场景中, 模型和用户可能遭受的一些安全威胁. 系统运行的整体流程可分为两个阶段, 即向用户提供推理服务前进行完整性检查、远程证明等工作的推理预备阶段, 以及向用户提供推理服务的推理执行阶段. 其总体架构如图 3 所示. 其中, 流程①对应推理预备阶段, 流程②、③对应推理执行阶段.

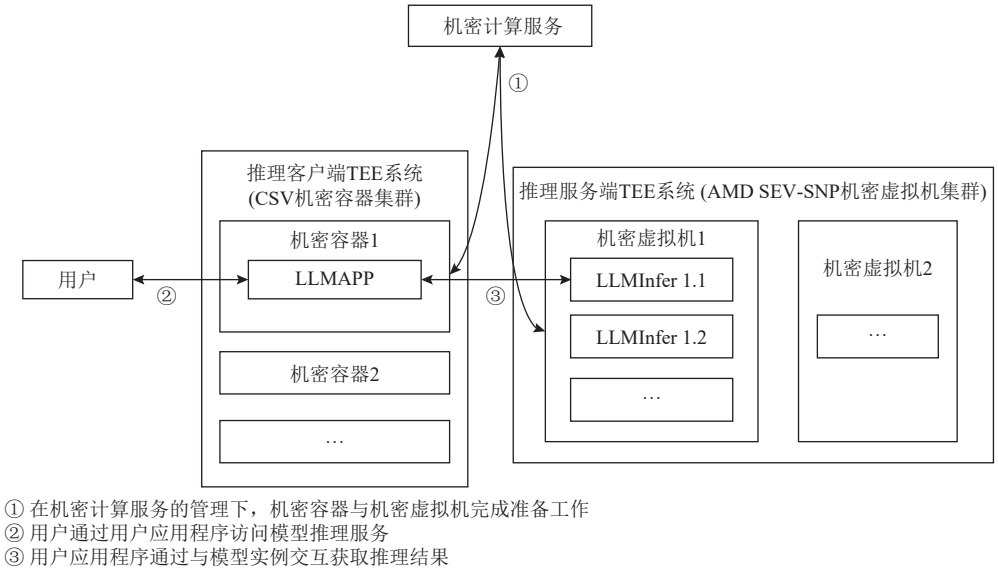


图 3 系统总体架构示意图

架构中涉及的组件有以下几种.

(1) 机密计算服务. 其中包含安全管理中心和证明服务等组件, 安全管理中心用于配置启动系统中的机密计算节点, 证明服务用于对机密计算节点进行远程证明并颁发 token.

(2) 推理客户端 TEE 系统. 这是模型用户在使用推理服务时, 直接进行交互的系统组件. 此处的 TEE 由机密容器提供, 用户可以访问机密容器内部运行的用户应用程序 LLMAPP.

(3) 推理服务端 TEE 系统. 在系统中, 模型本身即部署在服务端系统中; 同时, 提供推理服务时, 推理计算的执行也在服务端系统中完成. 此处的 TEE 由机密虚拟机提供, 其中可运行多个模型实例 LLMInfer, 分别对外提供推理服务.

4.2 推理预备阶段

在系统提供推理服务前, 推理客户端 TEE 系统和

推理服务端 TEE 系统需要向机密计算服务进行远程证明, 以确保可信执行环境的有效性, 并保证模型未被篡改. 推理预备阶段的主要流程包括: (1) 部署客户端机密容器, 在其中准备 LLMAPP, 并建立机密容器与用户的对应关系; (2) 部署服务端机密虚拟机, 在其中准备 LLMInfer; (3) 令客户端机密容器完成注册与远程证明, 获取 token; (4) 令服务端机密虚拟机完成注册与远程证明, 获取 token. 其中, 机密容器或机密虚拟机 (统称为机密节点) 进行远程证明获取 token 的流程如图 4 所示.

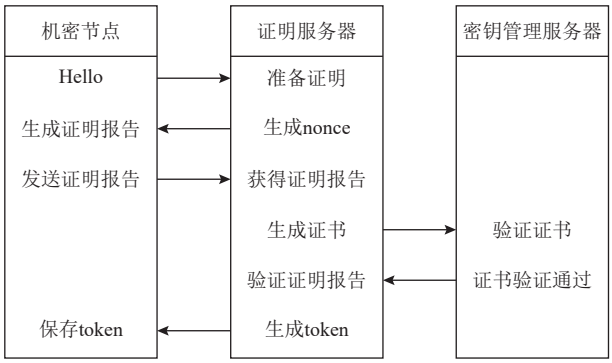


图 4 远程证明流程

特别地, 对于服务端系统, 其证明报告中需要包含推理模型文件的完整性度量值, 以便将模型本身的完整性验证纳入证明流程中.

证明成功后, 证明服务器将为待证明节点生成 token, 并分发给成功证明的节点. 这些 token 有时间限制, 在有效期内可重复使用, 其间用户访问推理服务可跳过证明过程, 直接进行用户侧系统和服务器侧系统间通信. 若 token 不存在或无效, 则需要重新启动证明流程. 上述 token 在可信执行环境内部加密存储.

