中国科学:信息科学 在审文章 onlinelatex

SCIENTIA SINICA Informationis

论文

面向大语言模型智能体的口令基数据安全传输方案

宋蜜[1,2,3], 汪定[1,2,3*]

1. 南开大学密码与网络空间安全学院, 天津 300350
2. 数据与智能系统安全教育部重点实验室, 天津 300350
3. 天津市网络与数据安全技术重点实验室, 天津 300350
* 通信作者. E-mail: wangding@nankai.edu.cn

国家自然科学基金 (批准号: 624B2071, 62572259)、天津市自然科学基金 (批准号: 25JCJQJC00230) 和中央高校基本科研业务费专项资金 (批准号: 63263258) 资助项目

摘要 大语言模型智能体通过集成任务理解、工具调用与自主决策能力, 快速推动人工智能技术由内容生成向任务执行演进, 在金融分析、医疗辅助、政务处理等敏感领域展现出广阔应用前景. 然而, 智能体在调用外部工具完成任务的过程中, 普遍存在用户与工具间身份认证缺失、用户隐私数据保护不足以及访问控制粒度过粗等安全问题, 易引发身份仿冒、非法访问、隐私泄露及信任边界模糊等风险, 难以满足敏感场景下用户与智能体之间的数据安全共享需求. 为解决上述问题, 本文提出一种面向大语言模型 (Large language model, LLM) 智能体的口令基数据安全传输方案. 该方案面向用户经由 LLM 调用外部工具的交互场景, 使用户与外部工具服务器能够基于低熵口令协商生成高熵会话密钥, 实现具有前向安全性的双向身份认证. 在数据传输过程中, 本文将访问控制策略与会话密钥共同嵌入密文, 实现基于属性的细粒度授权, 确保敏感数据仅能在通过认证且满足访问策略要求的授权端完成解密与推理. 同时, LLM 仅负责任务解析与响应模板生成, 全程无需接触明文数据, 实现任务调度与敏感数据处理的有效分离. 性能分析结果表明, 本文方案满足全部 12 项安全性评价指标, 在包含 5 个属性的典型策略下, 总计算开销为 239 毫秒, 通信开销为 7,744 比特, 与现有方案的效率相当, 但在功能与安全性上显著增强. 本文以用户可记忆的口令为信任根, 在不改变现有 LLM 智能体服务架构的前提下, 为大语言模型智能体场景下敏感数据的安全传输与访问控制提供了一种安全高效的解决方案.

关键词 大语言模型, 智能体, 身份认证, 细粒度访问控制, 隐私保护

1 引言

大语言模型 (Large language models, LLMs) 作为自然语言处理领域的重要技术, 近年来在文本生成、信息检索、对话系统等多种应用中展现出卓越的性能和广泛的适用性 [1,2]. 诸如 ChatGPT [3], LLaMA [4], DeepSeek [5] 等主流模型在学术、医疗、教育、金融等多个领域实现了接近甚至超越人类水平的语言理解与生成能力, 成为当前学术界与工业界共同关注的研究热点. 与此同时, 越来越多的研究工作开始探索将大模型作为智能体的控制中心以提高智能体在开放领域和动态环境中的性能 [6], 如以 AutoGen [7], HuggingGPT [8],

引用格式: 宋蜜, 汪定. 面向大语言模型智能体的口令基数据安全传输方案. 中国科学: 信息科学, 在审文章
Song M, Wang D. A password-based secure data transmission scheme for large language model agents. Sci Sin Inform, for review

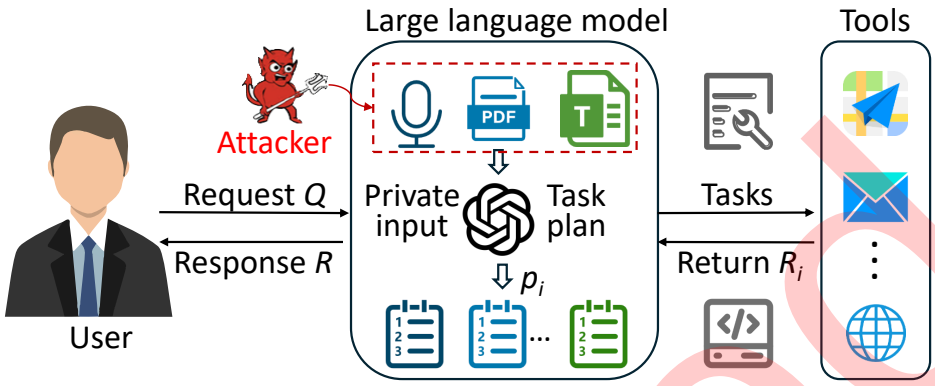

图 1　大语言模型智能体架构.

Figure 1　Architecture of a tool-using large language model agent.

MetaGPT [9] 为代表的工具驱动型大语言模型智能体. 该类智能体以大语言模型为核心控制器, 通过调用规划、记忆、工具使用等能力 [10], 完成用户提交给智能体的任务.

然而, 随着 LLM 智能体由 “内容生成” 走向 “任务执行”, 其所面临的安全挑战也由模型本身的生成安全问题进一步扩展到多主体协同过程中的数据安全与访问控制问题. 如图 1 所示, 当大模型驱动的智能体面向用户提供服务时, ①用户向智能体提交任务请求 Q; ②智能体在获取用户发来的请求之后, 先通过大模型的规划能力为 Q 制定出计划 p_i; ③智能体随后调用包括搜索引擎、代码解释器在内的多种外部工具, 按照计划来执行相应的操作, 获得响应 R_i; ④智能体将组合后的响应 R 返回给用户. 在此过程中, LLM 不仅掌握了用户的私有输入, 而且在任务执行阶段会将这些信息无意中共享给相应的工具. 例如, 当用户使用工具调用型 LLM 智能体来描述健康问题时, 会向 LLM 提供一张病历单, 而 LLM 随后会将该病历共享给第三方模型部署平台. 这种交互导致用户的隐私数据同时暴露给了 LLM 和第三方工具提供商平台.

与传统 “用户-服务器” 两方交互 [11,12] 模式相比, LLM 智能体 [7~9] 具有参与实体更多、处理阶段更长、信任关系更复杂等特点. 一方面, 用户往往直接与 LLM 平台交互, 而真正执行任务的却是由 LLM 调用的外部工具服务器, 导致用户与实际处理数据的工具端之间缺乏直接可信的认证关系; 另一方面, 用户敏感输入及工具计算结果在多方之间传递和聚合的过程中, 容易暴露于 LLM 平台、第三方工具服务以及中间通信链路之中, 从而带来隐私泄露、越权访问、责任边界不清等风险. 此外, 现有工具调用流程通常以功能可用性为主要设计目标, 对 “谁可以访问什么数据”“在哪个阶段可以解密和处理数据” 缺乏精细刻画, 因而难以满足敏感场景下对端到端安全共享与细粒度访问控制的要求. 随着大语言模型集成应用的兴起, 工具链环节的漏洞增多, 可能导致信息泄露和资金损失等实际危害. 据新闻报道, 2025 年 6 月, 瑞士网络安全公司 Invariant Labs [13] 发现 GitHub 官方 MCP 服务器存在漏洞, 攻击者可在公共仓库中隐藏恶意指令, 诱导 Claude 4 等 AI 智能体泄露 MCP 用户的私有仓库敏感数据, 随后该公司立即提出动态权限控制进行防御, 以限制 AI 智能体的访问权限.

身份认证是防止未授权访问和恶意交互的第一道防线. Wang 等人 [14] 明确把 “智能体身份认证威胁” 列为智能体互联网安全的首要维度, 指出智能体身份认证易遭受伪造、冒充、Sybil 攻击、权限提升和意图欺骗等威胁, 这些都会削弱访问控制在动态场景中的有效性, 文中强调应根据智能体能力、行为和交互模式进行细粒度、上下文感知调整, 从而在身份伪造或滥用时仍能限制其访问行为. 面对不断演变的攻击手段, 保护用户隐私数据与实现动态授权已成为该领域的关键问题. 现有智能体身份认证协议主要依赖 API 密钥或 OAuth 2.0 认证框架 [15], 用户向 LLM 服务商颁发长期访问令牌以调用外部工具服务. 此类机制存在三重固有缺陷: 一是静态凭证风险, API 密钥通常长期有效且不可变, 一旦泄露攻击者可永久冒充合法身份, 难以撤销, 且缺乏口令的 “用户可记忆” 特性; 二是前向安全性缺失, 令牌长期持续有效, 一旦泄露, 历史会话将面临风险; 三是缺乏双向认证, 用户向服务器证明身份依赖第三方 LLM 平台的中转授权, LLM 成为不可审计的信任中介.

1.1 相关工作

随着人工智能领域大语言模型的广泛应用, 工具驱动的 LLM 智能体数据隐私问题受到了广泛关注 [16]. Deng 等人 [17] 从社会关系和技术应用角度分析了大语言模型智能体安全的整体趋势, 并指出模型的生成能力越强, 对数据隐私保护的挑战就越大. 由于生成内容的技术发展迅速, 隐私问题也日益严峻. Wei 和 Wang [18] 系统评估了 18 个主流中英文大语言模型在处理用户个人信息时的合法性与合规性, 结果表明, 50% 的中国大语言模型没有额外加密或解释用户的私人数据, 在掩码关键的个人身份信息方面, 发现 5/8 的中国大语言模型忽视了安全和隐私因素, 并根据上下文填补缺失的用户个人信息. 值得警惕的是, LLMs 对掩码的个人信息恢复中均可达 100% 的成功率, 显示出其在隐私重构方面的强大能力. Pedro 等人 [19] 指出, 攻击者可通过提示注入使大语言模型生成恶意 SQL 查询, 若应用程序未对输入进行充分验证与清理, 该查询将被执行, 从而导致数据库被未授权访问, 危及数据的完整性和机密性.

为应对隐私数据泄露风险 [20], 研究者提出了多种防护机制, 以保障 LLM 推理阶段的隐私安全. 现有面向 LLM 推理阶段的隐私保护方法, 重点涵盖三类: 基于密码学的隐私保护方法 (如同态加密 [21~24], 差分隐私 [25~27], 安全多方计算 [28~30] 等), 通过检测实现的隐私保护方法 (如黑盒测试 [31] 等) 以及基于硬件的隐私保护方法 (如可信执行环境 [32] 等). CoGenesis [33] 将用户请求划分为不含隐私的指令信息与包含隐私的用户画像, 并提出草稿填充与概率融合同步生成机制, 使云端模型仅处理非敏感部分, 从而实现对用户隐私的保护. LatticeGen [34] 则通过在真实 token 之外混入多个虚假 token, 以混淆用户真实输入, 使云端模型难以直接识别真实请求内容. 进一步地, PrivacyAsst [35] 将研究场景扩展到工具调用型智能体, 提出基于同态加密和随机重排的两类隐私保护方法, 其中前者依赖可处理密文输入的模型与工具, 后者通过引入虚假图像并结合多工具协同来隐藏真实输入, 但相应地带来了较高的计算与通信开销, 且其安全性依赖于工具间不存在串通的假设.

与此同时, 已有研究表明, 即使不直接暴露原始输入, 仅泄露中间数据也可能造成严重隐私风险. Pan 等人 [36] 发现, 未加保护的向量表示中可能包含与原始输入高度相关的敏感信息, 并通过模式重建攻击和关键字推理攻击, 从 BERT 等模型的向量表示中成功恢复了身份证号中的生日信息、基因组序列中的敏感核苷酸特征, 以及评论和病例文本中的位置与疾病关键词. 与 Pan 等人 [36] 的工作不同的是, Song 等人 [37] 的攻击并不以输入文本存在特定结构和模式为前提, 分别在白盒和黑盒两种场景下, 通过多种逆向技术从较短输入文本的向量表示中以较高的准确率和召回率恢复了部分文本词汇. 这表明仅依赖模型内部表征或中间结果的隐藏, 并不能从根本上解决大模型场景下的隐私泄露问题.

为缓解智能体中的身份认证威胁, 访问控制机制 (如基于角色、属性和策略的访问控制 [38]) 对于防止未授权访问至关重要. 一般而言, 可通过建立清晰的数据访问控制策略, 并结合服务提供方的制度约束, 对多智能体系统中不同角色设置差异化访问权限. Huang 等人 [39] 提出了一种零信任身份框架, 将去中心化标识符与可验证凭证相结合, 用于支持动态细粒度访问控制与会话管理, 从而实现任务级隐私隔离与最小权限. Chan 等人 [40] 则在智能体侧引入实时监测组件, 通过捕获输入输出、提取操作特征并预测下游任务表现, 对响应进行动态调整, 以降低未授权交互带来的隐私风险. He 等人 [41] 指出, 智能体之间的数据传输可能导致敏感信息越权访问或泄露给未授权实体, 且由于多智能体交互规模大、过程不透明, 生成信息的监管难度进一步增加. 总体而言, 现有工作表明, 面向智能体场景的未授权访问亟需更加严格且细粒度的访问控制机制.

1.2 研究动机与贡献

大语言模型智能体凭借其任务规划、工具调用与自主决策能力, 快速推动人工智能系统从内容生成向任务执行范式演进. 然而, 这种强大的协作能力也引入了严峻且全新的安全挑战. 在典型的智能体工作流中, 用户的私有数据 (如内部报告、个人健康信息) 需连同任务指令一同提交给 LLM, 由 LLM 解析后分发给外部工具进行处理, 这些传输数据也成为攻击者潜在的高价值攻击目标. 据公开报道, 由于三星员工直接将企业机密信

息以提问的方式输入到 ChatGPT 中, 导致引入 ChatGPT 不到 20 天就发生 3 起数据外泄事件, 其中 2 次和半导体设备有关, 1 次和内部会议有关, 这导致三星半导体设备测量资料、产品良率原封不动传输给了美国公司 [42]. 近期, 一个名为 “EchoLeak” 的高危漏洞被公开披露[43], 该漏洞揭示了微软 Copilot 等 AI 助手在企业环境中的潜在安全风险, 攻击者不需要诱导点击, 也不需要投放恶意附件, 只借助一封看似 “正常” 的电子邮件, 即可借助语义操控手段绕过权限边界, 从而在用户无感知、安全系统无告警的 “零交互” 过程中窃取企业最敏感的数据. 由这些事件可看出数据在智能体场景下面临的风险体现在三个方面: 一是数据在传输或临时存储阶段可能被网络攻击者窃取或者篡改; 二是作为协调中心的 LLM 服务提供商, 本身也可能成为 “诚实但好奇” 的观察者有意或无意地接触用户数据; 三是外部工具对数据的访问往往缺少足够细粒度的控制, 一个已获得授权的工具服务器仍然可能出现权限滥用或数据外泄的问题. 随着智能体在不同任务中角色和权限的频繁变化, 静态的认证后长期授权模型不再适用, 需要能够根据当前任务、能力和环境实时调整身份认证与授权策略.

