

VeriFU: 可验证的联邦遗忘学习

江泽豪¹, 熊金波^{1,2}, 黄佳毅¹, 张媛媛^{1,2}, 田有亮³

(1. 福建师范大学计算机与网络空间安全学院, 福建 福州 350117;

2. 福建师范大学福建省网络安全与密码技术重点实验室, 福建 福州 350117;

3. 贵州大学大数据与信息工程学院, 贵州 贵阳 550025)

摘要: 联邦遗忘学习为客户提供遗忘贡献的支持, 然而, 客户端难以验证服务器是否诚实且有效地移除了其贡献。目前的大部分研究忽视了这一重要方面, 通过所构建的基于零知识证明的验证模型, 提出完整的可验证联邦遗忘学习框架, 并结合动态更新的默克尔树结构, 提出了一种可验证的联邦遗忘学习方案, 方案具有零知识性, 能够在保护其他客户端隐私数据不泄露的前提下高效生成服务器遗忘操作的密码学证明。通过实验评估了方案的有效性、计算开销, 与基于RSA (Rivest-Shamir-Adleman) 累加器和基于哈希链的方案相比, 当模型参数数量为 10^5 数量级时, 所提方案在证明生成速度上相较于RSA方案提升了约两个数量级, 在验证速度上相较于RSA方案提升了13.2倍, 且避免了哈希链方案验证开销随数据规模线性增长的瓶颈。

关键词: 联邦遗忘学习; 零知识证明; 默克尔树; 遗忘验证

中图分类号: TP181

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.AQ26024

VeriFU: verifiable federated unlearning

Jiang Zehao¹, Xiong Jinbo^{1,2}, Huang Jiayi¹, Zhang Yuanyuan^{1,2}, Tian Youliang³

1. College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China

2. Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350117, China

3. College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China

Abstract: Federated unlearning enables clients to withdraw their contributions from a global model. However, enabling clients to verify whether the server has honestly and effectively removed their contributions remains a critical challenge. To address this aspect, which has been largely overlooked in existing literature, a verification model based on zero-knowledge proofs was constructed, and a comprehensive framework for verifiable federated unlearning was proposed. Combined with a dynamically updated Merkle tree structure, a novel verifiable federated unlearning scheme was presented characterized by its zero-knowledge property. This allows for the efficient generation of cryptographic proofs for server unlearning operations while rigorously protecting the data privacy of other clients. We evaluate the effectiveness and computational overhead of the proposed scheme. Comparative experiments with Rivest-Shamir-Adleman (RSA) accumulator-based and Hash chain-based schemes demonstrate that, when the model parameter size reaches the order of 10^5 , the proposed scheme improves proof generation speed by

收稿日期: 2026-01-29; 修回日期: 2026-03-22

通信作者: 熊金波, jinbo810@163.com

基金项目: 国家自然科学基金资助项目 (No.62272102, No.62272123); 福建省自然科学基金资助项目 (No.2023J02014, No.2024J08163)

Foundation Items: The National Natural Science Foundation of China (No. 62272102, No. 62272123), The Natural Science Foundation Project of Fujian Province (No.2023J02014, No.2024J08163)

approximately two orders of magnitude and verification speed by 13.2 times compared to the RSA-based scheme. Furthermore, it effectively avoids the scalability bottleneck of data linear growth in verification overhead inherent in Hash chain-based schemes.

Key words: federated unlearning, zero-knowledge proof, Merkle tree, unlearning verification

0 引言

联邦学习 (federated learning, FL) 因其中心服务器和客户端只通过模型参数的方式进行交互, 能够在保护客户端隐私的基础上, 利用各客户端的数据, 以此有效解决数据孤岛问题, 当前备受关注^[1]。近年来, 各项法规如通用数据保护条例 (general data protection regulation, GDPR) 和加州消费者隐私法案 (California consumer privacy act, CCPA) 的颁布, 强化了利用数据者对数据整体的责任, 并且对数据提供者提供了“遗忘权”的保障^[2-4], FL无法支持遗忘操作以“消除”客户端贡献, 而联邦遗忘学习 (federated unlearning, FU) 基于FL的特点, 进一步提出并实现了从全局模型中删除特定客户端数据贡献的方案, 在保障客户端“遗忘权”的同时, 进一步强化了数据隐私保护^[5]。