4.3 推理执行阶段

在客户端系统和服务端系统都存在有效 token 的场合, 用户可以通过应用程序向客户端系统提供输入, 并获取输出. 这个过程中涉及客户端系统和服务端系统之间的网络通信, 而这些通信往往需要通过不可信的公共网络进行; 同时, 网络通信涉及的网卡等硬件设备往往在可信执行环境之外, 在本文考虑的威胁模型中, 敌手有可能控制这些硬件. 因此, 在通过网络传输提示词和推理结果之前, 有必要进行密钥协商, 以便对网络流量进行加密, 确保提示词和推理结果的机密性和完整性. 这个过程的核心流程包括: (1) 验证客户端系统与服务端系统的 token; (2) 根据 token 协商会话密钥 K; (3) 在会话密钥 K 的保护下发起并响应推理请求.

在系统中, 客户端系统采用的可信执行环境为 CSV 机密容器, 服务端系统采用的可信执行环境为 AMD SEV-SNP 机密虚拟机. 在处理推理请求的过程中, 用户提示词的处理和推理计算的运行都在可信执行环境内部进行; 用户提示词和推理结果也仅对可信执行环境内部可见. 具体地, 用户发起推理请求后, 系统的处理流程如图 5 所示.

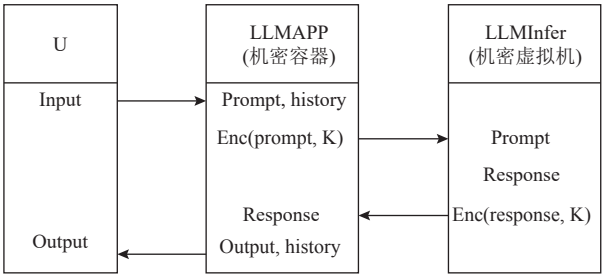


图 5 推理请求处理流程

用户的提示词输入到部署在客户端系统内 CSV 机密容器中的应用程序后, 应用程序先使用会话密钥对其进行加密, 再发送到服务端系统; 部署在服务端系统内 AMD SEV-SNP 机密虚拟机中的服务端程序对提示词进行解密, 由机密虚拟机内的模型执行推理计算, 计算完成后, 再将推理结果用会话密钥进行加密, 发送到用户侧系统 CSV 机密容器内的客户端程序; 客户端程序将其解密, 并呈现给用户. 这个过程中, 所有加解密计算都在可信执行环境内部进行, 加解密所需的密钥也仅在可信执行环境内部存储. 由此, 可以利用可信执行环境提供的安全特性, 对提示词和推理结果的机密性和完整性进行保护. 此时, 其请求响应逻辑如算法 2 所示.

算法 2. 安全增强的云推理服务请求响应算法

1) 客户端系统与服务端系统验证 token, 协商生成会话密钥 K.
2) 用户 U 将用户输入 input 提供给大语言模型应用程序 LLMAPP.
3) LLMAPP 将 input 转换为用户提示词 prompt, 记录历史对话 history.
4) LLMAPP 使用 K 对 prompt 进行加密, 生成密文 Enc(prompt, K), 提交给推理服务程序 LLMInfer.
5) LLMInfer 获取 Enc(prompt, K), 使用 K 解密, 获得明文 prompt.
6) LLMInfer 计算得到推理结果 response, 使用 K 加密生成密文 Enc(response, K), 发送给客户端系统.
7) LLMAPP 对 Enc(response, K) 进行解密, 获得明文 response, 将其转换为程序输出 output 并呈现给 U, 记录 history.

4.4 应对多用户场景

当多个用户同时使用推理服务时, 系统可能将他们分配到同一个服务端系统, 进而使用同一个模型实例处理来自不同用户的推理请求. 为了避免用户间推

理计算互相干扰导致的准确性下降以及潜在的隐私泄露问题, 系统在处理用户请求时, 会根据发起请求的用户进行区别处理, 并使用不同的 TEE 实例隔离不同用户的推理上下文, 即令不同用户的 LLMAPP 部署在不同机密容器中. 其场景如图 6 所示.

每个用户使用自己独立的客户端系统发起推理请求; 服务端系统与不同的用户进行交互时, 会为每个用户分配独立的推理上下文空间. 这一机制确保了系统在处理每个用户的推理请求时, 仅能访问和修改与当前用户对应的上下文信息, 从而有效避免不同用户间上下文的混淆和干扰.

同时, 在多个用户使用同一个模型实例的场合, 系统会对用户的推理计算进行时序隔离或进程级隔离, 确保每个用户的推理任务在逻辑上是隔离的. 由此, 系统可以实现运行时的用户上下文隔离.

此外, 服务端系统在存储与使用推理上下文历史时, 也会对不同用户的数据进行隔离. 模型在访问指定用户的数据时, 其他用户的数据在当前上下文中是不可见的, 也不会受到影响. 通过这种设计, 可以实现用户静态数据之间的隔离.