在当前的 LLM 智能体场景中, 用户与外部工具之间的身份认证机制虽然已经受到关注, 但仍是一个公认且亟待解决的重大挑战. 2026 年 2 月 NIST 正式发起 AI Agent 标准化倡议 [44], 明确指出跨边界调用过程中用户与工具之间缺乏可验证身份这一核心问题, 并认为智能体虽然已经具备完成复杂交易的能力, 但仍缺少使这些交易真正可信的 “信任基础设施”. 现有方案普遍存在认证碎片化、集成复杂以及缺乏对智能体 “非人类身份” 的细粒度管理等问题, 这与智能体本身具有动态性和自主性的特点并不匹配, 也使得相关挑战难以被系统性解决. 传统的传输层安全协议主要保证数据在传输过程中的机密性, 却不能阻止数据在 LLM 侧或已授权工具侧以明文形式被访问; 基于 API 密钥的静态认证方式又较容易泄露, 授权粒度也相对粗糙, 难以支持 “一次一密” 以及结合数据内容进行动态策略控制. 传统保密性 (Confidentiality)、完整性 (Integrity)、可用性 (Availability) 的 CIA 模型 [45] 已经不足以覆盖智能体场景中的实际安全需求, 在此基础上, 必须加入**真实性** (能否验证智能体身份) 和**可信性** (能否授权其解读与推理) 两个维度. 尽管属性基加密技术为细粒度访问控制提供了解决方案 [46], 但如何将其与轻量级、用户友好的口令认证相结合, 并自然嵌入到以 LLM 为协调中心的计算流程中, 目前仍缺少充分研究. 因此, 如何在大语言模型智能体场景下构建一个既能实现用户-智能体强身份认证, 又能支持动态访问控制的数据隐私保护方案, 成为一个重要问题.

针对上述问题, 本文提出一种基于口令的大模型智能体数据安全传输和访问控制方案. 在 LLM 作为不可信协调者的开放环境中, 本文以用户可记忆的口令为认证凭据, 实现用户私有数据到智能体间的安全传输与细粒度访问控制. 具体而言, 用户首先通过强口令认证密钥协商协议与目标外部工具服务器进行双向认证, 并协商出临时高熵会话密钥. 随后, 用户利用该会话密钥及自主定义的访问控制策略对私有数据进行加密, 并将生成的密文与明文查询一同提交给 LLM. LLM 扮演 “盲协调者” 角色: 基于明文查询解析任务、规划流程, 但仅能对加密数据执行路由与转发, 无法解密数据内容. 只有满足访问策略的指定外部工具, 才能使用其属性私钥和会话密钥解密数据并执行计算, 再将加密后的结果返回. 最终, 用户可恢复出最终响应. 本文并非单独解决推理隐私保护或传统两方认证问题, 而是面向 LLM 智能体场景下身份认证缺失、隐私数据泄露、访问控制粒度过粗等多重安全问题, 提出一种兼顾安全性与实用性的数据安全传输与受控访问解决方案. 本文贡献如下:

(1) 提出一种面向 LLM 智能体场景的安全数据传输架构. 针对现有 LLM 智能体明文传输和静态授权导致的隐私泄露与权限滥用问题, 本文首次将强口令认证与属性基加密有机结合, 以用户熟记的口令为信任根, 实现了从可信身份认证到数据内容访问的全程隐私保护. 该框架不仅实现了用户与目标工具服务器之间的双向可信身份认证, 同时实现了用户隐私数据的细粒度访问控制, 使数据的安全策略能够绑定于数据本身, 确保敏感数据的解密与处理仅在通过认证且满足访问策略要求的授权端完成.

(2) 设计了一种面向不可信 LLM 协调者的安全协同机制. 本文将所有与敏感数据相关的解密与推理操作限制在用户终端和经认证授权的外部服务器内部, 使 LLM 仅承担任务解析与密文调度功能, 而不接触数据明文, 确保 LLM 在无需解密的情况下完成复杂任务解析, 在保留 LLM 智能体任务规划与工具调用能力的同时,

实现了任务调度过程与敏感数据处理的有效分离, 从根本上防御了协调者层面对明文数据内容的窥探.

(3) 构建了面向 LLM 智能体数据安全传输场景的评价指标体系, 并完成了系统性的安全与性能分析. 本文从身份认证与密钥建立、数据机密性与访问控制以及智能体系统适配性三个层面构建了包含 12 项指标的评价体系, 并将所提方案与相关代表性方案进行比较分析. 结果表明, 本文方案能够满足全部评价指标要求, 同时保持可接受的计算开销与通信性能, 体现出较好的实用性与系统适配性.

2 预备知识

2.1 基于口令的认证密钥协商协议

为在用户与外部工具服务器之间建立安全信道并实现双向认证, 本文采用基于口令的认证密钥协商 (Password authenticated key exchange, PAKE) [11] 协议. 该类协议允许通信双方仅依赖低熵口令 pw 协商生成高熵会话密钥 sek, 并满足以下安全性质: 1) 工具服务器在协议执行过程中无需掌握用户明文口令, 仅保存与口令绑定的验证记录 Cre, 从而能够抵抗服务器数据库泄露场景下的离线字典攻击; 2) 在合法用户与服务器均诚实执行协议的前提下, 双方最终输出相同的会话密钥 sek; 3) 对于未持有正确口令的攻击者, 即使其能够完全窃听协议交互消息, 或主动发起重放、篡改等攻击, 也无法有效区分会话密钥 sek 与随机密钥. 在本文方案中, 该会话密钥不仅用于保障用户与目标外部工具服务器之间认证阶段的安全性, 还被进一步作为后续数据密钥与结果密钥派生的根密钥, 用于构建数据上传、工具处理和结果返回全过程的安全传输通道.

2.2 基于密文策略的属性基加密

为实现细粒度访问控制, 本文采用基于密文策略的属性基加密 (Ciphertext-policy attribute-based encryption, CP-ABE) [46], 由以下四个多项式算法构成:

$\mathsf{Setup}(1^\kappa) \to (mpk, msk)$: 系统建立阶段, 输入安全参数 κ, 生成系统的主私钥 msk 和公共参数 mpk.

$\mathsf{ABEnc}(mpk, m, \Gamma) \to ct$: 加密阶段, 输入公钥 mpk, 明文消息 m 和访问策略 Γ, 生成与 Γ 相关的密文 ct.

$\mathsf{KeyGen}(mpk, msk, Attr) \to sk$: 密钥生成阶段, 输入 mpk, msk 和属性集 $Attr$, 生成 $Attr$ 相关的密钥 sk.

$\mathsf{ABDec}(mpk, sk, ct) \to m$: 解密阶段, 输入 mpk, 解密密钥 sk 和与访问策略 Γ 相关的密文 ct. 如果解密密钥中的属性集 $Attr$ 满足密文中嵌入的访问策略 Γ, 即 $Attr \in \Gamma$, 则恢复明文消息 m, 否则输出 $\perp$.

正确性. 对于 CP-ABE 而言, 若属性集合 $Attr$ 满足相应的访问策略 Γ, 即 $\Gamma(Attr) = 1$, 则有 $\mathsf{Pr}[m = m' \mid \mathsf{ABEnc}(m, \Gamma) \to ct, \mathsf{ABDec}(ct, sk) \to m'] \leqslant \mathsf{negl}(\kappa)$, 其中 $\mathsf{Setup}(1^\kappa) \to (mpk, msk), \mathsf{KeyGen}(mpk, msk) \to sk$.

2.3 访问策略

在 CP-ABE 体制中, 访问策略用于刻画 “哪些属性组合被允许解密” [46]. 设 $\mathbb{S}$ 为系统中预先定义的属性全集, 每个实体 P(如用户或服务器) 由一个属性集合 $\Gamma \subseteq \mathbb{S}$ 描述. 访问策略通常形式化为一个访问结构 $\Gamma \subseteq 2^{\mathbb{S}}$, 其中 Γ 表示所有被授权的属性子集的集合: 当且仅当 $Attr \in \Gamma$ 时, 实体 P 被认为满足该策略并具有解密权限. 为了保证合理性, 一般要求 Γ 为单调访问结构, 即若 $Attr \in \Gamma$ 且 $Attr \subseteq Attr'$, 则必有 $Attr' \in \Gamma$. 在具体实现中, 访问策略常用布尔公式、访问树或者线性秘密共享结构的形式给出, 通过这种方式, 访问策略被直接嵌入密文之中, 实现了以属性为粒度的细粒度访问控制.

2.4 困难问题

设 G 是由安全参数 λ 确定的循环群, 其阶为素数 p, 生成元为 g. 随机选择 $a, b \xleftarrow{\$} Z_p$, 给定实例 (g, g^a, g^b), 计算 Diffie-Hellman 问题 (Computational diffie-hellman problem, CDH) [47] 的目标是在未知 a, b 的条件下计

算出 g^{ab}. 若对任意多项式时间概率算法 $\mathcal{A}$, 定义其成功输出 g^{ab} 的概率 $Pr[\mathcal{A}(g, g^a, g^b) = g^{ab}]$ 均可由关于 λ 的可忽略函数所上界, 则称 CDH 问题在群族 G_λ 上是困难的, 对应的 CDH 假设即认为不存在任何能以不可忽略的概率在多项式时间内求解 CDH 问题的算法.

2.5 大语言模型智能体

工具调用型 LLM 智能体是当前 LLM 应用的一种新范式, 旨在解决 LLM 固有的局限性, 例如多模态技能有限、数学计算能力初级以及难以保持信息实时更新等问题. 此类智能体使 LLM 能够利用外部工具 (如计算器、搜索引擎和机器学习模型), 而无需改变底层模型参数. 能够通过指令结合 LLM 进行操纵的工具领域日益多样化, 从而能够完成复杂任务. 从系统架构看, 一个典型的工具调用型 LLM 智能体包含以下核心组件: 1) 规划模块: 由 LLM 本身实现, 负责解析用户自然语言请求, 将其分解为结构化的可执行子任务序列, 并管理任务间的逻辑与依赖关系. 2) 工具集成层: 提供对外部工具 (如计算器、数据库、专业 AI 模型) 的标准调用接口. LLM 通过此层获取工具的功能描述与调用方式. 3) 执行与协调引擎: 根据规划模块的输出, 按序调用相应工具, 处理中间结果, 并将最终结果整合返回给用户. 这种 “Tool-using agent” 的架构 [35] 显著提升了系统在复杂任务上的能力, 但也带来了新的安全与隐私风险: 用户数据往往在多个组件之间多级传输, 且部分工具运行在独立的外部服务器上, 使得数据流向难以被用户直接感知和控制.

3 系统架构和安全模型

3.1 系统模型

本文提出一种面向大语言模型 (LLM) 智能体的数据安全共享方案, 该方案结合口令认证密钥交换 (PAKE) 与密文策略属性基加密 (CP-ABE) 机制, 以解决智能体工具调用链路中的身份认证与数据访问控制问题. 如图 2 所示, 本文协议包含以下三个实体: 用户 U, 大语言模型 L 以及外部工具服务器 S.

用户 U: 敏感数据的拥有者与任务请求的发起方, 持有低熵口令与属性私钥 (在数据上传阶段由属性授权中心分发). 用户在登录时向服务器提供自己的用户名和口令 PW, 通过与外部服务器之间的交互实现身份认证和加解密隐私数据的功能. 用户与目标外部服务器执行一个增强的口令认证密钥交换协议. 此过程完成双方基于口令的相互认证, 并协商出一个高熵的、临时会话密钥 sek, 为后续所有通信提供机密性与完整性保障.

大语言模型 L: 任务协调者, 由第三方 LLM 服务商提供 (如 OpenAI GPT-4), 负责解析用户自然语言查询, 将其分解为一系列具体的子任务, 并生成一个结构化的响应模板. 全程以密文形式作为输入参数, 其自身不直接访问用户明文数据. 后续将收集到的所有密文结果填入响应模板, 整合成最终的加密响应返回给用户.

外部工具服务器 S: 由大语言模型 L 调用的实际任务执行实体, 可表示为托管于第三方平台上的 AI 模型、数据库服务、代码执行环境或其他专用工具. 针对本文所关注的敏感数据处理场景, S 内部至少包含两个逻辑功能模块: 一是认证模块, 用于与用户执行口令认证密钥协商协议并建立高熵会话密钥; 二是执行模块, 用于在满足授权条件的前提下对加密输入进行解密、推理和结果生成. 为便于描述, 本文将工具服务器的认证与执行功能视为处于同一受控服务域内[1)]. 工具端需同时满足两个条件才能执行计算: 1) 模型的属性私钥必须匹配密文嵌入的访问控制策略; 2) 服务器必须持有正确的会话密钥 (即通过用户的身份认证). 满足条件后, 工具端解密获得明文数据并进行计算, 将推理结果使用相同的会话密钥和访问策略加密, 并将密文结果返回给 L.

用户 U 与大语言模型 L 之间, 以及 L 与工具服务器 S 之间分别存在双向通信信道, 如图 2 所示. 用户将请求 $Q = \{\mathsf{prom}, \Gamma_A, CT\}$ 发送给 L, 并期望获得正确的响应. L 在接收到用户请求后, 会利用自身能力解析该请求, 并将其分解为多个子任务 $\mathsf{task_i}$. 随后, L 基于其知识对任务的执行顺序和依赖关系进行规划, 并据此

1) 本文的 “同一受控服务域” 是指身份认证模块与工具执行模块处于相同的信任边界内, 两者之间的通信被视为安全、可信且无需额外密码学保护.

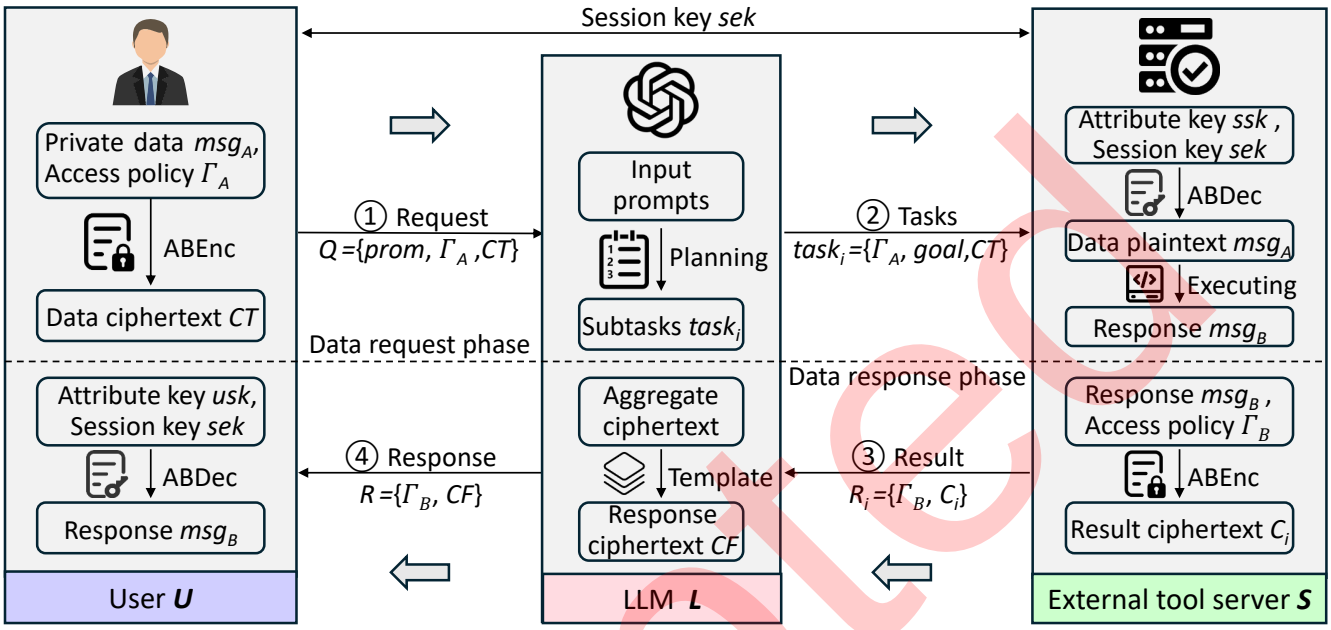

图 2　系统模型.
Figure 2　System model.