目前大量的研究聚焦于解决两个问题: 其一, 如何在维持模型效能的前提下高效地消除客户端贡献; 其二, 在删除客户端贡献后, 模型可能会面临预测准确率显著下降的风险, 如何快速恢复模型精度或减小模型精度下降^[6]。这些研究为FU领域奠定了重要基础, 通过不同遗忘机制与模型恢复策略的协同优化, 实现了客户端数据贡献的有效移除与模型性能的高效恢复。但是, 从客户端的角度来看, 删除客户端贡献只是保障“遗忘权”的初步操作, 客户端可能更注重其贡献是否真正被完整地遗忘。因此, 构建可验证机制是彻底落实“遗忘权”的必要补充。然而, 现有研究在很大程度上缺乏对遗忘验证这一核心问题的探讨。相对于传统的机器遗忘学习可以通过模型对未学习数据的预测来判断遗忘效果以及使用成员推理攻击 (membership inference attack, MIA) 验证遗忘有效性, FU由于参与训练的客户端较多、模型参数聚合多个客户端贡献难以划分等问题, 造成遗忘难以验证。

零知识证明 (zero-knowledge proof, ZKP)

因其“零知识”的特性, 广泛应用于加密货币与区块链、身份验证和访问控制、安全多方计算等领域^[7]。为了解决以上问题, 通过构建非交互式的零知识证明系统对联邦遗忘学习的中心服务器遗忘操作进行证明与验证, 能在确保隐私信息不泄露的情况下, 生成遗忘客户端贡献的证明。然而零知识证明系统仅能对遗忘操作做数学形式的证明, 无法体现遗忘过程前后全局模型的状态, 因而服务器可能重新引入已删除的贡献。默克尔树 (Merkle tree) 是一种分层的数据结构, 其根节点哈希值可以作为数据的唯一标识, 任何对数据的修改都会导致根节点哈希值变化, 因而可快速检测数据的完整性^[8-9], 同时默克尔树通过验证路径上的节点哈希值, 即可做出成员证明, 而不需要遍历整棵树, 提高了成员证明的效率^[10-11]。因此, 本文的主要贡献在于对既有FU范式引入可验证性约束, 构建基于ZKP的验证模型, 并结合动态默克尔树, 确保遗忘过程的完整性和隐私保护。与可验证计算在FL中的应用相比^[12-13], 本文方案扩展到遗忘阶段, 提供高效的遗忘证明。本文主要研究工作如下。

1) 提出了FU中“可验证性”的完整性约束与安全需求, 构建了基于ZKP的验证模型。

2) 提出完整的可验证FU框架, 在框架基础上本文提出了一种简洁、可审计的遗忘证明方案VeriFU。通过ZKP和动态更新默克尔树的结合, 在不泄露模型参数的前提下, 实现客户端对服务器遗忘操作的验证, 同时通过默克尔树与时间戳的防篡改特性, 确保服务器无法重新注入已删除贡献。

3) 本文对VeriFU进行了充分实验, 评估了方案在不同标签分布下的有效性, 同时评估了在不同规模客户端参与下, 模型遗忘对FU过程的影响。此外, 对比了基于RSA (Rivest-Shamir-Adleman) 累加器和基于哈希链的方案, 当模型参数数量为 10^5 数量级时, 所提方案在验证开销上比RSA累加器方案减少0.521 2 ms, 比哈希链方案减少10.637 ms。