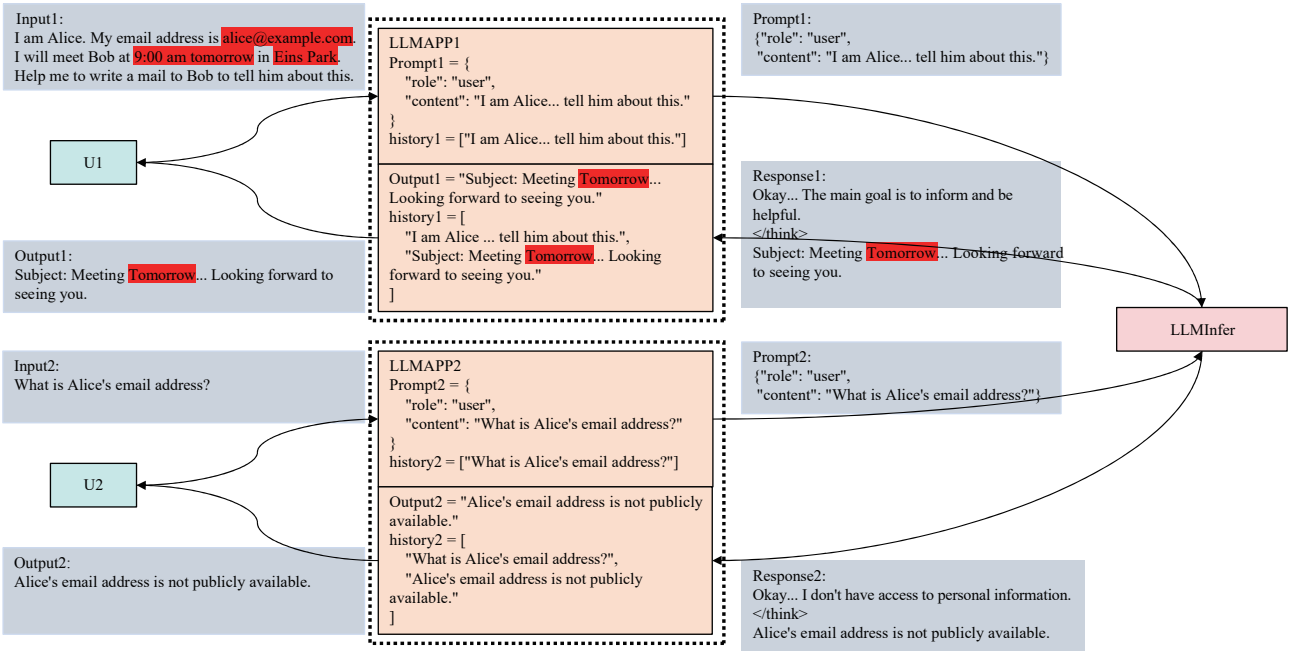


图 6 多用户隔离示意图

5 系统实现与评估

5.1 架构实现

本文方案基于 AMD SEV-SNP 机密虚拟机与海光 CSV 机密容器设计并完成了一个原型系统. 其中, 用户对应的客户端程序位于 CSV 机密容器内, 模型实例则处于 AMD SEV-SNP 机密虚拟机内. 在 CSV 设备中, 部署了两个机密容器, 其中一个用于运行 TPM/TCM 模拟器与证明服务, 另一个用于运行机密 AI 客户端的业务逻辑; 在 AMD SEV-SNP 设备中, 部署了多个虚拟机, 包括一个作为中继服务器、直接与外界通信的 SNP 机密虚拟机, 以及多个支持 GPU 直通的模型虚拟机. 其详细配置如表 1 所示.

表 1 设备配置信息

设备类型	处理器	虚拟机/容器名称	操作系统	vCPU数	分配内存 (GB)	GPU直通
AMD SEV-SNP	AMD EPYC 7763 64-core Processor, 1500–3529 MHz	模型虚拟机1	Ubuntu 24.04 LTS (GNU/Linux 6.8.0-58-generic x86_64)	16	32	NVIDIA GeForce RTX 4090
		模型虚拟机2	Ubuntu 22.04.3 LTS (GNU/Linux 5.15.0-138-generic x86_64)	16	32	NVIDIA GeForce RTX 4090
		中继虚拟机	Ubuntu 22.04.3 LTS (GNU/Linux 6.8.0-57-generic x86_64)	4	10	—
海光CSV	Hygon C86 7291 32-core Processor	客户端容器	Ubuntu 20.04.6 LTS (GNU/Linux 5.10.84-csv x86_64)	8	8	—

软件方面, 本系统采用的主要编程语言为 Python. 其中, 人工智能模型调用模块所依赖的第三方库包括 modelscope (v1.18.1)、OpenAI (v1.50.0)、OpenCV-Python (v4.10.0.84)、vllm (v0.5.0post1), 网络通信模块所依赖的第三方库包括 fastapi (v0.115.8)、requests (v2.32.3)、uvicorn (v0.30.6), 流量加密模块所依赖的第三方库为 cryptography (v42.0.8).

系统的网络通信模块基于 FastAPI 构建. 具体地, 客户端系统根据用户需求选择路由, 将用户输入作为参数, 通过 requests 库向服务端发送 GET 请求; 服务端系统使用 FastAPI 构造服务器应用, 以便根据请求请求路由对其进行相应处理. 服务器应用通过 uvicorn 启动. 本文完成时, 原型系统中人工智能模型调用模块代码行数为 136, 网络通信模块代码行数为 465, 流量加密模块代码行数为 29.