生成响应模板 temp. 响应模板 temp 通过向 L 输入预先设计的提示语中获取. 接着, L 将解析得到的各个任务分发给相应的工具服务商. 调用工具在给定的输入参数 CT 上执行分配到的任务, 并将预测结果 C_i 返回给 L. 在收到所有结果后, 大语言模型 L 将这些结果填入模板 temp, 生成最终响应 R 返回给用户. 系统中引入两类密码学组件: 一是基于口令的认证密钥协商协议, 用于在用户 U 与外部服务器 S 之间建立高熵共享会话密钥, 实现双方身份认证与安全信道协商; 二是密文策略属性基加密, 用于在密文层实现细粒度访问控制, 使得只有满足指定属性策略的实体才能对加密数据进行解密. 通过上述设计, 明文数据仅在用户终端和经授权的外部服务器内部出现, 智能体平台和中间传输节点始终只能看到密文, 从而在不改变大语言模型智能体任务规划与对话流程的前提下, 实现了基于口令和属性的双向身份认证与细粒度访问控制数据安全共享机制.

为具体说明本方案的适用场景, 我们参考当前主流的 “大语言模型-工具” 智能体架构范式 (如图 1), 设定一个由 OpenAI 的 GPT-4 作为大语言模型 L 的提供方 (核心推理与调度器), 并由 Replicate 平台作为外部工具服务方. Replicate 托管了大量开源、可专精处理图像、音频、视频的 AI 模型 (如用于图像生成的 Stable Diffusion、用于语音识别的 Whisper), 供外部通过 API 调用. 在此设定下, 用户可通过自然语言请求, 例如 “分析我上传的会议录音并生成一份图文摘要”, GPT-4 负责解析该复杂任务, 并安全地协调 Replicate 上的语音识别、文本摘要和文生图模型协同完成. 在本文提出的方案下, 会议录音及相关中间结果在上传和跨平台转发过程中均保持加密状态, GPT-4 仅负责任务规划与模型编排, 而不直接接触用户的明文数据.

3.2　威胁模型

在上述系统模型基础上, 本节进一步给出本文所考虑的威胁模型, 用以刻画攻击者能力、系统信任边界以及方案所关注的安全目标. 本文研究在以下攻击场景中如何保障用户敏感数据在智能体调用链路中的安全传输与受控访问: 大语言模型作为协调者不完全可信, 通信链路完全公开且易受窃听与篡改, 外部工具服务器存在未授权访问及内部数据泄露的风险. 基于此, 本文主要考虑以下几类攻击者, 其对应的攻击目标如表 1 所示.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

表 1 本文攻击者模型.

Table 1 Adversary models considered in our work.

Adversary model†	Attack goal	Attack target
$\mathcal{A}_{net}, \mathcal{A}_L, \mathcal{A}_S$	Privacy leakage	Obtain the private input and the final inference result.
$\mathcal{A}_{net}, \mathcal{A}_S$	Identity forgery	Impersonate a legitimate user without valid credentials.
$\mathcal{A}_S, \mathcal{A}_{col}$	Unauthorized access	Decrypt unauthorized ciphertexts with partial attributes.
$\mathcal{A}_{net}, \mathcal{A}_L, \mathcal{A}_S$	Result tampering	Tamper with protocol messages or ciphertexts.

† We consider partial key leakage, including leakage of the user's password or the participating entity's attribute key. The adversary is assumed unable to obtain both the user's password and a valid attribute key satisfying the access policy.

(1) 外部网络攻击者 $\mathcal{A}_{net}$. 外部网络攻击者指公开通信链路上的攻击者, 能够对用户 U、大语言模型 L 与外部工具服务器 S 之间的消息进行窃听、拦截、重放、篡改与伪造. 具体而言, $\mathcal{A}_{net}$ 可以观察用户提交的查询 Q、传输中的输入密文 CT、结果密文 C_i 以及最终密文响应 R, 并试图通过被动监听或主动干预获取用户隐私信息, 破坏协议正确执行. 换言之, 本文假设通信信道不可信, 攻击者具备典型公开信道攻击能力.

(2) 不可信大语言模型 $\mathcal{A}_L$. 本文将大语言模型 L 视为诚实但好奇的中间调度者. L 会严格按照既定协议流程完成用户任务解析、子任务分发与响应模板生成等调度功能; 同时, L 可能试图从用户查询、任务结构、传输密文及工具返回结果中推断用户敏感信息. 本文假设大语言模型能够观察其交互过程中所接触的全部明文查询和密文消息, 但不应掌握用户口令、会话密钥 sek 以及合法工具端或用户端的属性私钥.

(3) 未授权外部工具服务器 $\mathcal{A}_S$. 除目标工具服务器外, 本文还考虑未授权或错误接收密文的外部工具服务器. 此类实体可能获得由 L 转发的输入密文或结果密文, 并尝试在不具备合法授权的情况下恢复其明文内容. 攻击目标包括: 伪装为合法工具端、利用错误分发获得额外数据、或通过合并属性信息突破访问策略限制. 本文重点关注的是: 即使密文被分发至非目标工具服务器, 未满足访问策略的实体也无法解密得到明文.

(4) 串谋攻击者 $\mathcal{A}_{col}$. 本文进一步考虑多个未授权实体之间的串谋情形, 包括多个工具服务器之间的串谋, 或 L 与未授权工具服务器之间的信息联合. 此类攻击者试图通过共享各自持有的属性信息、密钥材料或中间消息, 构造超出各自权限范围的数据访问能力, 进行数据的越权访问.

需要指出的是, 本文方案并不假设大语言模型 L 的任务规划结果在语义层面完全正确, 即 L 可能因模型局限性产生不合理的子任务分解或工具选择, 导致任务执行失败或返回无意义结果, 该问题属于智能体系统可靠性范畴, 可通过上层应用的重试机制或人工校验解决. 此类功能层面的错误并不影响数据安全, 本文方案不依赖大语言模型调度的正确性来保证数据机密性, 而是通过密码学机制确保即使调度发生异常, 数据明文仍不会泄露给未授权实体, 即数据内容机密性不受大语言模型行为偏离的影响.

3.3 安全模型

本节在 BPR 模型 [12] 的基础上进行拓展, 以形式化定义本文协议的安全目标. 通过模拟攻击者在协议执行中的交互能力, 我们明确给出了本文协议应满足的会话密钥安全性与数据机密性. 主要关注两类安全目标: 一是用户与外部服务器之间基于口令的认证密钥交换 (PAKE) [11] 的安全性与认证性; 二是在上述会话密钥与密文策略属性基加密 (CP-ABE) [46] 共同保护下, 用户数据在智能体场景中的机密性与细粒度访问控制.

协议参与者. 涉及三类协议参与者: 用户 U、大语言模型 L 及外部工具服务器 S. 每个参与者可以拥有多个会话实例 [47], 记用户 U 的第 i 个实例为 U^i, 服务器 S 的第 j 个实例为 S^j. 特别地, 大语言模型 L 被建模为不持有任何解密密钥的中间调度实体, 仅参与明文查询解析、子任务分解与密文结果填充. 因此在安全模型中可将其视为公开信道上的一部分, 其可观察、转发甚至篡改消息的能力统一纳入到攻击者的信道控制能力中, 而不为其单独设立专用的密钥或安全假设. 敌手 $\mathcal{A}$ 与参与方 $P \in U \cup S$ 间的交互通过预言查询来模拟.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

长期密钥. 每个用户 U 拥有口令 $PW_U \in \mathcal{D}$, 其中 $\mathcal{D}$ 为服从 Zipf 分布 [48] 的口令空间, 用户与服务器基于口令 PW_U 进行密钥协商得到共享会话密钥 sek; 外部工具服务器 S 存储 $\{ID_U, Cre\}$, 其中 Cre 是与口令绑定的加密消息. 同时用户和外部工具服务器持有由属性授权中心分发的属性私钥 usk 和 ssk.

会话与会话标识. 当参与者实例被激活以执行协议时, 一个会话开始. 每个会话由唯一的会话标识符 sid 标识, 会话标识 sid 可由参与双方身份及其在本轮会话中交换的协议消息共同确定.

接受/拒绝. 协议执行结束时, 若成功输出会话密钥 sek 及对端身份标识, 则称该实例接受, 否则称为拒绝.

伙伴关系 (匹配会话). 两个实例 U^i 和 S^j 被称为合作伙伴, 伙伴标识符 pid 为预期的通信对方, 当且仅当: 1) 这两个实例均已接受; 2) 这两个实例拥有相同的会话标识符 sid; 3)U^i 的伙伴标识符 $pid_U^i = S^j, S^j$ 的伙伴标识符 $pid_S^j = U^i$; 4) 这两个实例确认对方为预期的通信伙伴; 5) 它们计算出了相同的会话密钥 sek.

敌手 $\mathcal{A}$ 被建模为一个概率多项式时间算法, 可以通过以下查询与系统交互, 模拟现实中的攻击能力:

-Execute(U^i, S^j): 模拟攻击者的被动攻击 (窃听) 能力, 输出为协议正常运行情况下 U^i 与 S^j 的交互信息;

-Send(U^i, msg): 该查询模拟攻击者的主动攻击 (如消息注入、篡改和重放), $\mathcal{A}$ 向实例 P^i(其中 $P \in \{U, S\}$ 发送消息 msg, P^i 根据协议规定向 $\mathcal{A}$ 返回处理 msg 所产生的相应的信息;

-Reveal(P^i): 模拟会话密钥的泄露, 若 P^i 已生成会话密钥 sek, 则查询输出 sek; 否则输出错误标识 $\perp$.

-Corrupt(P, s): 该查询模拟长期密钥的泄露. 若 $P = U, s = 1$, 返回用户的口令 PW_U; 若 $P = U, s = 2$, 返回用户的属性密钥 usk; 若 $P = S, s = 1$, 返回服务器后台数据库存储的 $\{ID_U, Cre\}$; 若 $P = S, s = 2$, 返回服务器端的属性密钥 ssk. 针对同一个实例, 攻击者至多进行以上两种询问中的一种.

-Test (U^i): 该查询并非模拟敌手 $\mathcal{A}$ 的攻击能力, 而是用来刻画会话密钥 sek 的语义安全性, 此查询在敌手攻击游戏中使用一次. 如果实例 P^i 是新鲜的且已生成会话密钥 sek, 则预言机抛掷一枚公平硬币 c. 若 $c = 1$, 向 $\mathcal{A}$ 输出 sek; 若 $c = 0$, 返回一个与 sek 等长的、从密钥空间中均匀随机选取的字符串.

新鲜性. 一个会话实例被称为新鲜实例当且仅当同时满足以下条件: 1) 实例 P^i 未被发起 Reveal 查询, 且若其存在伙伴实例, 则伙伴实例也未被发起 Reveal 查询; 2) 攻击者未在测试会话完成之前获得与该会话对应的口令相关长期秘密; 3) 攻击者未同时获得与该会话相关的用户口令和能够满足对应访问策略的合法属性私钥; 4) 若口令泄露发生在测试会话接受之后, 则该实例仍可被视为新鲜, 以刻画协议的前向安全性.

该定义体现了本文所考虑的部分密钥泄露威胁: 攻击者可以在特定时刻获得用户口令或参与实体的属性私钥, 但不能同时获得, 即不能在同一会话或同一攻击场景中同时获取二者. 若攻击者同时掌握用户口令和满足挑战访问策略的属性私钥, 则认证层与访问控制层均被突破, 此时对应的测试会话将不再被视为安全.

定义 1 (会话密钥安全性) 敌手 $\mathcal{A}$ 可以自适应地发起上述除 Test 外的任意查询. 最终, $\mathcal{A}$ 对一个新鲜实例发起 Test 查询, 并输出猜测值 b'. 记敌手攻击成功事件为 Succ, 定义 $\mathcal{A}$ 攻破本文协议 (简记为 $\mathcal{P}$) 的优势为

$$\mathsf{Adv}_{\mathcal{P}}^{\mathsf{AKE}}(\mathcal{A}) = |2\Pr[\mathsf{Succ}] - 1| = |2\Pr[c' = c] - 1|.$$

对于任何概率多项式时间的敌手 $\mathcal{A}$, 若攻破协议 $\mathcal{P}$ 的优势 $\mathsf{Adv}_{\mathcal{P}}^{\mathsf{AKE}}(\mathcal{A})$ 是可以忽略的, 则称 $\mathcal{P}$ 满足会话密钥安全性, 表明对于任意新鲜实例, 敌手 $\mathcal{A}$ 无法有效区分其真实会话密钥与一个独立均匀随机的字符串, 从而保证用户与目标工具服务器之间协商得到的会话密钥 sek 在计算上对敌手 $\mathcal{A}$ 保密.

口令认证安全性. 由于用户口令属于低熵秘密, 敌手始终可以通过与协议实例的主动交互发起在线口令猜测攻击. 如果对至多执行 q_{send} 次在线猜测攻击的任意概率多项式时间敌手 $\mathcal{A}$, 都存在可忽略函数 $\varepsilon(\cdot)$, 满足

$$\mathsf{Adv}_{P,\mathcal{D}}^{\mathsf{AKE}}(\mathcal{A}) \leqslant C' \cdot q_{send}^{s'} + \varepsilon(\kappa).$$

我们称协议 $\mathcal{P}$ 是安全的口令认证密钥协商协议, 能够实现用户与外部工具服务器之间的双向认证. 其中, $\mathcal{D}$ 为服从 Zipf 分布的口令空间, C' 和 s' 为 Zipf 参数 [48], 由大规模口令数据集拟合得到, κ 为系统安全参数.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

数据机密性与访问控制安全. 对于数据传输阶段, 本文采用基于访问策略的挑战密文安全模型刻画密文机密性. 攻击者可自适应地请求多个属性集合对应的私钥, 但所有查询的属性集合均不得满足挑战策略 Γ^*. CP-ABE 在选择明文攻击下的安全性可形式化为下列游戏. 设有挑战者 $\mathcal{C}$ 和敌手 $\mathcal{A}$, 进行预言机查询.

-Setup: 挑战者 $\mathcal{C}$ 运行算法 $\mathsf{Setup}(\lambda)$, 产生系统的公共参数 mpk 和主密钥 msk, 将 mpk 发送给 $\mathcal{A}$.

-Query 1: $\mathcal{A}$ 可以适应性地向 $\mathcal{C}$ 发起一系列属性密钥查询, 对每个查询的属性集合 $Attr_j$(这里 $Attr_j$ 不应满足挑战访问结构 Γ^*), $\mathcal{C}$ 运行 $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, Attr_j)$ 算法, 将生成的解密密钥 sk_{Attr_j} 发送给 $\mathcal{A}$.

-Challenge: 询问阶段 1 结束后, $\mathcal{A}$ 提交两个等长消息 msg_0 和 msg_1, $\mathcal{C}$ 随机选择一个比特 $b \in \{0,1\}$, 运行 $\mathsf{ABEnc}(mpk, msg_b, \Gamma^*)$ 算法, 生成挑战密文 CT^*, 并将 CT^* 发送给 $\mathcal{A}$.

-Query 2: 重复 Query 1, 每个查询的属性集合 $Attr_j$ 均不满足与该挑战对应的访问结构 Γ^*.

-Guess: $\mathcal{A}$ 输出一个比特 b' 作为猜测结果. 若 $b' = b$, 则判断 $\mathcal{A}$ 在上述游戏中获胜, 定义获胜优势为

$$\mathsf{Adv}_{\mathcal{P}}^{\mathsf{CP-ABE}}(\mathcal{A}) = |2\mathsf{Pr}[\mathsf{Succ}] - 1| = |2\Pr[b' = b] - 1| .$$

定义 2 (IND-CPA **安全**) 若对于任意 PPT 敌手 $\mathcal{A}$, 上述优势均为可忽略函数, 则称协议 $\mathcal{P}$ 在选择明文攻击 (Indistinguishability under chosen plaintext attack, IND-CPA) 下是安全的. 我们定义即使敌手能够观察由大语言模型 L 转发的输入密文和输出密文, 并获得若干不满足挑战策略的属性私钥, 仍无法区分挑战密文中加密的具体明文. 由于访问策略直接绑定于密文, 因此该性质同时刻画了数据级的细粒度访问控制安全.