1 相关工作

FU 的大量研究集中于遗忘方法的改进和遗忘验证方法的设计两个主要方面, 因此本文梳理了这两个方面相关工作。

1.1 FU 遗忘方法

为了实现客户端能够撤回其影响这一目标, 研究者们设计了多种策略以有效消除客户端贡献, 可以大致分为重新训练、贡献删除、参数调整、训练校正、噪声干扰等几种遗忘类型^[14-16]。其中重新训练类型遗忘精度最高, 但是计算成本高、时间开销巨大^[17], Liu 等^[18]提出一种基于增量学习的快速重新训练算法, 通过近似黑塞 (Hessian) 矩阵, 引入动量技术, 不仅降低了重新训练的计算成本, 还减小了近似误差, 使模型收敛效率提升。然而, 频繁通信带来的开销依然显著。Su 等^[19]利用异步 FL 的特点, 通过集群划分客户端, 减少了重新训练带来的通信开销。相对于重新训练的高计算成本和高通信开销, 贡献删除方法可以实现快速遗忘, 降低通信成本, Wu 等^[4]通过知识蒸馏技术, 使用原始全局模型快速蒸馏训练新的学生模型, 不需要重新训练, 并且完全在服务器上进行遗忘操作, 显著降低了计算和通信成本, 但其遗忘效果依赖于原始模型, 且易产生偏差。参数调整方法直接修改模型参数, 如 Wang 等^[14]通过剪枝模型中的通道参数消除特定贡献, 不需要迭代训练, 但仅适用于卷积神经网络分类模型。Gao 等^[20]提出的 Scale-to-Unlearn (S2U) 方法通过在训练中缩放遗忘客户端的梯度并放大其他参与者的梯度, 以此削弱遗忘客户端对全局模型的特定贡献, 属于训练校正方法, 遗忘效率高, 但遗忘效果不够稳定, 且对动态数据环境的适应性有限。噪声干扰方法主要是添加可控的随机噪声模糊或掩盖客户端贡献, 能够提供隐私保证, 但模型性能下降较为明显, 且遗忘效果不稳定。

1.2 FU 遗忘验证方法

遗忘方法是 FU 的关键研究点, 随着对可验证遗忘的研究不断深入, FU 的验证方法也取得了较大的进展^[21-22]。其中文献^[23-24]提出的后门水印方法, 可以在 FU 中提供有效的遗忘验证, 但是, 若参与训练的客户端存在恶意行为, 可能引发后门攻击, 增加了一定的风险^[25]。Gao 等^[20]提出

的遗忘记忆 (forgettable memory, FM)、错误记忆 (Erroneous memory, EM) 方法, 通过预测独特样本、错误样本验证遗忘, 但在面临独特样本较少、错误样本错误率低等情况时, 验证的准确率可能下降。Eisenhofer 等^[26]提出可验证计算框架, 通过 ZKP 和哈希链等密码学技术要求服务器在每次迭代中生成证明, 验证模型是否已删除请求的数据点, 框架的核心在于初始化、训练以及遗忘函数, 如式(1)所示。

$$\begin{aligned}(\mathbf{st}_f, \mathbf{m}) &:= f_I(\mathbf{pp}_f) \\ (\mathbf{st}_f, \mathbf{m}) &:= f_T(\mathbf{st}_f, D^+) \\ (\mathbf{st}_f, \mathbf{m}) &:= f_U(\mathbf{st}_f, U^+)\end{aligned}\quad (1)$$

其中, $f_I()$ 表示初始化函数, $f_T()$ 表示训练函数, f_U 表示遗忘函数, $\mathbf{st}_f$ 为模型状态, $\mathbf{m}$ 为模型参数, $\mathbf{pp}_f$ 为公共超参数, D^+ 为要添加的数据点集合, U^+ 为要删除的数据点集合。该框架使用简洁非交互式零知识证明 (SNARKs) 和哈希链来实例化协议, 确保模型更新的可验证性, 然而, 哈希链在验证证明方面的效率较低。