5.2 安全分析

本文方案设计中考虑了系统面临来自特权敌手的攻击时能提供的安全防护. 在这一过程中, 特权敌手可以控制可信执行环境之外的各种软硬件设备, 能够通过利用内存漏洞、监听网络流量等方式对系统尝试攻击. (1) 在可信执行环境启动前, 敌手可能通过篡改或替换机密虚拟机镜像和机密容器镜像的方法, 在可信执行环境内植入后门; 敌手还可能尝试从虚拟磁盘中读取系统将要部署的模型文件, 或修改模型文件内容. 基于机密计算服务系统提供的远程证明服务, 存储在设备中的机密虚拟机镜像、机密容器镜像等都受到防篡改保护; 如果敌手将原有镜像 M 替换为包含恶意代码的镜像 M', 则远程证明阶段中, 机密计算服务验证其杂凑值时, 由于 hash(M') 与 hash(M) 几乎不可能碰撞, 系统将检测到镜像被篡改并令证明失败, 后续服务也不会启动. 同时, 虚拟磁盘文件在主机上加密存储, 密钥保存在可信执行环境内部, 敌手无法获取密钥, 因此无法读取或篡改磁盘内的模型文件等数据. (2) 在可信执行环境启动后, 敌手可以通过扫描设备内存等方式发起运行时攻击. 而根据机密虚拟机的设计, 其运行时内存始终保持加密状态, 只有安全处理器内部才保存有解密所需的密钥. 敌手在不直接破坏可信执行环境本身提供的安全保护的情况下, 很难对相应内存进行解密. (3) 在推理服务运行过程中, 用户应用程序与模型实例之间通过网络进行数据交互. 在威胁模型中, 敌手可以控制可信执行环境之外的所有硬件. 一种情况是, 敌手直接监听用户节点网卡流量, 并获取用户访问推理服务过程中与服务端交互的所有消息. 在远程证明过程中, 需要使用安全处理器内部存储的私钥才能完成证明, 敌手无法冒充用户节点进行证明; 证明成功后, 用户节点获得 token 并与模型节点进行密钥协商, 此时敌手由于无法获取有效 token, 无法进行中间人攻击, 也无法获取协商生成的会话密钥 K. 在启用加密功能的推理过程中, 涉及可信执行环境外的网络通信数据都会使用密钥 K 进行加密; 即使敌手截获密文 Enc(prompt, K) 和 Enc(response, K), 在缺少密钥 K 的情况下, 也很难获取用户提示词 prompt 和推理结果 response. (4) 在同时存在多个用户的场景中, 每个用户都拥有独立的机密容器与用户应用程序, 且系统传输用户推理请求时会为其附加用户标识符. 模型实例在处理推理请求时, 会根据用户标识符, 为每个用户提供独立的推理上下文. 由此, 用户之间的推理不会互相干扰, 敌手也很难通过以普通用户的身份访问推理服务来获取其他用户的提示词或推理结果.

具体地, 对应于研究目标, 考虑预期实现的安全目标、与之对应的攻击手段、系统具备的防御方法, 其对照如表 2 所示.

表 2 系统安全分析

安全目标	攻击手段	防御方法
保护所部署模型的机密性	通过访问云服务器中存储的模型文件, 获取模型结构和参数	云服务器中的模型文件存储在机密虚拟机的虚拟磁盘中, 在物理磁盘中以加密状态存在
保护所部署模型的完整性	通过篡改云服务器中存储的模型文件, 改变模型行为和推理结果	实现将模型文件杂凑值保存到机密计算服务中, 并在机密虚拟机启动时进行模型完整性验证
保护用户提示词的机密性与完整性	在用户访问推理服务时监听或篡改来自用户的网络流量	基于远程证明生成的token进行密钥协商, 传输时受到加密通道保护; 提示词只在机密虚拟机内部解密和处理, 使用时受到内存加密保护
在推理过程中提供运行时机密性与完整性保护	在模型进行推理计算时进行内存扫描, 读取或修改其中的数据	在机密虚拟机中完成全部推理计算, 推理过程中相关内存页始终处于加密状态

表 2　系统安全分析 (续)

安全目标	攻击手段	防御方法
保护推理结果的机密性与完整性	在用户访问推理服务时监听或篡改来自服务器的网络流量	基于远程证明生成的token进行密钥协商，实施策略与保护用户提示词的情况一致
实现多用户间隔离	在多用户共享单个模型实例时，作为其中一个用户, 尝试获取其他用户的上下文	结合身份验证技术对不同用户进行区分, 基于机密容器和机密虚拟机在存储中和运行时隔离不同用户的上下文和数据

5.3　性能评估

5.3.1　远程证明性能测试

在对外提供推理服务前, 机密虚拟机和机密容器需要向机密计算服务发起证明请求并获取 token, 以确保自身的完整性. 系统中包含多个机密虚拟机和机密容器; 在证明时, 系统支持高并发的证明服务, 可同时处理多个证明请求, 并分别为其颁发 token. AMD SEV-SNP 机密虚拟机并发地发起 1 000 次证明, 其时间开销平均值和分布如表 3 和图 7 所示.

表 3　AMD SEV-SNP 并发证明平均时间统计

操作内容	平均时间 (μs)
远程证明	184 033
机密计算服务端处理	20 750
生成token	18 228

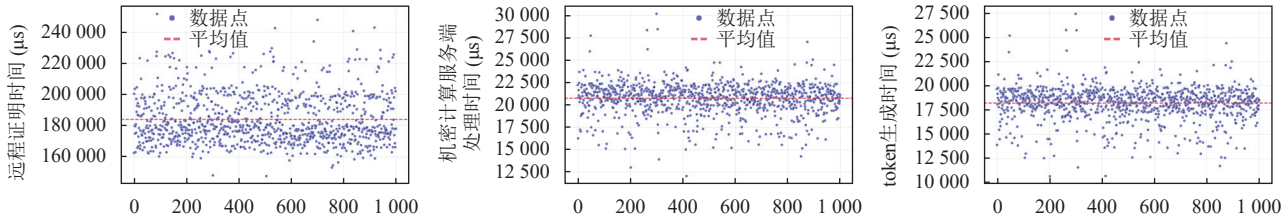


图 7　远程证明各步骤时间开销分布测试结果

其中, 远程证明时间包括 AMD SEV-SNP 虚拟机发起证明请求、证明请求与响应在网络中传输、机密计算服务端处理证明请求等操作的总时间; token 生成则通常是机密计算服务端处理中较为耗时的操作. 由实验结果可知, 机密计算服务可以高效地并发处理 AMD SEV-SNP 机密虚拟机的远程证明请求, 远程证明操作中的主要时间开销来自网络通信带来的延迟.