4 具体方案

本节提出了一个基于大语言模型智能体的数据安全共享方案, 通过口令和属性基加密实现了用户和智能体之间身份认证和细粒度的访问控制. 如图 2 所示, 用户首先与外部服务器通过口令认证密钥协商协议协商出一个高熵会话密钥 sek, 随后用户利用共享密钥 sek 和访问控制策略 Γ_A 对其私有输入数据 msg_A 进行加密, 得到密文 CT. 用户进一步将明文查询 prom 与加密后的输入 CT 一并发送给大语言模型 L. L 将明文查询 prom 解析为多个子任务 $\mathsf{task_i} = (\mathsf{aid}, \mathsf{goal}, \mathsf{inputs})$, 即 “调用哪个工具 (aid)、让它干什么 (goal)、用什么数据干 (inputs), 并生成响应模板 temp, 模板 temp 使 L 只负责理解需求、设计回答结构, 而把具体计算完全外包给外部服务器. 需要强调的是, 请求解析与响应模板生成过程与原始流程 (即图 1 中的步骤) 相同, 只是任务的输入由明文变为密文, 即 $\mathsf{inputs} = CT$. L 将解析得到的各个任务发给外部服务器. 服务器得到加密数据上 CT 后利用其属性私钥 ssk 和共享密钥 sek 进行解密, 得到用户输入的请求数据明文 msg_A 后进行工具推理. 推理完成后, 外部服务器利用共享密钥 sek 和访问控制策略 Γ_B 将工具端最终输出的响应 msg_B 进行加密, 得到密文 C_i, 返回给 L. L 将这些加密结果填入响应模板 temp 中, 从而得到完整响应结果密文 CF. 最后, 用户使用自己的属性私钥 usk 和共享密钥 sek 对密文 CF 进行解密, 恢复响应结果 msg_B.

本协议包含了 5 个阶段: 系统初始化阶段, 用户注册阶段, 认证与密钥协商阶段, 数据上传阶段, 数据下载阶段. 方案 [2)]中所使用的符号缩写及其含义见表 2 , 具体构造如下所述.

4.1 系统初始化阶段

协议输入安全参数 1^λ, 定义一个阶为素数 p, 生成元为 g 的循环群 G, 选择五个安全哈希函数 $H : \{0,1\}^* \to \mathbb{Z}_p^*, H_0 : \{0,1\}^* \to G, H_i : \{0,1\}^* \to \{0,1\}^{\lambda_i}, i = 1,2,3$, 其中 λ_i 为哈希函数 H_i 输出的比特长度, 用来表示认证元 A_s、认证元 A_u、会话密钥 sek 三者之间的独立性. 定义 AuthEnc/AuthDec 为安全的认证加密及其解密算法, ABEnc/ABDec 为安全的属性基加密及其解密算法, KE 为底层认证密钥交换过程 (即通信双方利用各自的长期公私钥对以及会话相关的临时公私钥对交互协商生成共享会话密钥). 算法随机选择

2) 本文中 “方案” 和 “协议” 表达相同的概念, 交替使用

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

表 2 符号定义.
Table 2 Notation definitions.

Symbol	Definition	Symbol	Definition	Symbol	Definition
ID_u, PW	Identity and password	Cre	Encrypted credential	prom	User prompts
usk, ssk	Attribute private keys	$Attr$	Attribute set	task_i	Subtask
msg_A, msg_B	Inputs and response	sek	Session key	temp	Response template
Γ_A, Γ_B	Access policies	k_s	OPRF key	KE	Key exchange

$k_s, u, v \in \mathbb{Z}_p^*$, 并将 k_s 设置为 OPRF 密钥, 由服务器端进行安全存储, 参数 u, v 进行公开. 时间被分成多个周期, 外部服务器保存一个列表 honeylist, 该列表记录每个用户向其请求的次数, 每个周期开始时初始化为 0, 并设定 n_0 为每个周期内用户可请求次数的上限. 系统输出公共参数 $PP = \{G, p, g, u, v, H, H_0, H_1, H_2, H_3\}$.

4.2 用户注册阶段

该阶段是用户 U 首次在外部工具服务器 S 上建立账户和关联凭证的过程. 在服务器 S 不接触用户口令明文的前提下, 为用户生成可验证的账户凭证. 最终, 服务器中仅保存一个由口令派生的密码文件, 从而实现口令的零知识存储, 为系统提供抗泄露安全性, 具体步骤如下:

Step R1. $U \Rightarrow S : \{ID_u, \alpha, A\}$. 用户 U 选择身份标识 ID_u, 口令 PW 和随机数 $r, a \in \mathbb{Z}_p^*$, 计算 $\alpha = H_0(PW)^r, A = g^a$, 随后将消息发送给外部工具服务器 S.

Step R2. $S \Rightarrow U : \{\beta, B\}$. 外部工具服务器 S 在时间 T 接收到用户 U 的注册请求后, 选择随机数 $b \in \mathbb{Z}_p^*$, 计算 $\beta = \alpha^{k_s}, B = g^b$, 其中 k_s 为服务器端为用户生成的 OPRF 密钥.

Step R3. $U \Rightarrow S : \{Cre\}$. 用户 U 收到外部服务器 S 的 $\{\beta, B\}$ 后, 令 $\mathsf{RW} = H\left(PW \| \beta^{1/r}\right) = H\left(PW \| H_0(PW)^{k_s}\right)$, 计算 $Cre = \mathsf{AuthEnc}_{\mathsf{RW}}(a, A, B)$ 发送给服务器 S.

Step R4. 外部服务器 S 在注册表中为用户 U 创建一条新的表项, 并存储 $\mathsf{file[sid]} = \{ID_u, Cre, A, B, b, T_{reg} = T, \mathsf{honeylist} = 0\}$ 在注册表中. 其中, T_{reg} 为当前的时间戳.

4.3 认证与密钥协商阶段

该阶段是基于口令的双向认证与密钥交换过程. 用户与外部工具服务器通过交换并验证与口令相关的凭证, 在确认彼此身份同时, 协同生成一个具有前向安全性的会话密钥, 为后续的数据安全传输提供了安全信道.

Step A1. $U \Rightarrow S : \{ID_u, \alpha', X\}$. 用户 U 选择身份标识 ID_u, 口令 PW 和随机数 $r', x \in \mathbb{Z}_p^*$, 计算 $\alpha' = H_0(PW)^{r'}, \quad X = g^x$, 随后将消息发送给外部工具服务器 S.

Step A2. $S \Rightarrow U : \{\beta', Y, Cre, A_s\}$. 外部服务器 S 接收到用户 U 的登录请求后, 首先检查 $\alpha' \in G$, 并检索本地存储的会话文件 $\mathsf{file[sid]} = \{ID_u, Cre, A, B, b, T_{reg} = T, \mathsf{honeylist} = 0\}$, 若存在, 选择随机数 $y \in \mathbb{Z}_p^*$, 计算 $\beta' = (\alpha')^{k_s}, Y = g^y$, 令 $K = \mathsf{KE}(b, y, A, X), A_s = H_1\left(\alpha' \| g^K\right)$, 其中 k_s 为服务器 S 为用户 U 生成的 OPRF 密钥, KE 为底层认证密钥交换函数; 若 $\alpha' \notin G$, 终止该会话, 同时令 $\mathsf{honeylist} = \mathsf{honeylist} + 1$, 且当 honeylist 的值超过阈值 n_0(代表每个周期内用户可请求次数的上限) 时, 则冻结该用户的账户, 直至用户 U 重新注册.

Step A3.$U \Rightarrow S : \{A_u\}$. 用户 U 收到外部服务器 S 的 $\{\beta', Y, Cre, A_s\}$ 后, 令 $\mathsf{RW}' = H\left(PW \| (\beta')^{1/r'}\right) = H\left(PW \| H_0(PW)^{k_s}\right)$, 计算 $\mathsf{AuthDec}_{\mathsf{RW}'}(Cre) = (a, A, B)$, $K = \mathsf{KE}(a, x, B, Y), A_s^* = H_1\left(\alpha' \| g^K\right)$, 并验证 A_s^* 是否等于服务器 S 发送的 A_s, 若相等, 表明 U 成功认证 S, 并计算认证元 $A_u = H_2\left(\alpha' \| g^K\right)$ 给服务器 S.

Step A4. 外部工具服务器 S 收到用户 U 的认证元 A_u 后, 计算 $A_u^* = H_2\left(\alpha' \| g^K\right)$, 判断 $A_u^* = A_u$ 是否成立. 若相等, S 成功认证 U; 否则, 服务器 S 立即终止当前会话, 拒绝后续任何数据请求.

Step A5. 用户 U 和外部工具服务器 S 协商出会话密钥 $sek = H_3\left(\alpha' \| g^K\right)$, 用于后续安全通信.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

4.4 属性密钥分发阶段

在本方案中, 我们引入属性基加密机制实现细粒度访问控制. 为了兼顾安全性与可信性, 在生成属性密钥的这一阶段引入一个可信的属性授权实体, 该实体由平台身份与访问控制模块统一管理. 属性私钥经独立安全信道分发给各参与方, 一旦密钥下发完毕属性授权实体即退出协议流程, 后续会话不再依赖该机构. 在属性空间设计上属性授权实体与系统管理员共同维护全局属性集合. 该集合主要包含两类属性: 一类用于描述用户的身份特征与权限范围 (例如 “所属机构: 高校”“等级:plus 会员”); 另一类用于描述大语言模型智能体所调用工具的功能类型及访问权限 (例如模型类型: 文本嵌入, 服务等级: 高级, 授权任务: 情感分析). 为避免歧义并便于后续策略匹配, 系统为每个属性分配唯一标识符.

Step K1. 属性授权实体根据安全参数 λ, 生成系统主密钥 mpk 与系统主私钥 msk. 系统主密钥 mpk 公开给所有系统参与者 (包括用户与外部工具服务端), 而 msk 由属性授权实体秘密存储. 对于每一个合法实体 $P \in \{U, S\}$, 在其通过口令认证与密钥协商协议建立安全会话之后, 属性授权实体对参与者身份与权限信息进行校验, 得到属性集合 $Attr(P)$, 如用户的角色、组织、数据敏感级别, 或调用工具的功能类型、处理范围等.

Step K2.用户属性私钥生成. 当用户 U 完成注册与身份验证后, 需向可信机构提交其权威属性声明. 可信机构验证声明的真实性与有效性后, 基于 msk 和该用户被验证的属性集合 $Attr(U)$, 调用 $\mathsf{KeyGen}(mpk, msk, Attr(P))$ 生成对应的属性私钥 usk, 并通过由会话密钥 sek 保护的认证加密信道安全传送给用户 U.

Step K3. 外部服务器属性私钥生成. 每个工具在纳入系统服务池前, 由其提供者向属性授权实体提交注册信息, 包括其功能描述、服务级别协议及允许访问的数据类别等元数据. 属性授权实体对其进行检测和验证后, 将元数据映射为形式化的属性集合 $Attr(S)$, 并同样运行 $\mathsf{KeyGen}(mpk, msk, Attr(P))$ 算法, 生成外部服务器 S 的属性私钥 ssk, 通过安全信道分发给该智能体的调用工具服务器, 用于后续用户输入的数据密文解密.

关键安全假设与性质: 本设计遵循密文策略属性基加密模型 [46]. 属性授权实体被假设为完全可信, 且不会滥用主密钥生成越权属性私钥. 用户与外部服务器的私钥均直接编码了其被授权的属性, 任何实体无法自行生成或篡改密钥中的属性授权. 私钥的生成过程确保了 “抗串谋” 安全, 即多个持有不同属性私钥的实体无法通过合并密钥来获得超出其各自属性范围的访问能力. 属性私钥 usk 和 ssk 在语义上与各自的属性集合绑定, 仅当访问策略 Γ 对某实体的属性集合满足 $\Gamma(Attr(P)) = 1$ 时, 该实体才能从相应的 ABE 密文中解封装出数据密钥或结果密钥; 否则即使获取到密文也无法恢复明文. 通过上述设计, 一方面避免了大语言模型智能体本身直接持有任何属性私钥, 保证其 “可用而不可见” 的角色定位; 另一方面在密码学层面实现了基于属性的细粒度访问控制, 将后续的密文计算与响应生成严格约束在通过口令认证且属性满足访问策略的实体范围内.

4.5 数据请求阶段

在数据上传之前, 用户 U 和外部服务器 S 已经共同协商出会话密钥 sek, 结合属性基加密, 在不泄露原始数据明文的前提下, 将计算任务委托给大语言模型智能体的后端 AI 模型, 外部服务器 S 首先在满足属性基访问控制策略的前提下, 对经大语言模型 L 分发的任务输入进行解密与推理. 具体步骤如下:

Step U1. $U \Rightarrow L : \{\mathsf{prom}, \Gamma_A, CT\}$, 用户 U 首先输入私有数据 msg_A, 随机选择 $\gamma \in \mathbb{Z}_p^*$, 计算 $C_M = msg_A \cdot g^{H(\gamma \| usk)}, D = g^{H(\gamma \| usk)} \cdot u^{H(sek)}$, 其中 usk 为用户持有的属性私钥, usk 与随机数 γ 共同哈希后作为指数掩码, 将解密能力与用户的属性授权及当前会话上下文进行绑定. sek 为上述认证与密钥协商阶段生成的会话密钥. 设置其访问控制策略为 Γ_A(该策略定义了允许处理此数据的外部工具所需满足的属性条件). U 生成密文 $C_A = \mathsf{ABEnc}_{\{mpk, \Gamma_A\}}(D)$, 令 $CT = \{C_A, C_M\}$, 最后用户 U 发送消息 $\{\mathsf{prom}, \Gamma_A, CT\}$ 给 L, 其中 prom 为明文查询请求, 用于描述计算任务的自然语言指令 (如 “分析该数据的情感倾向”).

Step U2. $L \Rightarrow S : \{\mathsf{task}_i\}_{i \in [m]}$: 大语言模型 L 智能体收到用户 U 的数据请求后, 基于查询请求 prom 解析任务逻辑, 理解用户意图, 将解析得到的各个任务 $\mathsf{task_i} = (\Gamma_\mathsf{A}, \mathsf{goal}, \mathsf{CT})$ 分发给外部服务器 S, 其中 $i \in [m], m$

表示调用的工具数量, Γ_A 表示允许处理该子任务的外部工具必须满足的属性条件, goal 表示任务目标. L 仅接触到查询请求 prom 及与之关联的数据密文与策略描述, 而无法直接恢复私有输入数据 msg_A.

Step U3. 外部工具服务器 S 接收到来自 L 的任务 $\{\mathsf{temp}, \mathsf{task}_i\}_{i\in[m]}$ 后, 检测自身属性集 $Attr(S)$ 是否满足访问策略 Γ_A, 若不满足, 推理任务终止; 否则, S 通过自己的属性私钥 ssk 解密 C_A 得到 $D = \mathsf{ABDec}_{\{ssk\}}(C_A)$, 计算 $g^{H(\gamma\|usk)} = \frac{D}{u^{H(sek)}}$, 得到用户明文数据 $msg_A = \frac{C_M}{g^{H(\gamma\|usk)}}$, 其中 sek 为共享的会话密钥.

4.6 数据响应阶段

该阶段描述智能体对用户数据的响应流程, 始于外部工具完成对密文数据的推理计算, 止于用户成功恢复并验证明文响应, 整个过程中, 所有中间结果 (包括工具的输出、聚合后的响应等) 均以密文形式存在, 确保对 L 及任何未授权方保密, 实现了计算结果在传输与聚合过程中的机密性.