翁嘉思等^[27]结合交互式 ZKP 和动态通用 RSA, 基于可公开可验证隐蔽安全模型, 实现了方案的公开可验证性。利用 RSA 累加器对训练数据进行认证, 生成累加值。

$$\text{acc} := g^{H_{\text{pnm}}(r)} \bmod n \quad (2)$$

其中, g 为未知阶群 G 的生成元, $H_{\text{pnm}}()$ 为抗碰撞哈希函数, r 为数据记录, n 为群 G 的模数。该方案通过 ZKP 协议, 允许数据拥有方验证模型训练和遗忘操作的正确性, 确保模型不使用未经授权的数据, 但是 RSA 累加器在生成证明和验证上效率较低。

综上所述, 现有的联邦遗忘学习遗忘验证方法在初步实现遗忘效果验证方面取得了进展, 这些工作为 FU 的可审计性奠定了基础, 然而, 现有的验证方案仍存在一定局限。此外, 可验证计算在联邦学习中也有广泛的应用, 如, Jia 等^[12]提出的学习证明 (Proof-of-Learning) 与 Qu 等^[13]提出的联邦学习证明 (Proof of Federated Learning) 主要聚焦于训练过程的诚实性与贡献公平性, 但尚未充分扩展至遗忘阶段的验证。

本文方案正是在这一背景下, 结合 ZKP 与动态默克尔树, 实现了高效的零知识遗忘证明,

并在不依赖特定遗忘方法的前提下,支持遗忘前后模型状态的完整性审计,进一步完善了现有研究在动态、可审计遗忘验证方面的空白。

2 预备知识

本节首先介绍FU的定义以及一般流程,接着介绍ZKP相关知识,最后介绍默克尔树的定义和构造方法。

2.1 联邦遗忘学习

FU在FL的基础上,为客户端提供“遗忘权”,通过特定的遗忘操作消除客户端贡献,进一步保护客户端隐私。FU流程如图1所示。

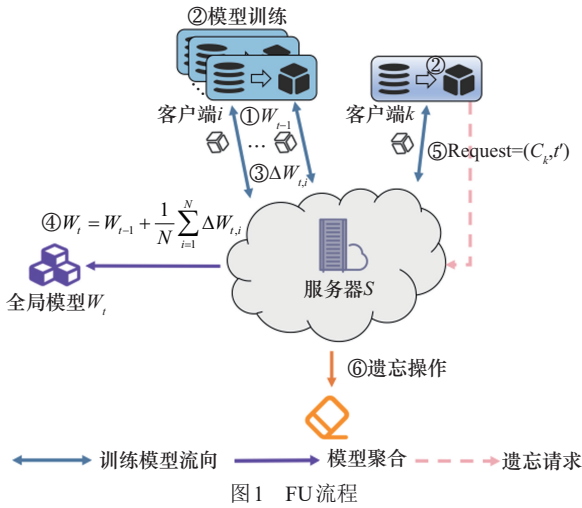


图1中, FU基于FL, 多个客户端与中央服务器协同配合, 若是客户端提出遗忘请求, 中央服务器需要根据相应的遗忘方法如基于知识蒸馏的联邦遗忘学习(FUKD)等^[5,28-33]进行客户端贡献的“擦除”, 并发起重新训练, 恢复模型精度。FU的问题可以表述为:

$$w_{\text{new}} = \underbrace{\sum_{i=1, i \neq k}^N \frac{n_i}{N - n_k} w^{(i)}}_{\text{权重重分配}} - \underbrace{\lambda \cdot R(w^{(k)})}_{\text{贡献消除项}} \quad (3)$$

如式(3)所示, 通过两阶段移除遗忘客户端 k 的影响, 第一阶段是权重重分配, 将全局模型参数重新计算为剩余客户端的加权平均(权重从 $\frac{n_i}{N}$ 调整为 $\frac{n_i}{N - n_k}$), 直接排除客户端 k 的显式贡献; 第二阶段是贡献消除项, 通过正则化项 $R(w^{(k)})$ (如梯度逆向更新 $\nabla w^{(k)}$ 或影响函数 $\nabla_w \mathcal{L}_k(w)$)消