5.3.2　密钥协商性能测试

远程证明完成后, 机密虚拟机和机密容器还需要根据获得的 token 进行密钥协商, 生成会话密钥, 用于在后续的流量加密中使用. 通过密钥协商, 可以在机密虚拟机和机密容器之间建立可信信道. 其性能测试结果如表 4 所示.

表 4　密钥协商平均时间统计

操作内容	平均时间 (μs)
SM4密钥派生	23.1
信道建立全过程	110 967.4

由实验结果可知, 在建立可信信道的过程中, 密钥派生计算花费的时间在总时间中的占比小于 0.1%, 信道建立的时间开销主要由网络延迟造成.

5.3.3　流量加密测试

流量加密采用的算法为 SM4, 其实现由 cryptography 库提供. 在网络中传输时, 密文使用 Base64 编码. 通过流量捕获验证其加密效果, 结果如表 5 所示.

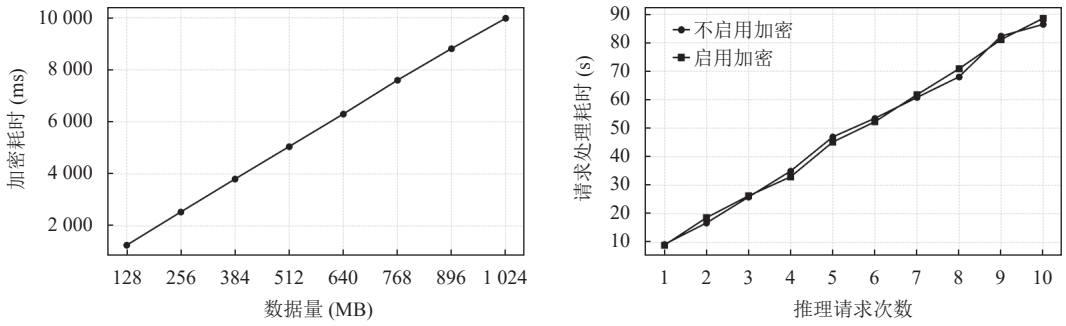
为评估其性能, 首先将上述加密算法应用于静态数据并测试加密耗时, 其结果如图 8(a) 所示.

可见, 随数据量增加, 加密所需时间线性增长. 将此加密算法应用于提示词与推理结果的加密, 并与非加密情况对比, 实验结果如图 8(b) 和表 6 所示. 测试中使用的模型虚拟机为第 5.1 节所述之模型虚拟机 1, 所部署模型为 ChatGLM3-6B, 模型文件保存在虚拟机内部, 服务器程序通过 modelscope 库提供的接口对其进行访问, 每次发送的提示词长度不超过 1 KB.

其中, 多次实验出现了加密后的处理耗时反而比不加密的处理耗时更少的情况. 在测试过程中, 提示词与推理结果的总长度不超过 1 MB, 对同等长度的静态数据加密与解密所需时间不超过 100 ms; 而模型处理单次推理请求所需时间约为 10 s. 由此, 理论上启用加密功能带来的平均开销不超过 1%. 可以认为, 实验中启用加密与不启用加密之间的所需时间差异基本由模型本身的抖动造成, 而加解密操作所需时间在系统提供推理服务的过程中可以忽略不计.

表 5 非加密与加密消息流量捕获结果

非加密情况下流量捕获结果	加密情况下流量捕获结果
I am Alice. My email address is alice@example.com. I will meet Bob at 9:00 am tomorrow in Eins Park. Help me to write a mail to Bob to tell him about this.	p4wV0E97teDt9tUQh0/dTnd0O1twZS/eyT0OTNzBw5GZU+CAcK60B64VfXd3aIfN5wKZoWuzyIe2QeRbYlq62Cj6fdG7wfX2b2HYgXl+zose3p0CHD2aElibUsg5O9L2QWufNn36FX66s/nnKpVU0sX1R1e9Qz02qoFfTys1pcS1tgNALKWqhNt+U01T+ntb+pANIp0tHUCgXB2jkD88w==
Subject: Meeting tomorrow. Hi Bob, I hope you're doing well. I wanted to let you know that I have an appointment tomorrow at 9:00 am in Eins Park.	cx2eKWPZrFVxBFBmaLsrJcmeGTQOv9rSx+TN7sU8B8wbMlyE6+Us2XWhK1OoC5TOSv2HLDq6Ljd0SPFYNP550zDWNe1fZmJwxN0R9551n2SE/OTHBSYUzpsLxQ67d3cuFbgSIXwPGogFa2w81ZAg/usgOgjQrWoYzKnXJqthWD2g/Byszmcvs13oHAF4urB7OBRbutHCzmE3Bz2VmABPzQ==
What is Alice's email address?	9hEzxmm7S+SKqOTTqku4rsnYrcmOW/+wQ6z+31S+lCc=
Alice's email address is alice@example.com.	z86DVw2UBoYWmjxaSMBTefuB9ilOgztT8wCwq/GuRcQYtEkG+oowSAbwW24rbkKi



(a) 静态数据加密耗时 (b) 加密与非加密推理请求耗时

图 8 加密开销统计数据

表 6 各推理请求次数下的加密开销

推理请求次数	加密开销 (%)
1	−3.45
2	11.05
3	1.60
4	−5.75
5	−3.82
6	−2.30
7	1.60
8	4.16
9	−1.49
10	2.46