Step R1. $S \Rightarrow L : \{\Gamma_B, C_i\}$: 各外部工具在完成对用户数据 msg_A 的推理后, 生成明文结果 msg_B. 随后, 工具服务器 S 随机选择 $\tau \in \mathbb{Z}_p^*$, 计算 $C_R = msg_B \cdot g^{H(\tau\|ssk)}$, $F = g^{H(\tau\|ssk)} \cdot v^{H(sek)}$, 设置其访问控制策略为 Γ_B(该策略定义了允许获取此结果的用户所需满足的属性条件). S 生成密文 $C_B = \mathsf{ABEnc}_{\{mpk,\Gamma_B\}}(F)$, 令 $C_i = \{C_B, C_R\}$, S 收集到所有子任务对应的加密结果集合后发送消息 $\{\Gamma_B, C_i\}$ 给大语言模型 L.

Step R2. $L \Rightarrow U : \{\Gamma_B, CF\}$: 大语言模型 L 智能体作为协调者, 将收到的加密结果 C_i 逐一填入之前生成的响应模板 temp 的对应位置, 得到聚合后的加密结果 CF. 在此过程中, L 仅能处理密文形式的中间结果, 而无法获知任何具体的推理结果数据 msg_B, 从而实现了 “计算可知、结果不可见” 的安全目标.

Step R3. 用户端 U 收到 L 的加密响应 $\{\Gamma_B, CF\}$ 后, 执行解密算法 $\mathsf{ABDec}(\cdot)$. 解密成功的充要条件是: 1) 用户属性 $Attr(U)$ 满足外部服务器所设定的策略 Γ_B; 2) 用户持有正确的会话密钥 sek. 用户端 U 通过自己的属性私钥 usk 解密 C_B 得到 $F = \mathsf{ABDec}_{usk}(C_B)$, 计算 $g^{H(\tau\|ssk)} = \frac{F}{v^{H(sek)}}$, 得到推理后的明文数据 $msg_B = \frac{C_R}{g^{H(\tau\|ssk)}}$, 其中 sek 为共享的会话密钥, 用户 U 将收到的响应密文解密为可读的最终答案.

本阶段协议通过 CP-ABE 机制实现了以下安全属性: 1) 结果机密性, 所有响应结果在传输与返回过程中均以密文形式存在; 访问控制一致性, 结果的解密权与数据的加密策略严格匹配; 用户端可验证性, 成功的解密本身即是对计算执行者的间接身份认证, 若响应能被用户的有效属性私钥和会话密钥正确解密, 则证明该响应确实是由满足策略的授权工具生成且未被篡改; 反之, 任何伪造或篡改都将导致解密失败并立即被用户察觉.

本文方案可自然扩展至多任务链场景, 其核心思想是将链式调用分解为多个顺序执行的 “加密、解密、再加密” 单轮环节. 设任务链包含 n 个工具 $S_1, S_2, \cdots, S_n$, 其中 S_i 的输出 msg_{B_i} 需作为 S_{i+1} 的输入. 在此场景下, 用户 U 在初始请求中除提交首级访问策略 Γ_{A_i} 外, 还可通过任务描述附加各级策略约束, 或由各级工具在返回结果时动态指定下一级策略 $\Gamma_{A_{i+1}}$, 访问策略按任务需求重新设置. 具体而言本文支持以下两种方式: 1) 策略继承: 后续工具可以沿用用户初始设定的访问策略 Γ_{A_i}, 即中间结果继续受同一策略保护. 此时, 后续工具若满足 Γ_{A_i}, 即可直接解密上一工具的输出并继续计算, 无需重新加密. 该方式适用于整条任务链中各工具具有相同属性授权的场景. 2) 策略更新: 若下级工具需要更细粒度的权限控制, 当前工具在加密中间结果时可设置新的访问策略 $\Gamma_{A_{i+1}}$, 该策略限定了哪些后续工具可以访问该中间结果. 此时, 中间结果需要重新加密后再传递给下级. 策略的更新由工具根据任务上下文或用户预定义的策略映射规则执行. 该方式能够在不改变本文基本协议结构的前提下, 将单轮调用扩展至复杂工具链场景, 同时避免了访问权限在任务链中被非法扩大或滥用.

5 安全性证明

本节在前述安全模型基础上, 对方案的安全性进行形式化分析. 为便于表述, 记整体协议为 $\mathcal{P}$.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

定理 1. 设 $\mathcal{P}$ 是本文提出的协议, G 为阶为 p 的循环群, 且 CDH 问题在 G 上是困难的; 设随机预言机 H 可被攻击者查询, 底层 CP-ABE 机制 $\mathsf{ABEnc}(\cdot)$ 满足 IND-CPA 安全与抗串谋性, $\mathcal{D} \subseteq \mathbb{Z}_p^*$ 是一个规模为 $|\mathcal{D}|$, 且服从 Zipf 分布的口令空间. $\mathcal{A}$ 表示在时间界限 t 中进行至多 q_{send} 次 Send 查询、q_{exe} 次 Execute 查询、q_H 次随机预言机查询来破坏协议安全性的 PPT 攻击者. 我们有

$$\begin{aligned}\mathsf{Adv}_{\mathcal{P}}(\mathcal{A}) \leqslant & C' \cdot q_{send}^{s'} + 2q_H \cdot \mathsf{Adv}_G^{\mathsf{CDH}}\left(t + 2q_{exe}\tau_e + 3\tau_e\right) + 2q_H^2 \cdot \mathsf{Adv}_G^{\mathsf{CDH}}\left(t + 3\tau_e\right) \\ & + 2\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP-ABE}}\left(t, 2q_{exe}\right) + \frac{\left(q_{send} + q_{exe}\right)^2}{p}.\end{aligned}$$

其中 τ_e 表示单次指数运算所需的时间, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP-ABE}}\left(t, 2q_{exe}\right)$ 定义为敌手在时间界 t 内、最多执行 $2q_{\mathrm{exe}}$ 次查询时攻破底层 CP-ABE 方案的优势. 后续 CP-ABE 的安全性采用了标准的 “挑战密文嵌入” 归约方法, 可确保任意不满足挑战访问策略的敌手均无法以非忽略优势区分挑战密文内所含消息.

证明. 为了证明定理结论, 我们构造了一组混合仿真游戏, 记作 $\mathsf{Game}_0, \mathsf{Game}_1, \ldots, \mathsf{Game}_4$. 其中, Game_0 模拟的是真实的协议攻击场景. 从 Game_0 出发, 后续的每个游戏都会逐步修改预言机的响应方式, 直到最后一个游戏 Game_4, 在这个游戏中敌手 $\mathcal{A}$ 的优势被限制为零. 所有游戏里, 预言机都严格按照协议规范来应答各类查询. 对于每个游戏 Game_n(其中 n 取 0 到 4) , 我们记事件 Game_n 为敌手成功猜中 Test 查询里所用随机比特 c 的情况. 接下来的证明中, 我们会逐一分析任意两个相邻实验之间敌手优势的差值, 并由此推算出真实攻击中敌手优势的上限.

Game_0: 这一实验对应于实际的协议攻击过程, 也就是真实协议的执行情形. 在此游戏中, 敌手可根据安全模型的规定, 自适应地发起 Execute、Send、Reveal、Corrupt 以及 Test 等查询. 根据定义, 可以得到

$$\mathsf{Adv}_{\mathcal{P}}(\mathcal{A}) = \left|2\Pr\left[\mathsf{Succ}_0\right] - 1\right|.$$

Game_1: 在此实验中, 所有哈希预言机及 Game_3 中所使用的哈希函数 $H_i(i = 0, 1, 2, 3)$ 均采用哈希表 $L_{\mathcal{H}}$ (并另设 $L_{\mathcal{A}}$ 记录敌手直接发起的哈希查询) 来模拟. 我们按照真实攻击规则执行 Send、Execute、Reveal 和 Test 模拟查询. 显然, 本游戏下敌手 $\mathcal{A}$ 的优势与其在真实协议攻击中不可区分. 从而可得

$$\Pr\left[\mathsf{Succ}_1\right] = \Pr\left[\mathsf{Succ}_0\right].$$

Game_2: 这一实验里, 各种查询询问的模拟规则与 Game_1 中的保持一致, 但会去除掉那些在协议交互消息 $((CT, \Gamma_A)\,(CF, \Gamma_B)$ 上发生了碰撞的会话实例, 也就是那些在 $((D^*, \Gamma_A)\,(F^*, \Gamma_B))$ 里出现碰撞的会话情况. 因为每个会话实例里至少有一个参与实体是诚实的, 所以 D^* 或 F^* 二者之一属于随机选取. 依据生日攻击原理, 可以得到下面这个不等式:

$$\left|\Pr\left[\mathsf{Succ}_2\right] - \Pr\left[\mathsf{Succ}_1\right]\right| \leqslant \frac{\left(q_{\mathsf{send}} + q_{\mathsf{exe}}\right)^2}{2p}.$$

Game_3: 此游戏中对密文的解密过程做了改动, 不再直接使用原来的 H, 而是引入一个按下面方式模拟出来的私有谕言 H′ 来替代它. 攻击者能区分开 Game_2 和 Game_3 的唯一方法是进行如下询问:

$$H\left(\mathsf{CDH}\left(F^*/v^{H(sek)}, D^*/u^{H(sek)}\right), sek, \mathsf{ABEnc}_{(mpk,\Gamma_A)}\left(D^*\right), \Gamma_A, \mathsf{ABEnc}_{(mpk,\Gamma_B)}\left(F^*\right), \Gamma_B\right),$$

并将询问结果与 $H'\left(\mathsf{ABEnc}_{(mpk,\Gamma_A)}\left(D^*\right), \Gamma_A, \mathsf{ABEnc}_{(mpk,\Gamma_B)}\left(F^*\right)\right)$ 进行比较. 记上述事件为 AskH_3, 我们有

$$\Pr\left[\mathsf{AskH}_3\right] \leqslant q_H \cdot \mathsf{Adv}_{\mathsf{G}}^{\mathsf{CDH}}\left(t + 2q_{exe}\tau_e + 3\tau_e\right).$$

我们利用 Diffie-Hellman 问题的随机自归约性质来构造模拟器. 给定一个 CDH 实例 (W, Z)(即 $W = g^w, Z = g^z$, 其中 w, z 未知), 按下述方式修改 Execute 查询: 随机选取 $x, y \in \mathbb{Z}_p^*$, 并计算 $D = W \cdot g^x, F = Z \cdot g^y$.

当事件 AskH(sek)4 发生时, $\mathsf{CDH}(X,Y)$ 就在列表 $L_{\mathcal{H}}$ 中, 从而可得 $\mathsf{CDH}(W,Z)=\frac{\mathsf{CDH}(D,F)}{W^y\cdot Z^x\cdot g^{xy}}$. 因此, 我们有

$$|\Pr[\mathsf{Succ}_3]-\Pr[\mathsf{Succ}_2]|\leqslant\Pr[\mathsf{AskH}_3]\leqslant q_H\cdot\mathsf{Adv}_{\mathsf{G}}^{\mathsf{CDH}}(t+2q_{exe}\tau_e+3\tau_e).$$

Game_4: 这个实验的目的是模拟攻击者的主动攻击. 为此, 我们按照如下方法模拟 Execute 查询: $D^*=D, F^*=F$. 在这个实验中, 我们定义事件 AskH(sek)4 为攻击者直接进行下述询问:

$$H\left(\mathsf{CDH}\left(F^*/v^{H(sek)},D^*/u^{H(sek)}\right),sek,\mathsf{ABEnc}_{(mpk,\Gamma_A)}(D^*),\Gamma_A,\mathsf{ABEnc}_{(mpk,\Gamma_B)}(F^*),\Gamma_B\right).$$

在安全归约中, 除非事件 AskH_4 发生 (即敌手主动查询了与挑战密文相关的 CDH 哈希值), 否则实验 Game_3 与 Game_4 在敌手视角下是计算不可区分的. 这是因为 Game_4 中所有公开参数和预言机响应均按照与 Game_3 相同的分布进行模拟, 唯一的差异被 AskH_4 事件所覆盖, 敌手无法获得任何关于挑战比特的信息, 因此其成功猜对的概率恰好为 $1/2$. 综合上述分析, 可得以下不等式:

$$|\Pr[\mathsf{Succ}_4]-\Pr[\mathsf{Succ}_3]|\leqslant\Pr[\mathsf{AskH}_4],\Pr[\mathsf{Succ}_4]=\frac{1}{2}.$$

为便于分析, 我们终止以下这些会话: 对在一次主动攻击中使用的 (D^*,F^*), 有两个不同的共享密钥 $sek_0\neq1$ 和 $sek_1\neq1$, 使得在列表 $L_{\mathcal{H}}$ 中存在记录 $\{\mathsf{CDH}\left(F^*/v^{H(sek_0)},D^*/u^{H(sek_1)}\right),H(sek),\mathsf{ABEnc}_{(mpk,\Gamma_A)}(D^*),\Gamma_A,\mathsf{ABEnc}_{(mpk,\Gamma_B)}(F^*),\Gamma_B\}$. 我们记上述事件为 Coll, 并有下述结论: $\Pr[\mathsf{Coll}]\leqslant q_H\cdot\mathsf{Adv}_G^{\mathsf{CDH}}(t+2\tau_e)$. 假设存在一个会话中的 (D^*,F^*) 以及 $sek_0\neq sek_1\neq1$, 使得在 $L_{\mathcal{H}}$ 中有记录 $\{\mathsf{CDH}\left(F^*/v^{H(sek_0)},D^*/u^{H(sek_1)}\right),H(sek),\mathsf{ABEnc}_{(mpk,\Gamma_A)}(D^*),\Gamma_A,\mathsf{ABEnc}_{(mpk,\Gamma_B)}(F^*),\Gamma_B\}$. 则对 $i=0,1$, 我们可以计算

$$T_i=\mathsf{CDH}\left(F^*/v^{H(sek_i)},D^*/u^{H(sek_i)}\right)=\frac{\mathsf{CDH}(D^*,F^*)^{H(sek_i)}\cdot\mathsf{CDH}(u,v)^{H(sek_i)}}{\mathsf{CDH}(u,F^*)^{H(sek_i)}\cdot\mathsf{CDH}(v,D^*)^{H(sek_i)}}.$$

由于在会话中已经使用了 (D^*,F^*), 我们模拟了 $D^*=g^x=g^{H(\gamma\|usk)}$ 或 $F^*=g^y=g^{H(\tau\|ssk)}$, 即模拟器至少知道 x 和 y 中的一个值 (取决于被攻陷的实体). 因而能够计算出 $\mathsf{CDH}(u,F^*)$ 或 $\mathsf{CDH}(v,D^*)$. 通过消去公共因子, 可以构造出关于 $\mathsf{CDH}(u,v)$ 的方程. 最终, 我们可以解得

$$\mathsf{CDH}(u,v)=\left((T_0/T)^{H(sek_1)}\cdot(T_1/T)^{-H(sek_0)}\right)^{\frac{1}{H(sek_1)H(sek_0)(H(sek_0)-H(sek_1))}}.$$

其中 $T=\frac{\mathsf{CDH}(D^*,F^*)}{\mathsf{CDH}(u,F^*)\cdot\mathsf{CDH}(v,D^*)}$ 为已知公共项. 因此, 模拟器能够从敌手的两次哈希询问中恢复出 CDH 实例的解, 从而将事件 AskH_4 的概率归约到 CDH 问题的困难性.