除潜在非线性残留。其中 w_{new} 表示遗忘后的全局模型参数, N 为客户端数量, $w^{(i)}$ 为客户端 i 的本地模型参数, $N - n_k$ 为剩余总样本数, n_i 为客户端 i 的本地样本数, n_k 为客户端 k 的权重, λ 为控制遗忘强度的超参数。

2.2 零知识证明

ZKP由Goldwasser等^[34]提出, 在密码学中, 该协议的一方称为证明者, 用 P 表示, 另一方称为验证者, 用 V 表示。ZKP的核心在于 P 在不向 V 泄露任何有用的信息的情况下, 试图让 V 相信某个论断是正确的。ZKP一般包含以下阶段^[35]。

1) 证明。在这一阶段, 证明者 P 拥有一些秘密信息或特定的知识, P 需要向验证者 V 证明其了解这些信息或知识, 但又不能泄露具体的内容。证明者 P 可以通过一系列形式化的数学过程构造一个证明来实现其目标。

2) 挑战。在收到证明者 P 的证明后, 验证者 V 需要审查证明并对证明发起随机挑战。

3) 回应挑战。在收到验证者 V 的挑战后, 证明者 P 回应挑战, 通过将收到的随机数结合给出证明。返回挑战的回应, 这一回应必须满足验证者 V 要求, 同时不泄露具体信息。

4) 验证。验证者 V 验证挑战的回应是否正确, 如果正确, 则证明成功, 反之, 则证明失败。

ZKP需满足完整性、可靠性、零知识性3个关键性质^[36], 5.1节将对方案的3个性质进行证明。

2.3 默克尔树

默克尔树又称为哈希树, 是一种分层的数据结构, 其核心思想是通过对数据对象进行哈希运算得到哈希值并组成完全二叉树^[8]。设有一组数据 $D = \{d_1, d_2, \dots, d_m\}$, 默克尔树的构造需要计算每个数据块的哈希值作为叶节点, 因此每个数据块需进行哈希值计算, 得到叶节点集合 $H = \{h_1, h_2, \dots, h_m\}$, 然后向上递归合并相邻节点 h_i 和 h_j 的哈希值形成父节点哈希 $h_p = H(h_i || h_j)$, 其中 $||$ 表示拼接操作, 重复此过程, 直至产生根节点哈希值。根节点哈希值可以作为数据的唯一标识, 通过构建的默克尔树即可快速检测数据的完整性^[9], 同时默克尔树因其只需验证路径上的节点哈希值, 即可做出成员证明, 在成员证明方面具有高效率^[10,37]的特征。

3 问题描述

在典型的FU中,客户端可能出于隐私保护等需求提出遗忘请求,选择撤回其数据对全局模型的贡献,服务器应当对客户端的遗忘请求进行响应并正确删除客户端贡献。然而,客户端提出遗忘请求后,即使服务器告知客户端其贡献已成功删除,客户端无法对遗忘进行验证,亦难以确信服务器已诚实移除其贡献。因此,需要一种遗忘验证方案,在尽可能不影响FU训练和遗忘进程的情况下为客户端提供遗忘验证。此外,验证方案需要足够可信赖,不对FU系统造成安全隐患。接下来,主要介绍系统模型和威胁模型。