5.3.4 多用户并发推理处理效率测试

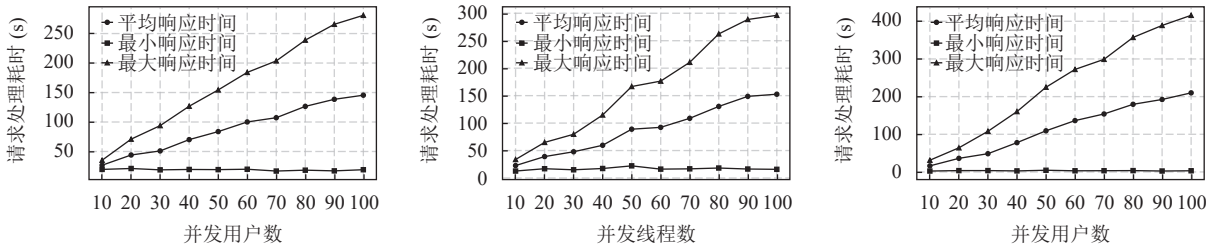
服务器网络通信框架基于 fastapi 搭建, 其路由处理逻辑支持异步操作. 模型方面, 则使用 vllm 在本地主机上启动推理服务, 服务器程序通过 OpenAI 对其进行访问. 用户使用推理服务时, 将提示词由中继服务器发送到模型服务器, 再转发到 vllm 本地服务器进行推理. 在实验过程中, 借助 httpx 库提供的异步客户端, 使用 asyncio 库创建多个协程发起推理请求, 从而模拟多用户场景. 本次实验中使用的推理模型为 DeepSeek-R1-Distill-Qwen-7B. 在所有用户使用相同提示词的情况下, 不同并发用户数量下的请求处理耗时如图 9(a) 和表 7 所示.

可见, 随着并发用户数增加, 推理服务的最小响应时间基本不变, 而平均响应时间和最大响应时间线性增长; 同时, 系统处理所有请求所需的总时间与最大响应时间非常接近 (图 9 中未标出). 在提示词相同的情况下, 相较于串行处理推理请求的场合, 效率可提升至 5–7 倍. 由此可知, 推理服务系统可以充分利用计算资源, 并发地处理多个推理请求, 并及时将已计算完成的推理结果返回给用户.

为了评估引入可信执行环境对时间开销的影响, 在服务器上创建多个线程, 就地访问推理服务, 进行相同的实验, 结果如图 9(b) 所示. 由实验结果可知, 在服务器本地直接发起推理请求并进行处理所需的时间, 与从客户端发起推理请求获取推理结果所需的时间接近. 由此, 在系统运行的过程中, 相较于模型推理的时间开销, 引入可信执行环境以及网络通信造成的额外开销可忽略不计.

另一方面, 在多用户场景中, vllm 可能在多个用户之间共享针对相同前缀 token 序列的 KV 缓存, 从而提高推理效率. 上述实验中所有用户使用的提示词相同,

系统可能利用这种缓存机制对推理计算进行了加速. 为了验证这一点, 令不同用户发送前缀不同的提示词进行实验, 此时缓存命中率应当会下降，进而导致推理效率降低. 请求处理耗时如图 9(c) 所示.



(a) 相同提示词并发用户数与请求处理耗时 (b) 相同提示词本地并发线程数与推理耗时 (c) 相异提示词并发用户数与请求处理耗时

图 9 并发推理耗时统计数据

表 7 并发请求相较于串行请求的性能提升

相同提示词并发用户数	最小响应时间×用户数/并发请求处理总耗时
10	5.57
20	5.96
30	6.02
40	6.16
50	6.23
60	6.48
70	5.79
80	6.14
90	5.80
100	6.79

与所有用户使用相同提示词的情况进行对比, 可以发现, 当多个用户同时发送不同前缀的提示词时, 系统处理全部请求的平均时间和总时间有所增加, 而平均响应时间、最小响应时间、最大响应时间随并发用户数增加的变化趋势则与发送相同提示词时相似. 由此可知, 在本系统中, 使用 vllm 部署的 DeepSeek-R1-Distill-Qwen-7B 模型能够通过 KV 缓存共享, 减小处理相同前缀提示词的总时间开销, 提高并发处理同类提示词推理请求的效率.

6 结束语

针对现有人工智能模型 (特别是大语言模型) 云推理服务运行时可能存在的安全风险, 本文提出了一种基于机密计算的大语言模型安全推理方案, 通过建立完整的机密计算系统对推理服务的全过程进行保护, 利用可信执行环境提供的安全属性, 将模型本身的结构和参数、用户提示词与推理结果的机密性和完整性均纳入保护范围. 同时, 通过将模型推理服务置于机密虚拟机中并将推理客户端程序置于机密容器中, 实现了与富执行环境中程序的兼容性与可迁移性. 此外, 实验结果表明, 系统提供的安全增强功能带来的额外开销相较于人工智能模型的推理开销可忽略不计; 在多用户高并发的场景中, 也表现出良好的性能.