现在我们考虑在事件 Coll 不发生的条件下事件 AskH4 发生的概率. 也就是说, 对每一对 (D^*,F^*), 至多存在一个共享会话密钥 sek, 使得在列表 $L_{\mathcal{H}}$ 中存在记录 $\{\mathsf{CDH}\left(F^*/v^{H(sek_0)},D^*/u^{H(sek_1)}\right),H(sek),\mathsf{ABEnc}_{(mpk,\Gamma_A)}(D^*),\Gamma_A,\mathsf{ABEnc}_{(mpk,\Gamma_B)}(F^*),\Gamma_B\}$. 由于在计算明文消息的过程中不再使用共享会话密钥 sek(由用户持有的口令生成), 我们可以到模拟结束的时候再选择口令值. 可以看出, 只有当口令值给定之后, 我们才可以判定事件 AskH4 是否发生. 为便于估计, 我们将事件 AskH4 分为以下两种不相交的情况:

(1) 参与实体属性私钥已经泄露, 而用户口令没有泄露. 由于在此情形下, 敌手无法获知正确的会话密钥, 至多存在一个 sek, 使得如下元组 $\mathsf{CDH}\ (F^*/v^{H(sek_0)},D^*/u^{H(sek_1)}),sek,\mathsf{ABEnc}_{(mpk,\Gamma_A)}(D^*),\Gamma_A,\mathsf{ABEnc}_{(mpk,\Gamma_B)}(F^*),\Gamma_B$ 出现在列表 $L_{\mathcal{H}}$ 中, 攻击者在每次会话中只能验证一个口令, 因此其成功优势受限于 $C'\cdot q_{send}^{s'}$.

(2) 参与实体属性私钥未泄露, 而用户口令已经泄露. 在这种情况下, 此时敌手虽然掌握了用户口令, 但无法获得满足挑战策略 Γ_A 或 Γ_B 的属性私钥. 若要区分 Game_3 与 Game_4, 敌手必须攻破底层 CP-ABE 方案, 以获取策略对应的解密能力. 具体而言, 敌手需要计算 $\mathsf{CDH}(D^*/u^{H(sek)},\ F^*/v^{H(sek)})$, 而这等价于以优势 $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP-ABE}}(t,2q_{\mathrm{exe}})$ 攻破 CP-ABE. 因此, 在未发生碰撞事件 Coll 的条件下, 我们有

$$\Pr\left[\mathsf{AskH}_4\mid\overline{\mathsf{Coll}}\right]\leqslant\mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP-ABE}}(t,2q_{exe})+C'\cdot q_{send}^{s'}.$$

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

综合以上两种情况, 并考虑碰撞事件 Coll 的概率, 可得

$$\begin{aligned}\mathsf{Pr}\left[\mathsf{AskH}_4\right] &\leqslant \mathsf{Pr}\left[\mathsf{AskH}_4\right] \wedge \mathsf{Coll}] + \mathsf{Pr}\left[\mathsf{AskH}_4\right] \wedge \overline{\mathsf{Coll}}] \\ &\leqslant \mathsf{Pr}[\mathsf{Coll}] + \mathsf{Pr}\left[\mathsf{AskH}_4\right] \wedge \overline{\mathsf{Coll}}] \\ &\leqslant q_H^2 \cdot \mathsf{Adv}_G^{\mathsf{CDH}}\left(t+3\tau_e\right) + \mathsf{Adv}_{\mathcal{A}}^{\mathsf{CP-ABE}}\left(t, q_{exe}\right) + C' \cdot q_{send}^{s'}.\end{aligned}$$

由以上分析可知定理结论成立.

定理 2. 设 G 为阶为素数 p 的通用双线性群, 哈希函数 H 建模为随机预言机. 对于任何概率多项式时间敌手 $\mathcal{A}$, 令 q 为 $\mathcal{A}$ 在安全游戏中访问群运算、双线性映射、哈希函数以及与挑战者交互的总次数. 则在选择明文攻击 (IND-CPA) 模型下, $\mathcal{A}$ 攻破本文 CP-ABE 方案的优势满足:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{IND-CPA}} \leqslant O\left(\frac{q^2}{p}\right).$$

其中敌手优势为 $\mathsf{Adv}_{\mathcal{P}}^{\mathsf{CP-ABE}}(\mathcal{A}) = |2\Pr[b'=b]-1|$,$b$ 为挑战者加密时随机选取的比特, b' 为 $\mathcal{A}$ 输出的猜测.

证明. 对于所采用的 CP-ABE 机制, 采用通用双线性群模型与随机预言机模型, 通过一系列混合游戏将敌手的优势归约到代数表达式碰撞的概率. 具体而言, 首先, 将原始 IND-CPA 安全游戏中的挑战密文转换为一个等价的修改游戏, 从而将敌手区分两条挑战消息的能力转化为区分挑战密文中目标双线性项的能力; 随后, 在通用双线性群模型下, 将敌手在群运算、双线性映射、哈希预言机以及安全游戏中的所有查询统一表示为关于系统随机变量的形式代数表达式, 并分析不同表达式在随机取值下发生 “意外碰撞” 的概率. 利用 Schwartz–Zippel 引理[49] 可证明, 该碰撞事件发生的概率至多为 $O\left(\frac{q^2}{p}\right)$. 在排除上述意外碰撞后, 可进一步证明: 敌手无法利用公共参数、私钥查询结果以及挑战密文构造出与解密所需核心项等价的目标表达式, 因此无法以非忽略优势区分挑战密文中所加密的两条等长消息. 故该 CP-ABE 机制在随机预言机模型和通用双线性群模型下满足 IND-CPA 安全性, 其攻击优势至多为 $O\left(\frac{q^2}{p}\right)$. 此外, 由于该安全游戏允许敌手查询多个不满足挑战访问结构的属性私钥, 上述安全性定义同时隐含刻画了方案的抗串谋能力, 本文不再赘述, 完整证明可参考文献 [46] .

6 启发式安全性分析

6.1 抗离线口令猜测攻击

在本文方案中, 用户仅通过口令 PW 与外部工具服务器协商会话密钥, 服务器侧仅保存与口令绑定的口令验证记录, 而不保存任何可直接用于本地验证口令猜测的哈希值或明文口令. 假设攻击者窃听了完整的登录流程, 获得了用户端发送的盲化值 $\alpha' = H_0(PW)^{r'}$ 和临时公钥 $X = g^x$, 以及服务器返回的 OPRF 响应 $\beta' = \alpha'^{k_s} = H_0(PW)^{r'k_s}$. 为了验证一个口令猜测 PW', 攻击者需要计算 $\mathsf{RW} = H\left(PW \| \beta^{1/r}\right) = H\left(PW \| H_0(PW)^{k_s}\right)$, 以便与协议中的值 $Cre = \mathsf{AuthEnc}_{\mathsf{RW}}(a, A, B)$ 进行比对. 然而, 从截获的信息中, 攻击者只能得到: 与特定随机盲化因子 r' 绑定的 $\beta' = \alpha'^{k_s} = H_0(PW)^{r'k_s}$, 但不知道服务器的 OPRF 密钥 k, 由于不知道 k_s, 攻击者无法从 β' 中提取出 $H_0(PW)^{k_s}$(涉及离散对数问题), 也无法为猜测的 PW' 计算 $H_0(PW')^{k_s}$. 因此, 攻击者无法利用截获的 β' 来离线测试 PW' 是否正确. 攻击者若想验证 PW', 必须模拟一个登录请求, 即利用 PW' 生成一个新的盲化值 $\alpha^* = H_0\left(PW'\right)^{r^*}$, 并将其发送给拥有正确密钥 k_s 的服务器, 以获得对应的响应 β^*, 进行后续计算. 这本质上是一次在线交互. 而由于 honeylist 记录了认证失败的次数, 攻击者在线尝试次数有限, 获得口令的概率极小. 换言之, 在仅窃听通信的攻击模型下, 攻击者无法执行离线口令猜测的攻击.

6.2 抗重放攻击

重放攻击是指攻击者通过截获并重复发送历史会话消息, 试图冒充合法实体或获取新的会话机密信息. 在本文提出的协议中, 用户与外部工具服务器在每轮会话中均需分别选取新的随机数 $\gamma \in \mathbb{Z}_p^*$ 和 $\tau \in \mathbb{Z}_p^*$, 并据此

重新计算生成新鲜参数 D 和 F. 由于这些关键参数均与当前会话的临时随机数绑定, 且在不同会话中彼此独立. 因此攻击者即使重放先前截获的消息, 协议验证过程依赖于当前会话中新生成的随机数及其派生参数, 也无法通过一致性验证或生成合法的新会话数据. 由此, 本文方案能够有效抵抗重放攻击.

6.3 抗内部攻击

在本文给定的威胁模型下, 所提方案能够有效抵御服务器侧和用户侧的内部攻击. 一方面, 用户与服务器之间采用基于口令的认证密钥协商协议, 服务器仅保存与口令绑定的口令验证记录和经 OPRF 计算后的加密凭据 Cre, 并不掌握任何可直接用于本地验证口令猜测的哈希值或明文口令. 即使内部攻击者窃取了用户数据库或会话日志, 也只能获得在计算上难以与随机值区分的验证记录和密文, 无法在本地对候选口令进行离线口令猜测, 其攻击能力被限制为需与真实用户在线交互的猜测尝试. 而由于 honeylist 记录了认证失败的次数, 攻击者在线尝试次数有限. 另一方面, 用户输入数据及推理结果在全流程中均通过会话密钥和属性基加密保护, 大语言模型仅接触密文和访问策略描述而不持有任何解密密钥, 服务器端只有在同时满足属性策略并完成口令认证的前提下才可恢复明文进行计算. 因此, 即便内部人员获取了大语言模型侧的日志数据或服务器侧的存储记录, 也无法越权恢复用户隐私数据或扩大访问权限, 从而实现对内部攻击 (日志窃取等) 的有效防护. 需要指出的是, 即使大语言模型因外部干扰或自身规划偏差将密文 CT 或中间结果密文 C_i 错误转发至非目标工具服务器, 由于所有密文均通过访问策略 Γ_A 与会话密钥 sek 双重保护, 未授权工具既无法满足密文嵌入的属性策略 (缺少对应属性私钥) , 也无法获知正确的会话密钥 sek, 因此仍无法解密密文或恢复任何明文信息. 这也表明本方案对大语言模型异常调度行为具备内在的鲁棒性, 数据机密性不依赖于大语言模型调度的正确性.

6.4 数据细粒度访问控制

本文方案通过引入密文策略属性基加密机制实现细粒度访问控制. 用户在为私有数据加密时, 可以自定义一个基于属性逻辑表达式的访问策略 (如任务: 情感分析 ∩ 安全级别: 高), 并将该策略直接嵌入到密文中, 使数据访问能力由 “是否满足策略” 而非 “是否获得密文” 决定. 只有属性集合 $Attr(P)$ 满足该策略的外部工具 (即持有相应属性私钥的工具端) 才能成功解密密文, 获得明文数据. 这一设计使得访问控制的粒度从传统的 “用户、角色” 级别细化到 “属性、策略” 级别, 用户可以根据数据敏感性、任务上下文以及工具的实时属性灵活定义授权范围, 且策略的验证由密码学算法强制执行, 无需依赖第三方授权服务器. 因此, 本文方案能够在不牺牲系统灵活性的前提下, 实现真正意义上的细粒度、可定制的访问控制.

6.5 抗串谋攻击

本文方案能够有效抵抗串谋攻击, 其安全性主要依赖于底层密文策略属性基加密 (CP-ABE) 固有的抗串谋特性. 在 CP-ABE 机制中, 密钥生成中心为每个实体颁发属性私钥时, 会为每个私钥引入独立的随机化因子, 使得不同实体的私钥在代数结构上互不兼容. 因此, 即使多个恶意实体联合, 将其各自持有的属性私钥进行合并, 也无法构造出一个能够满足更高权限访问策略的有效解密密钥. 此外, 本方案中用户与外部工具服务器之间每次会话独立协商的高熵会话密钥 sek 仅对本次会话有效, 进一步限制了串谋攻击的收益. 即使多个被攻陷的工具或用户共享其属性私钥, 也无法解密那些需要额外属性或不同会话密钥保护的密文.

6.6 前向安全性

在本文方案中, 前向安全性主要指即便攻击者在某一时刻成功获取用户口令或服务器长期状态信息, 仍无法回溯解密之前已经完成的会话中所传输的用户输入数据 msg_A 及推理结果 msg_B. 具体来说, 过去的数据一旦在相应会话结束后被正确遗忘其密钥, 即便之后发生密钥泄露, 攻击者也只能影响后续会话, 而不能破坏既有会话的机密性. 本方案提供了前向安全性保障, 其核心在于将数据机密性的基于临时的、前向安全的会话密

钥, 而非任何长期身份凭证, 不同会话之间的会话密钥在密码学上相互独立. 具体而言, 在每次会话的初始阶段, 用户与外部服务器通过前向安全的口令认证密钥交换协议协商出一个高熵的临时会话密钥 sek, 该密钥的安全性本质上依赖于每次会话随机生成的临时密钥 x, y. 这意味着, 即使未来用户的长期口令 PW 或工具端的属性私钥 ssk 发生泄露, 敌手也无法据此回溯推导出过往任何一次会话中所使用的 sek. 由于方案中所有用户私有数据 msg_A 及其推理结果 msg_B 均使用当次会话的 sek 作为主密钥进行加密, 任何历史会话中产生的密文 (包括任务输入密文 CT 与响应密文 CF) 的机密性因此得到永久性保护, 长期密钥的泄露无法解密过去已安全完成的交互历史, 从而实现了 “一次一密” 式的长期数据机密性.

7 性能分析

本节对本文提出的大语言模型智能体数据安全传输方案进行性能分析. 协议由两类核心密码组件构成: 用户 U 与外部工具服务器 S 之间的口令认证密钥交换; 基于密文策略属性基加密的细粒度访问控制, 用于将解密能力与属性集合绑定. 本节从安全性、计算开销和通信开销三个方面, 对本文协议进行全面的性能分析.

7.1 安全性分析

为全面评估本文方案在大语言模型智能体场景中的安全性与适用性, 本文从功能实现与安全属性两个维度出发, 将本文方案与近年来基于密码学的 LLM 推理隐私保护方案进行了系统比较, 包含以下主流技术: 差分隐私 (Differential privacy, DP)[25~27]、同态加密 (Homomorphic encryption, HE)[21,24,35]、多方安全计算 (Multi-party computation, MPC)[28~30]、函数秘密共享 (Function secret sharing, FSS)[50~52] 以及属性基加密方案 (Attribute-based encryption, ABE)[53], 具体比较结果如表 3 所示.

本文构建了一组包含 12 项指标的评价体系, 该指标体系既能反映本文方案的设计目标, 也能够反映 DP、HE、MPC、FSS 等现有方法在大语言模型智能体场景下的适用性差异. 需要说明的是, 这 12 项评价指标并非任意选取, 而是结合本文所考虑的 “用户, 大语言模型, 外部工具服务器” 三方交互架构. 1) 这些指标覆盖了传统密码协议对身份认证与密钥协商、数据机密性与访问控制等方面的核心安全需求; 2) 这些指标也反映了大语言模型智能体场景下由任务解析、工具调用、结果聚合和不可信调度所带来的特有安全需求[54]. 此外, 在指标设计过程中, 本文遵循了 ISO/IEC 15408 [55] 等标准所倡导的安全评估指标应具备相关性、完备性与非冗余性的原则, 各指标均直接对应本文场景中的实际安全需求, 能够覆盖从用户请求提交到接收响应的完整链路, 且不同指标分别刻画不同层面的安全属性与系统能力. 因此, 采用这 12 项指标能够在保证公平性的同时, 更准确地反映相关方案在本文应用场景中的安全性能与适用边界. 具体而言, 各评价指标定义如下:

C1. 双向身份认证 (Mutual authentication): 在用户向智能体请求服务之前, 协议应当确保用户与目标工具服务器能够彼此验证对方身份的真实性与合法性, 避免未授权实体参与后续交互或访问数据. 本文方案采用前向安全性的口令基认证协议使得通信双方仅凭共享的低熵口令即可完成双向身份认证, 防止身份仿冒攻击.