3.1 系统模型

系统模型如图2所示,主要由一个中央服务器和多个参与训练的客户端等实体组成。系统模型的工作流程如下。

1) 初始化阶段。在初始化阶段,服务器需根据任务需求构建证明电路,通过安全参数和算术电路生成证明与验证所需密钥,定义哈希函数并初始化默克尔树。

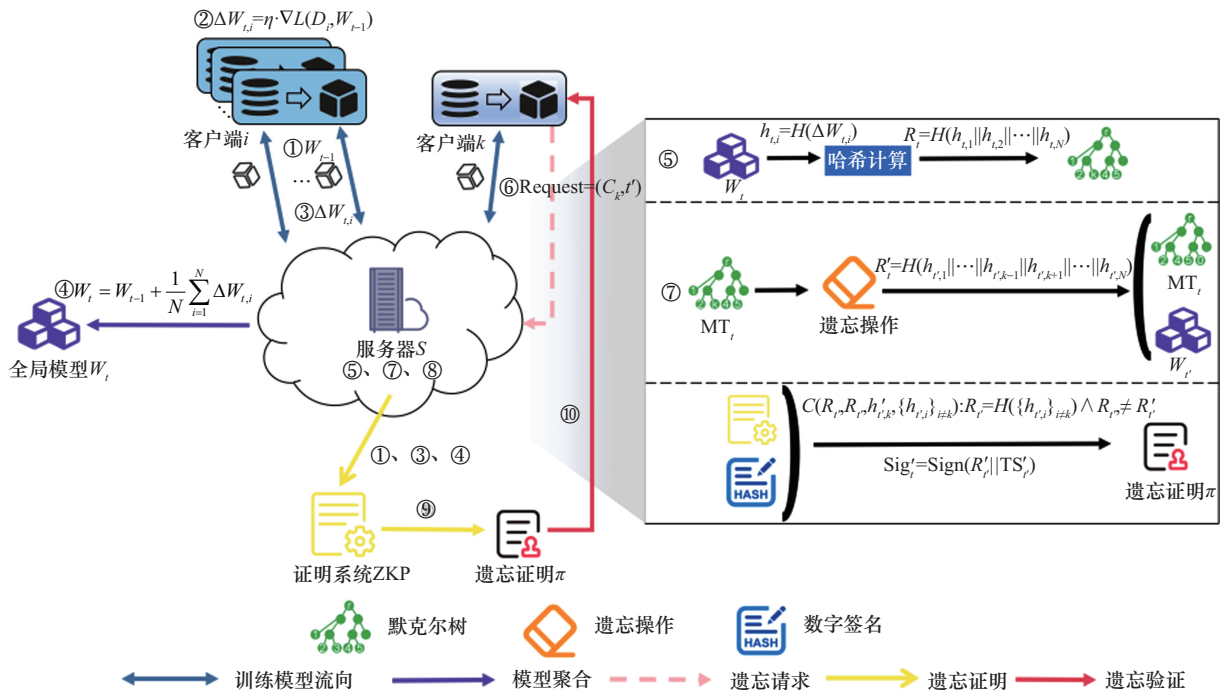
2) 训练阶段。在训练伊始,服务器将初始化模型参数、证明电路广播至各个客户端。对于每个参与训练的客户端而言,它们需根据本地数据

进行训练得到本地模型更新,计算模型更新的哈希值,并上传至服务器,服务器将接收到的局部模型更新哈希值加入默克尔树,对局部模型更新进行聚合得到新一轮的全局模型参数。

3) 遗忘与承诺证明阶段。模型训练的过程通常需要多轮迭代,在此过程中,做出“贡献”的客户端可以提出遗忘请求,要求服务器删除其“贡献”。服务器响应客户端遗忘请求后,执行遗忘操作,通过构建的默克尔树、ZKP电路以及签名和时间戳共同生成遗忘证明,并将遗忘证明发送至客户端。客户端可以根据服务器的遗忘证明验证遗忘有效性,以此实现可验证的FU方案。

3.2 威胁模型

本文考虑一个恶意的服务器模型,假定服务器控制全局模型以及聚合与分发的全部权限,且拥有全局模型 W 和所有客户端 C_i 上传的本地梯度更新 ΔW_i 。在该模型下,攻击者的潜在恶意行为包括:伪造遗忘证明或在诚实进行遗忘操作后重新引入已遗忘的模型参数。上述行为的主要动机是服务器不愿意承受遗忘操作导致的模型性能下降,或试图规避为恢复模型精度所需的再训练成本。为此,本文设计VeriFU方案来验证服务器是否满足客户端提出的遗忘请求,并通过默克