参考文献

1 OpenAI. GPT-4 technical report. arXiv:2303.08774, 2024.

2 Team GLM. ChatGLM: A family of large language models from GLM-130B to GLM-4 all tools. arXiv:2406.12793, 2024.

3 DeepSeek-AI. DeepSeek-V3 technical report. arXiv:2412.19437, 2025.

4 He XL, Xu GW, Han XS, *et al*. Artificial intelligence security and privacy: A survey. Science China Information Sciences, 2025, 68(8): 181101. [doi: 10.1007/s11432-025-4388-5]

5 Sumalatha U, Prakasha KK, Prabhu S, *et al*. Touch of privacy: A homomorphic encryption-powered deep learning framework for fingerprint authentication. IEEE Access, 2025, 13: 59057–59073. [doi: 10.1109/ACCESS.2025.3555311]

6 Paradžik P, Derek A, Horvat M. Formal security analysis of the AMD SEV-SNP software interface. IEEE Transactions on Dependable and Secure Computing, 2025, 22(4): 3289–3306. [doi: 10.1109/TDSC.2025.3528737]

7 冯登国. 机密计算发展现状与趋势. 信息安全研究, 2024, 10(1): 2–5

8 Feng DG, Qin Y, Feng W, *et al*. Survey of research on confidential computing. IET Communications, 2024, 18(9): 535–556. [doi: 10.1049/cmu2.12759]

9 冯登国, 秦宇. 机密计算: 进展与展望. 中国计算机学会通讯, 2024, 20(12): 38–46

10 Shepherd C, Markantonakis K. Trusted Execution Environments. Cham: Springer, 2024. 4–7.

11 Akram A, Giannakou A, Akella V, *et al*. Performance

analysis of scientific computing workloads on general purpose TEEs. Proceedings of the 2021 IEEE International Parallel and Distributed Processing Symposium. Portland: IEEE, 2021. 1066–1076.

12 Lutsch A, El-Hindi M, Heinrich M, *et al*. Benchmarking analytical query processing in Intel SGXv2. Proceedings of the 28th International Conference on Extending Database Technology. Barcelona: OpenProceedings, 2025. 516–528.

13 Cheng PC, Ozga W, Valdez E, *et al*. Intel TDX demystified: A top-down approach. ACM Computing Surveys, 2024, 56(9): 238.

14 Masanori M, Stavrakakis D, Santos N, *et al*. Confidential VMs explained: An empirical analysis of AMD SEV-SNP and Intel TDX. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2024, 8(3): 36.

15 Xu YC, Pangia J, Ye CC, *et al*. Data enclave: A data-centric trusted execution environment. Proceedings of the 2024 IEEE International Symposium on High-performance Computer Architecture. Edinburgh: IEEE, 2024. 218–232.

16 Wang WH, Chen GX, Pan XR, *et al*. Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas: Association for Computing Machinery, 2017. 2421–2434.

17 Lee JH, Jang JS, Jang Y, *et al*. Hacking in darkness: Return-oriented programming against secure enclaves. Proceedings of the 26th USENIX Security Symposium. Vancouver: USENIX Association, 2017. 523−539.

18 Biondo A, Conti M, Davi L, *et al*. The guard's dilemma: Efficient code-reuse attacks against Intel SGX. Proceedings of the 27th USENIX Security Symposium. Baltimore: USENIX Association, 2018. 1213−1227.

19 Johnson MA, Volos S, Gordon K, *et al*. Parma: Confidential containers via attested execution policies. arXiv:2302.03976, 2023.

20 Johnson MA, Volos S, Gordon K, *et al*. Confidential container groups: Implementing confidential computing on Azure container instances. Acmqueue, 2024, 22(2): 57–86.

21 Pecholt J, Wessel S. CoCoTPM: Trusted platform modules for virtual machines in confidential computing environments. Proceedings of the 38th Annual Computer Security Applications Conference. Austin: Association for Computing Machinery, 2022. 989–998.

22 Randazzo A, Tinnirello I. Kata containers: An emerging architecture for enabling MEC services in fast and secure way. Proceedings of the 6th International Conference on Internet of Things: Systems, Management and Security. Granada: IEEE, 2019. 209−214.

23 Costan V, Devadas S. Intel SGX explained. IACR Cryptology ePrint Archive, 2016. 86: 1–86. https:/eprint.iacr.org/2016/086

24 Huang HY, Zhang FW, Yan SM, *et al*. SoK: A comparison study of arm TrustZone and CCA. Proceedings of the 2024 International Symposium on Secure and Private Execution Environment Design. Orlando: IEEE, 2024. 107–118.

25 Kieu-Do-Nguyen B, Nguyen KD, Dang TK, *et al*. A trusted execution environment RISC-V system on chip. Proceedings of the 2024 IEEE Hot Chips 36 Symposium. Stanford: IEEE, 2024. 1.

26 Tian Z, Yang S, Zhang CY. Accelerating sparse general matrix-matrix multiplication for NVIDIA Volta GPU and Hygon DCU. Proceedings of the 32nd International Symposium on High-performance Parallel and Distributed Computing. Orlando: ACM, 2024. 329–330.

27 Shen YL, Zhuang ZX, Yuan K, *et al*. Medical multimodal model stealing attacks via adversarial domain alignment. Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia: AAAI, 2025. 6842–6850.

28 Pei GZ, Lyu SJ, Ma K, *et al*. Exploring query efficient data generation towards data-free model stealing in hard label setting. Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia: AAAI, 2025. 667–675.

29 Aryal K, Gupta M, Abdelsalam M, *et al*. A survey on adversarial attacks for malware analysis. IEEE Access, 2025, 13: 428–459. [doi: 10.1109/ACCESS.2024.3519524]

30 Jiménez-López D, Rodríguez-Barroso N, Luzon MV, *et al*. Membership inference attacks fueled by few-short learning to detect privacy leakage tackling data integrity. arXiv:2503.09365, 2025.