C2. 输入数据机密性 (Input data confidentiality): 用户私有输入数据 msg_A 在传输、存储以及工具处理的整个过程中应始终对未授权实体不可见, 其中包括承担协调功能的 LLM 智能体. 本文方案使用会话密钥 sek 与用户指定的访问策略 Γ_A 对 msg_A 进行认证加密, 明文恢复严格限定于属性满足策略的授权工具端.

C3. 输出数据机密性 (Output data confidentiality): 智能体推理结果 msg_B 在回传给用户的过程中对中间节点及未授权参与方不可获取. 本文方案对模型输出采用会话密钥 sek 与工具服务器端指定的访问策略 Γ_B 进行加密从而保证推理结果仅对合法用户可见, 其他实体即使截获密文数据 CF 也无法恢复其明文内容.

C4. 输出结果正确性 (Output correctness): 在引入安全保护机制之后, 用户最终响应须与无保护场景下的直接执行结果一致, 不因额外的保护引入噪声、偏差或计算误差. 本文方案的加解密过程均属于确定性的精确变换, 不会改变底层数据的语义信息和数值内容, 因此能够保持输出结果的正确性.

https://www.sciengine.com/doi/10.1360/SSI-2026-0175

C5. 大语言模型侧明文隔离 (LLM-side plaintext isolation): 大语言模型不应接触、恢复或解析用户数据的明文内容和推理结果, 只能处理密文及必要的相关元数据 (task_i, Q, R). 本文将 L 限定为仅转发加密后的用户输入与工具输出, 从而在密码学层面达成可用但不可见的隔离效果, 降低半可信实体窥探用户隐私的风险.[3)]

C6. 功能保持 (Functionality preservation): 安全机制的引入不应破坏智能体系统原有的核心功能, 具体而言, LLM 仍需能够完成查询解析、任务规划、工具调度以及多源响应的聚合操作. 本文方案中所采用的加密与解密操作对 LLM 保持透明, 其既有工作流逻辑无须变动, 因此智能体的各项基础能力仍可得以完整保持.

C7. 最小信任 (Trust minimization): 协议设计应尽可能减少对系统中任一单一实体的信任依赖, 例如 LLM 服务提供商或外部工具服务器, 局部实体的沦陷不得引发系统性安全失效. 本文方案通过端到端加密与访问控制策略的联合约束, 用户仅需维护个人口令与属性私钥的可信性, 其余参与者均置于最小信任假设之下.

C8. 细粒度访问控制 (Fine-grained access control): 用户应能够结合数据敏感程度、任务上下文以及外部工具的属性信息, 灵活规定哪些实体可以解密和使用其数据. 本文方案利用 CP-ABE 机制, 将访问策略 (涵盖与、或、门限等逻辑组合的属性结构) 内嵌于密文, 由此达成面向多接收方、基于布尔逻辑表达式的精准授权.

C9. 抗串谋性 (Collusion resistance): 即使多个恶意实体 (如多个被攻陷的工具模型或用户) 联合, 也无法通过合并各自的属性私钥或局部信息获得超出各自权限的数据访问能力. 本文的抗串谋性由 CP-ABE 私钥中与主体绑定的随机化因子和仅对目标会话双方可见的会话密钥 sek 共同保证, 即使多个恶意工具之间进行联合, 也无法通过合并属性私钥和局部信息获得超出自身权限的数据访问能力.

C10. 密钥协商 (Key agreement): 用户与外部工具服务器之间应能安全协商出一次性的高熵会话密钥, 该密钥用于后续数据加密与完整性保护. 本方案采用前向安全的口令认证密钥协商协议 (如 OPAQUE [11]), 在不依赖公钥基础设施的前提下完成双向身份认证与密钥协商, 并确保会话密钥具备语义安全性与前向安全性.

C11. 端到端安全共享 (End-to-end secure sharing): 从用户加密数据并发起请求开始, 到授权工具完成解密和计算, 再到结果加密返回并由用户恢复, 整个处理过程都应具备端到端的机密性、完整性和访问控制保障. 本文方案将会话密钥机制与基于属性的访问控制策略结合起来, 使数据在传输、处理和返回过程中始终以受保护的形式存在, 从而保证其在整个流转过程中的安全共享.

C12. 智能体-工具兼容性 (Agent-tool compatibility): 方案应能够适配大模型智能体调用外部工具的基本流程, 而不影响其原有的任务解析、任务拆分、工具选择和结果整合功能. 本文不改变 LLM 智能体现有的工作方式, 只在数据输入输出环节加入口令认证和密文保护机制. 外部工具服务器在满足访问策略的条件下, 可对密文进行处理, 并将结果以密文形式返回给 LLM, 因此该方案能够较好地兼容现有智能体的工具调用流程.

在多用户并发场景下, 本文通过 “用户级独立密钥、会话级状态隔离与任务级结果绑定” 的分层机制实现数据隔离: 每个用户独立协商会话密钥并生成唯一会话标识, 任务进一步绑定唯一任务标识, L 仅基于会话标识进行任务分发与聚合而不接触明文内容. 即使发生任务错发或调度异常, 由于缺乏对应会话密钥, 错误接收方也无法解密或验证数据, 从而有效防止跨用户信息泄露, 避免并发环境中的任务混淆和结果错配.

7.2 效率分析

为进一步评估本文方案的实用性, 本节从计算开销和通信开销两个方面, 将本文方案与相关代表性工作 [53, 56∼65] 进行对比, 结果如表 4 所示. 实验在一台个人计算机上完成, 其配置为 Intel(R) Core(TM) i7-13650HX(2.60 GHz) 和 24 GB RAM. 我们对双线性对运算、模指数运算以及哈希运算等基本密码学操作的计算开销进行了全面评估. 双线性对密码库为椭圆曲线参数生成和双线性对运算提供了有效支持, 基于该库我们实现了协议中的相关算法, 以完成多种密码学运算. 其中, T_M、T_H、T_P 和 T_E 分别表示一次标量乘法、哈希运算、双线性对运算以及模指数运算, T_{Enc} 表示一次对称加密操作的执行时间, T_{ABE} 表示执行一

3) 本方案确保用户私有输入数据 msg_A 与工具输出结果 msg_B 在传输与调度过程中保持密文状态, 防止大语言模型 L 获取数据内容. 对于明文查询 prom 及任务目标 goal 中包含的任务意图或上下文语义, 属于智能体正常任务调度功能的必要组成部分, 方案不试图隐藏用户意图层面的语义信息.

表 3 面向大语言模型的相关密码学方案对比.

Table 3 Comparison of relevant cryptography-based schemes for large language model.

Scheme	Year	Ref.	Cryptography-based approaches	Security goal	The proposed twelve evaluation criteria†											
					C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Xue et al.	2026	[52]	Function secret sharing	Function-hiding computation	–	✓	✓	✓	✓	✓	×	×	×	×	✓	✓
You et al.	2026	[51]	Function secret sharing	Function-hiding computation	–	✓	✓	✓	✓	✓	✓	×	×	×	✓	×
Deng et al.	2025	[26]	Differential privacy	Preventing output leakage	–	✓	×	✓	✓	✓	✓	–	×	–	×	×
Zhang et al.	2025	[21]	Homomorphic encryption	Computation over ciphertexts	×	✓	✓	✓	✓	✓	✓	×	×	–	✓	×
Zhang et al.	2025	[24]	Homomorphic encryption	Computation over ciphertexts	×	✓	✓	✓	✓	✓	✓	×	×	–	✓	×
Tan G.	2025	[28]	Multi-party computation	Confidential joint computation	–	✓	×	✓	✓	✓	✓	×	✓	×	–	×
Xiao et al.	2025	[53]	Attribute-based encryption	Fine-grained access control	×	✓	✓	✓	×	✓	×	✓	✓	×	✓	×
Garg-Torra	2024	[27]	Differential privacy	Preventing output leakage	–	✓	×	✓	×	✓	×	–	–	–	×	×
Zhang et al.	2024	[35]	Homomorphic encryption	Computation over ciphertexts	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓
Gupta et al.	2024	[50]	Function secret sharing	Function-hiding computation	–	✓	✓	✓	✓	✓	×	×	×	–	✓	×
Pang et al.	2024	[30]	Multi-party computation	Confidential joint computation	✓	✓	×	✓	✓	✓	✓	×	×	×	✓	✓
Du et al.	2023	[25]	Differential privacy	Preventing output leakage	–	✓	×	✓	✓	✓	✓	–	×	–	×	×
Akimoto et al.	2023	[29]	Multi-party computation	Confidential joint computation	–	✓	✓	✓	✓	✓	✓	×	✓	×	✓	✓
Our scheme	2026	–	Attribute-based encryption	Fine-grained access control	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

C1: Mutual authentication.
C2: Input data confidentiality.
C3: Output data confidentiality.
C4: Output correctness.
C5: LLM-side plaintext isolation.
C6: Functionality preservation.
C7: Trust minimization.
C8: Fine-grained access control.
C9: Collusion resistance.
C10: Key agreement.
C11: End-to-end secure sharing.
C12: Agent-tool compatibility.

✓: Achieving the goal.
×: Not achieving the goal.
–: Not considering the goal.

次标准密文策略属性基加密 [46] 的完整耗时. 在本文方案中, 用户侧和工具服务器侧的主要计算开销分别为 $T_U = 9T_E + 4T_M + 11T_H + T_{Enc} + T_{ABE}, T_S = 7T_E + 4T_M + 5T_H + T_{Enc} + T_{ABE}$, 因此, 本文方案的总计算开销约为 $T_U + T_S \approx 239.236$ ms. 从表 4 可以看出, 相比同类协议, 本文方案在总计算时间上并非最低, 这主要源于两方面原因: 本文在用户侧与工具服务器侧均引入了 CP-ABE 机制, 以实现访问策略绑定下的数据保护与细粒度授权控制; 同时, 本文不仅完成身份认证, 还需要保证敏感数据输入与输出结果在智能体调用链路中的机密性和授权一致性. 换言之, 本文方案所增加的计算代价对应的是认证、密钥协商、访问控制和端到端密文共享等综合安全能力的获得. 特别地, 从表 3 的安全属性对比结果可以发现, 本文方案是在满足全部 12 项评价指标的前提下实现上述计算性能, 因此虽然计算开销有所增加, 但显著提升了系统的整体安全性与场景适用性.

本文对方案的通信开销进行了评估, 并对各类参数的比特长度作了统一设定. 具体而言, 我们取参数长度如下 $|G| = |G_T| = 1024$bits,$|Z_p^*| = 160$bits,$|\kappa| = 160$bits, 认证访问策略 $|\Gamma| = 64$bits. 在 PBC 库中, Type A 曲线表示为 $E(F_q) : y^2 = x^3 + x$, 其中阶为 p 的群 G 和 G_T 都是椭圆曲线 $E(F_q)$ 的子群. 参数 p 和 q 的长度分别为 160 bits 和 512 bits. 为便于统一比较, 本文进一步设定单个身份标识和口令长度均为 64 bits, 一次分组加密或哈希运算的输出长度为 256 bits. 本文方案的用户侧、工具服务器侧及总通信量分别为 3,392 bits、4,352 bits 和 7,744 bits, 通信开销主要来自用户侧提交加密输入、大语言模型对密文任务的转发, 以及工具服务器返回加密结果三个部分, 这一设计与智能体场景下 “由大语言模型负责编排、由外部工具执行实际计算” 的系统模型保持一致, 与表 4 中列出的已有方案相比, 本文方案的总通信开销处于同一数量级, 因此本文所提方案的通信开销是较为合理且可以接受的. 此外, 表 4 中记 l 为用户提交的属性数量, 相关方案的部分计算项随 l 线性增长. 本文方案同样包含基于访问策略的属性处理过程, 因此其开销也会随着属性数量增加而上升. 但由于本文应用场景下的 CP-ABE [46] 为协议组件用于输入数据与输出结果的封装与解封装, 而非在整条计算链路上反复调用, 因此该开销在实际部署中是可接受的. 特别是在工具调用型大语言模型智能体场景中, 敏感数据通常仅在请求提交和结果返回两个关键阶段需要进行策略保护, 进一步减少了属性加密引入的性能压力.

在实际部署中, 外部工具服务器的调用可能因网络超时、服务不可用或属性验证失败等原因无法正常完成.

表 4 相关方案性能比较.

Table 4 Performance comparison among relevant schemes.

Scheme	Ref.	Computation cost			Communication cost		
		User	Server	Total(ms)	User(bits)	Server(bits)	Total(bits)
Xiao et al.	[53]	$2T_P + (2l+2)T_E + (2l+2)T_M$	$(2l+1)T_P + (2l+5)T_E + (3l+4)T_M$	≈ 235.089	4,032	4,224	8,256
Li et al.	[56]	$12T_E + 9T_H$	$5T_M + 14T_E + 16T_H + 3T_P$	≈ 124.539	3,528	4,216	7,744
Abbasinezhad et al.	[57]	$12T_M + 10T_H + T_{Enc}$	$9T_M + 14T_H + 3T_{Enc}$	≈ 117.431m	2,928	4,632	7,560
Vijayakumar et al.	[58]	$6T_P + 13T_H$	$13T_M + 9T_P$	≈ 141.467	3,344	3,216	6,560
Pussewalage et al.	[59]	$9T_P + 12T_M + 11T_H$	$lT_M + (2l+2)T_H$	≈ 139.855	3,092	4,728	7,820
Sucasas et al.	[60]	$(3l+8)T_E + l \cdot T_P$	$(5l+6)T_E + 2l \cdot T_M$	≈ 176.478	3,018	4,408	7,426
Luo et al.	[61]	$2l \cdot T_{Enc} + 2l \cdot T_H + 2l \cdot T_M$	$5l \cdot T_M + 3l \cdot T_P$	≈ 234.685	3,152	4,458	7,610
Li et al.	[62]	$(2l+6)T_E + (2l+7)T_P$	$(l+3)T_E + (4l+2)T_P + (l+5)T_M$	≈ 284.694	3,890	6,464	10,354
Song et al.	[63]	$5T_P + (2l+9)T_E + (2l+4)T_M + 5T_H$	$5T_P + (2l+9)T_E + (2l+4)T_M + 5T_H$	≈ 246.900	4,160	4,160	8,320
Li et al.	[64]	$(l+1)T_P + (5l+4)T_E + (l+1)T_{Enc}$	$(2l+3)T_P + (4l+5)T_E + (3l+5)T_M$	≈ 294.360	4,160	4,416	8,576
Zheng et al.	[65]	$10T_P + (3l+3)T_E$	$(2l+2)T_P + 5T_E + (2l+3)T_M$	≈ 207.283	4,096	5,120	9,216
Our scheme	–	$9T_E + 4T_M + 11T_H + T_{Enc} + T_{ABE}$	$7T_E + 4T_M + 5T_H + T_{Enc} + T_{ABE}$	≈ 239.236	3,392	4,352	7,744

*One hash operation $T_H \approx 1.436$ ms, one block encryption $T_{Enc} \approx 1.826$ ms, one bilinear pairing operation $T_P \approx 5.064$ ms, one modular exponential operation $T_E \approx 2.132$ ms, one elliptic curve scalar multiplication $T_M \approx 3.603$ ms, one ciphertext-policy attribute-based encryption $T_{ABE} \approx 74.836$ ms. "l" represents the number of attributes submitted by the user, the computation cost increases linearly with l. In our scheme, assuming $l = 5$, the computation time of our scheme is 239.236 ms and the corresponding communication cost is 7,744 bits.

针对上述场景, 本文方案将异常处理视为安全协议的扩展功能, 而非独立模块. 若会话密钥协商超时或属性验证未通过, 工具服务器将向大语言模型返回明确的错误标识 (如 $\perp$), 大语言模型应将错误信息转发给用户, 由用户决定是否重试或调整请求参数. 对于因服务不可用导致的失败, 本文方案作为密码学协议层不规定具体的重试次数或超时阈值, 可由具体部署环境根据容错需求灵活配置 (如设置最大重试次数或切换备用工具).

8 总结与展望

本文针对大语言模型智能体在调用外部工具时面临的身份认证缺失、敏感数据易泄露以及访问控制粒度不足等问题, 提出了一种基于口令的数据安全传输方案. 方案以用户低熵口令为信任根, 通过前向安全的口令认证密钥协商协议实现用户与工具服务器端的双向身份认证, 利用密文策略属性基加密机制对用户输入数据和工具输出结果实施细粒度保护, 使敏感数据的解密与处理仅在授权端完成, 从而实现数据的可用不可见. 理论分析表明, 本文方案能够满足双向身份认证、输入/输出数据机密性、细粒度访问控制等 12 项安全目标. 实验分析表明, 在满足全部评价指标的前提下, 本文方案保持了较好的计算开销与通信开销.

冯登国院士等人在综述 [16] 中通过对人工智能安全与隐私攻防层面的系统梳理, 强调当前尚不存在 "一劳永逸" 的通用防御方案, 实际部署中必须在安全性、性能开销、系统复杂度与可用性之间进行精细权衡, 并根据应用场景构建分层、多点联动的纵深防御体系. 未来将进一步围绕复杂开放环境中的工具动态接入、跨平台协同调用以及更高效的轻量化实现展开研究, 以进一步提升方案在真实智能体系统中的适用性与鲁棒性.

参考文献

1 Romera-Paredes B, Barekatain M, Novikov A, et al. Mathematical discoveries from program search with large language models. Nature, 2024, 625: 468–475

2 Hutson M. Hackers easily fool artificial intelligences-adversarial attacks highlight lack of security in machine learning algorithms. Science, 2018, 361: 215–215

3 Achiam J, Adler S, Agarwal S, et al. GPT-4 technical report, Apr. 2023. https://arxiv.org/pdf/2303.08774

4 Touvron H, Lavril T, Izacard G, et al LLaMA: Open and efficient foundation language models, Feb. 2023. https://arxiv.org/pdf/2302.13971

5 DeepSeek-AI. Deepseek-v3 technical report, Feb. 2025. https://github.com/deepseek-ai/DeepSeek-V3

6 Lin C, Han Z, Zhang C, et al. Parrot: Efficient serving of llm-based applications with semantic variable. In: Proceedings of USENIX Symposium on Operating Systems Design and Implementation, 2024. 929–945

7 Microsoft. AutoGen: Microsoft agent framework migration guide, Apr. 2026. https://github.com/microsoft/autogen

8 Shen Y, Song K, Tan X, et al. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 38154–38180

9 Hong S, Zhuge M, Chen J, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In: Proceedings of International Conference on Learning Representations, 2024. 23247–23275

10 Weng L. LLM Powered Autonomous Agents, Jun. 2023. https://lilianweng.github.io/posts/2023-06-23-agent/

11 Jarecki S, Krawczyk H, Xu J. Opaque: an asymmetric pake protocol secure against pre-computation attacks. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2018. 456–486

12 Bellare M, Pointcheval D, Rogaway P. Authenticated key exchange secure against dictionary attacks. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, 2000. 139–155

13 Dutta T S. GitHub MCP server vulnerability let attackers access private repositories, May. 2025. https://reurl.cc/Xa8QNa

14 Wang Y, Pan Y, Guo S, et al. Security of internet of agents: Attacks and countermeasures. IEEE Trans Pattern Anal Mach Intell, 2025, 6: 1611–1624

15 Hardt D. RFC 6749: The OAuth 2.0 authorization framework, Oct. 2012. https://www.rfc-editor.org/info/rfc6749

16 He X, Xu G, Han X, et al. Artificial intelligence security and privacy: a survey. Sci China Inf Sci, 2025, 68: 181101

17 Deng Z, Guo Y, Han C, et al. Ai agents under threat: A survey of key security challenges and future pathways. ACM Comput Surv, 2025, 57(07): 1–36

18 Wei T, Wang D. Unveiling privacy issues in large language models: insights from real-world application scenarios. Sci Sin Inform, 2026, 56: 1390–1406 [魏同心, 汪定. 揭示大语言模型中的隐私问题: 来自真实应用场景的见解. 中国科学: 信息科学, 2026, 56(06): 1390–1406]

19 Pedro R, Coimbra M E, Castro D, et al. Prompt-to-sql injections in llm-integrated web applications: Risks and defenses. In: Proceedings of 47th International Conference on Software Engineering, 2025. 1768–1780

20 Bagdasarian E, Yi R, Ghalebikesabi S, et al. Airgapagent: Protecting privacy-conscious conversational agents. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2024. 3868–3882

21 Zhang J, Yang X, He L, et al. Secure transformer inference made non-interactive. In: Proceedings of Annual Network and Distributed System Security Symposium, 2025. 1–17

22 Bredehoft R, Frery J. Towards encrypted large language models with FHE, Aug. 2023. https://huggingface.co/blog/encrypted-llm

23 Hao M, Li H, Chen H, et al. Iron: Private inference on transformers. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 15718–15731

24 Zhang R, Zheng Z, Bao W. Practical secure inference algorithm for a fine-tuned large language model based on fully homomorphic encryption. IEEE Trans Inf Forensics Secur, 2025, 21: 17–29

25 Du M, Yue X, Chow S S, et al. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2023. 2665–2679

26 Deng Z, Sun R, Xue M, et al. Hardening llm fine-tuning: From differentially private data selection to trustworthy model quantization. IEEE Trans Inf Forensics Secur, 2025, 20: 7211–7226

27 Garg S, Torra V. Task-specific knowledge distillation with differential privacy in llms. In: Proceedings of European Symposium on Research in Computer Security, 2024. 374–389

28 Tan G. Scalable cryptography for trustworthy machine learning in the llm era. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2025. 4866–4868

29 Akimoto Y, Fukuchi K, Akimoto Y, et al. Privformer: Privacy-preserving transformer with mpc. In: Proceedings of IEEE European Symposium on Security and Privacy, 2023. 392–410

30 Pang Q, Zhu J, Möllering H, et al. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In: Proceedings of IEEE Symposium on Security and Privacy, 2024. 4753–4771

31 Kim S, Yun S, Lee H, et al. Propile: Probing privacy leakage in large language models. In: Proceedings of Advances in Neural Information Processing Systems, 2023. 20750–20762

32 Wang P, Dong B, Cai Y, et al. Game of arrows: On the (in-)security of weight obfuscation for on-device TEE-shielded llm partition algorithms. In: Proceedings of the USENIX Security Symposium, 2025. 279–298

33 Zhang K, Wang J, Hua E, et al. Cogenesis: A framework collaborating large and small language models for secure context-aware instruction following. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, 2024. 4295–4312

34 Zhang M, He T, Wang T, et al. Latticegen: Hiding generated text in a lattice for privacy-aware large language model generation on cloud. In: Proceedings of Findings of the Association for Computational Linguistics, 2024. 2674–2690

35 Zhang X, Xu H, Ba Z, et al. Privacyasst: Safeguarding user privacy in tool-using large language model agents. IEEE Trans Depend Secur Comput, 2024, 21: 5242–5258

36 Pan X, Zhang M, Ji S, et al. Privacy risks of general-purpose language models. In: Proceedings of IEEE Symposium on Security and Privacy, 2020. 1314-1331

37 Song C, Raghunathan A. Information leakage in embedding models. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2020. 377–390

38 Hu C T, Ferraiolo D F, Kuhn D R, et al. Guide to attribute based access control (ABAC) definition and considerations, Jan. 2014.

https://doi.org/10.6028/NIST.SP.800-162
39 Huang K, Narajala V S, Yeoh J, et al. A novel zero-trust identity framework for agentic ai: decentralized authentication and fine-grained access control, May. 2025. https://arxiv.org/pdf/2505.19301
40 Chan C M, Yu J, Chen W, et al. Agentmonitor: A plug-and-play framework for predictive and secure multi-agent systems, Aug. 2024. https://arxiv.org/pdf/2408.14972
41 He Y, Li B, Liu L, et al. Towards label-only membership inference attack against pre-trained large language models. In: Proceedings of the USENIX Security Symposium, 2025. 1609–1628
42 Maddison L. Samsung workers made a major error by using ChatGPT, Apr. 2023. https://reurl.cc/8e9jRd
43 Lakshmanan R. Zero-click AI vulnerability exposes Microsoft 365 Copilot data without user interaction, Jun. 2025. https://thehackernews.com/2025/06/zero-click-ai-vulnerability-exposes.html
44 NIST. Announcing the AI agent standards initiative for interoperable and secure agents, Feb. 2026. https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure?utm_source=chatgpt.com
45 Feng D, Liu J, Qin Y, et al. Trusted computing theory and technology in innovation-driven development. Sci Sin Inform, 2020, 50: 1127–1147 [冯登国, 刘敬彬, 秦宇, 等. 创新发展中的可信计算理论与技术. 中国科学: 信息科学, 2020, 50(08):1127–1147]
46 Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proceedings of IEEE Symposium on Security and Privacy, 2007. 321-334
47 Lai J, Huang X, He D. Efficient hierarchical identity-based encryption based on sm9. Sci Sin Inform, 2023, 53: 918–930 [赖建昌, 黄欣沂, 何德彪, 等. 基于商用密码 SM9 的高效分层标识加密. 中国科学: 信息科学, 2023, 53(05):918–930]
48 Wang D, Cheng H, Wang P, et al. Zipf's law in passwords. IEEE Trans Inf Forensics Secur, 2017, 12: 2776–2791
49 Schwartz J T. Fast probabilistic algorithms for verification of polynomial identities. J. ACM, 1980, 27: 701–717
50 Gupta K, Jawalkar N, Mukherjee A, et al. SIGMA: Secure GPT inference with function secret sharing. In: Proceedings of Privacy Enhancing Technologies Symposium, 2024. 1–19
51 You Z, Dong X, Cheng K, et al. Prifft: Privacy-preserving federated fine-tuning of large language models via hybrid secret sharing. IEEE Trans Depend Secur Comput, 2026, 23(03): 6167–6182
52 Xue Y, Liu L, Luo Y, et al. Flute: Fss-based secure two-party llm inference using partial transformer encryption. IEEE Trans Depend Secur Comput, 2026, 23: 685–702
53 Xiao P, Yang S, Wang H, et al. Privacy-preserving revocable access control for llm-driven electrical distributed systems. Peer Peer Netw Appl, 2025, 18(3): 148–160
54 Luo H, Dai S, Ni C, et al. Agentauditor: Human-level safety and security evaluation for llm agents. In: Proceedings of Advances in Neural Information Processing Systems, 2026. 43241–43298
55 ISO/IEC 15408-1. Information security, cybersecurity and privacy protection— Evaluation criteria for IT security—Part 1: Introduction and general model, Mar. 2026. https://www.iso.org/standard/88134.html/
56 Li W, Li X, Gao J, et al. Design of secure authenticated key management protocol for cloud computing environments. IEEE Trans Depend Secur Comput, 2021, 18: 1276–1290
57 Abbasinezhad-Mood D, Mazinani S M, Nikooghadam M, et al. Efficient provably-secure dynamic id-based authenticated key agreement scheme with enhanced security provision. IEEE Trans Depend Secur Comput, 2022, 19: 1227–1238
58 Vijayakumar P, Azees M, Kozlov S A, et al. An anonymous batch authentication and key exchange protocols for 6g enabled vanets. IEEE Trans Intell Transp Syst, 2022, 23: 1630–1638
59 Pussewalage H S G, Oleshchuk V A. A delegatable attribute based encryption scheme for a collaborative e-health cloud. IEEE Trans Serv Comput, 2023, 16: 787–801
60 Sucasas V, Mantas G, Papaioannou M, et al. Attribute-based pseudonymity for privacy-preserving authentication in cloud services. IEEE Trans on Cloud Comput, 2023, 11: 168–184
61 Luo F, Wang H, Lin C, et al. Abaeks: Attribute-based authenticated encryption with keyword search over outsourced encrypted data. IEEE Trans Inf Forensics Secur, 2023, 18: 4970–4983
62 Li X, Xie Y, Peng C, et al. Eprear:an efficient attribute-based proxy re-encryption scheme with fast revocation for data sharing in aiot. IEEE Trans Mob Comput, 2025, 24: 11005–11018
63 Song M, Wang D. Ab-pake: Achieving fine-grained access control and flexible authentication. IEEE Trans Inf Forensics Secur, 2024, 19: 6197–6212
64 Li J, Zhang E, Zhang Y, et al. A traceable and revocable ciphertext policy attribute-based encryption with policy authentication. IEEE Trans Depend Secur Comput, 2026, 23: 2962–2973
65 Zheng K, Liu J, Hou G, et al. Secure hierarchical multi-modal data sharing scheme in satellite internet. Sci Sin Inform, 2026, 56: 581–602 [郑开发, 刘嘉奕, 侯高攀, 等. 卫星互联网中层次化多模态数据安全共享方案. 中国科学: 信息科学, 2026, 56(03):581–602]

A password-based secure data transmission scheme for large language model agents

Mi SONG[1,2,3] & Ding WANG[1,2,3*]

1. *College of Cryptology and Cyber Science, Nankai University, Tianjin 300350, China*
2. *The Key Laboratory of Data and Intelligent System Security (Ministry of Education), Tianjin 300350, China*
3. *Tianjin Key Laboratory of Network and Data Security Technology, Tianjin 300350, China*

* Corresponding author. E-mail: wangding@nankai.edu.cn

Abstract Large language model (LLM) agents integrate task understanding, tool invocation, and autonomous decision-making, accelerating the transition of artificial intelligence technologies from content generation to task execution, and enabling broad applications in sensitive domains such as finance, healthcare, and public services. However, when invoking external tools, existing agents often lack direct authentication between users and tools, provide insufficient protection for private user data, and rely on coarse-grained access control. These limitations may lead to identity impersonation, unauthorized access, privacy leakage, and unclear trust boundaries, hindering secure data sharing in sensitive scenarios. To address these issues, this paper proposes a password-based secure data transmission scheme for LLM agents. In the considered user-LLM-tool interaction, a user and an external tool server authenticate each other using a low-entropy password and establish a high-entropy session key. The proposed scheme combines the negotiated session key with ciphertext-policy attribute-based encryption to enforce fine-grained authorization, ensuring that sensitive data can be decrypted and processed only by authenticated and authorized endpoints satisfying the specified access policy. The LLM executes task parsing, planning, and response-template generation without accessing the plaintext of the user's private input or the tool's output. Performance evaluation shows that the proposed scheme satisfies all twelve evaluation criteria. For a representative policy with five attributes, the total computation cost is 239 ms and the communication cost is 7,744 bits, which are comparable to those of existing schemes while providing broader functionality and security coverage. This paper uses a user-memorable password as the root of trust, and provides a secure and efficient solution for the secure transmission and access control of sensitive data in LLM-agent scenarios without changing the existing LLM-agent service architecture.

Keywords large language models, agent, identity authentication, fine-grained access control, privacy-preserving