31 Zhang SN, Ye L, Yi X, *et al*. "Ghost of the past": Identifying and resolving privacy leakage from LLM's memory through proactive user interaction. arXiv:2410.14931, 2024.

32 Zhang CL, Luo SL, Li JW, *et al*. Self-enhancing defense for protecting against model stealing attacks on deep learning systems. Expert Systems with Applications, 2025, 269: 126438. [doi: 10.1016/j.eswa.2025.126438]

33 Lou JD, Yuan X, Zhang R, *et al*. GRID: Protecting training graph from link stealing attacks on GNN models. Proceedings of the 2025 IEEE Symposium on Security and Privacy. San Francisco: IEEE, 2025. 2095–2113.

34 Zhang XL, Chen JL, Li QH, *et al*. LSSMSD: Defending against black-box DNN model stealing based on localized

stochastic sensitivity. International Journal of Machine Learning and Cybernetics, 2025, 16(3): 2041–2056. [doi: 10.1007/s13042-024-02376-0]

35 Mei JP, Zhang WB, Chen J, *et al*. Defense against model stealing based on account-aware distribution discrepancy. Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia: AAAI, 2025. 604–611.

36 Das S, Mishra S. Advances in differential privacy and differentially private machine learning. In: Gountia D, Dalei DK, Mishra S, eds. Information Technology Security. Singapore: Springer, 2024. 147–188.

37 Wang L, Yan HN, Lin XD, *et al*. Protecting bilateral privacy in machine learning-as-a-service: A differential privacy based defense. Proceedings of the 1st International Conference on Artificial Intelligence Security and Privacy. Guangzhou: Springer, 2023. 237–252.

38 Pustozerova A, Baumbach J, Mayer R. Differentially private federated learning: Privacy and utility analysis of output perturbation and DP-SGD. Proceedings of the 2023 IEEE International Conference on Big Data. Sorrento: IEEE, 2023. 5549–5558.

39 Gupta R, Singh AK. A differential approach for data and classification service-based privacy-preserving machine learning model in cloud environment. New Generation Computing, 2022, 40(3): 737–764. [doi: 10.1007/s00354-022-00185-z]

40 Lai P, Hu H, Phan N, *et al*. Lifelong DP: Consistently bounded differential privacy in lifelong machine learning. Proceedings of the 1st Conference on Lifelong Learning Agents. Montreal: PMLR, 2022. 778–797.

41 Yadav K, Gupta BB, Chui KT, *et al*. Differential privacy approach to solve gradient leakage attack in a federated machine learning environment. Proceedings of the 9th International Conference on Computational Data and Social Networks. Dallas: Springer, 2020. 378–385.

42 Blanco-Justicia A, Sánchez D, Domingo-Ferrer J, *et al*. A critical review on the use (and misuse) of differential privacy in machine learning. ACM Computing Surveys, 2023, 55(8): 160.

43 Domingo-Ferrer J, Sánchez D, Blanco-Justicia A. The limits of differential privacy (and its misuse in data release and machine learning). Communications of the ACM, 2021, 64(7): 33–35. [doi: 10.1145/3433638]

44 Zhao BZH, Kaafar MA, Kourtellis N. Not one but many tradeoffs: Privacy vs. utility in differentially private machine learning. Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop. New York: ACM, 2020. 15–26.

45 Liu ZY, Zhou T, Luo YK, *et al*. TBNet: A neural architectural defense framework facilitating DNN model protection in trusted execution environments. Proceedings of the 61st ACM/IEEE Design Automation Conference. San Francisco: Association for Computing Machinery, 2024. 310.

46 Casella B. A performance analysis of VM-based trusted execution environments for confidential federated learning. Proceedings of the 33rd Euromicro International Conference on Parallel, Distributed, and Network-based Processing. Turin: IEEE, 2025. 204–208.

47 Lee D, António J, Khan H. Privacy-preserving decentralized AI with confidential computing. arXiv:2410.13752, 2024.

48 Hashemi H, Wang YQ, Annavaram M. DarKnight: An accelerated framework for privacy and integrity preserving deep learning using trusted hardware. Proceedings of the 54th Annual IEEE/ACM International Symposium on Microarchitecture. New York: ACM, 2022. 212–224.

49 Xu RW, Fang ZX. Tempo: Confidentiality preservation in cloud-based neural network training. Proceedings of the 2024 International Joint Conference on Neural Networks. Yokohama: IEEE, 2024. 1–10.

50 Tramèr F, Boneh D. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. Proceedings of the 7th International Conference on Learning Representations. New Orleans: OpenReview.net, 2019. 1–19.

51 Feng EH, Feng DH, Du D, *et al*. sNPU: Trusted execution environments on integrated NPUs. Proceedings of the 51st ACM/IEEE Annual International Symposium on Computer Architecture. Buenos Aires: IEEE, 2024. 708–723.

52 Mohan A, Ye MM, Franke H, *et al*. Securing AI inference in the cloud: Is CPU-GPU confidential computing ready? Proceedings of the 17th IEEE International Conference on Cloud Computing. Shenzhen: IEEE, 2024. 164–175.

53 Dhar A, Thorens C, Lazier LM, *et al*. Ascend-CC: Confidential computing on heterogeneous NPU for emerging generative AI workloads. arXiv:2407.11888, 2024.

54 Tan YF, Mi ZY. Performance analysis and optimization of nvidia H100 confidential computing for AI workloads. Proceedings of the 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications. Kaifeng: IEEE, 2024. 1426–1432.

(校对责编: 李慧鑫)