尔树的防篡改特性, 确保服务器无法在后续阶段重新引入已移除的模型。本文研究重点在于遗忘验证机制, 通过构建的ZKP系统在保障验证有效性的同时, 防止验证过程泄露其他客户端的隐私数据。此外, VeriFU具有良好的兼容性, 可与其他隐私保护机制协同使用以提供更全面的模型隐私保障。为便于参考, 本文涉及的数学符号见表1。

表1 数学符号

符号	含义
N	参与FU的客户端总数
T	总训练轮次
t	当前训练轮次, $t \in [1, 2, \dots, T]$
C_i	第 i 个客户端, $i \in [1, 2, \dots, N]$
D_i	客户端 C_i 的本地数据集
W_0	初始全局模型参数
W_t	第 t 轮的全局模型参数
$\Delta W_{t,i}$	第 t 轮客户端 C_i 的本地模型更新
$h_{t,i}$	客户端 C_i 在第 t 轮的贡献哈希
MT_t	第 t 轮的默克尔树
R_t	第 t 轮的默克尔树根
TS_t	第 t 轮的时间戳
Sig_t	第 t 轮的签名, $Sig_t = \text{Sign}(R_t)$
$\pi_{t,i}$	针对 $h_{t,i}$ 的默克尔树证明
$\pi_{zk,t,i}$	针对 $h_{t,i}$ 移除的ZKP
$H(\cdot)$	哈希特征函数
$\text{Sign}(\cdot)$	数字签名函数
$\text{Verify}(\cdot)$	验证函数
Q	ZKP算术电路
PK, VK	ZKP的证明和验证密钥

4 算法设计

VeriFU为FU中提出遗忘请求的客户端提供了准确、可靠的遗忘验证, 详细设计如方案1所示。

方案1 可验证的联邦遗忘学习 VeriFU
初始化阶段

1) $S: Q, H(\cdot), MT_0, (PK, VK) \leftarrow \text{keyGen}(1^\lambda, Q)$;

训练与聚合阶段

2) $S \rightarrow C: W_0, R_0$;

3) 第 t 轮次, $C_i: \text{Train}(D_i) \rightarrow \Delta W_{t,i}$;

4) $C_i: h_{t,i} = H(\Delta W_{t,i}) \rightarrow S$;

5) $S: W_t = W_{t-1} + \sum_i \Delta W_{t,i}$,

$MT_t, R_t \leftarrow \text{AddNode}(MT_{t-1}, h_{t,i})$;

6) $S: \text{Sig}_t = \text{Sign}(R_t || TS_t) \rightarrow C_i$;

遗忘与承诺证明阶段

7) 第 t 轮次, $C_k: \text{Request} = (C_k, t')$;

8) $S: MT_{t'} = \text{MerkleRemove}(MT_t, h_{t',k})$,

$R_{t'} = \text{MerkleRoot}(\{h_{t',i}\}_{i \neq k})$;

9) $S: \pi_{zk,t',k} = \text{Prove}(PK, R_t, R_{t'}, h_{t',k}, \{h_{t',i}\}_{i \neq k})$;

10) $S: \pi_{t',k} = \{h_j | j \in \text{path}(h_{t',k}, R_{t'})\}$;

11) $S: TS_{t'} = \text{time.time}(), \text{Sig}_{t'} = \text{Sign}(R_{t'} || TS_{t'})$,

$(R_{t'}, \pi_{zk,t',k}, \pi_{t',k}, \text{Sig}_{t'}, TS_{t'}) \rightarrow C_k$;

12) $C_k: \text{Verify}_{\text{Sig}}(\text{Sig}_{t'}, R_{t'} || TS_{t'})$,

$\text{Verify}(VK, \pi_{zk,t',k}, R_t, R_{t'}, h_{t',k})$,

$R_{\text{computed}} = \text{Reconstruct}(h_{t',k}, \pi_{t',k})$,

$TS_{t'} > TS_t$;

if 所有验证通过

return True

else

return False

4.1 VeriFU训练与聚合阶段

初始化阶段对任务所需证明电路以及参数初始化后, 进入训练与聚合阶段。服务器 S 发起训练, 初始化全局模型 W_0 和初始默克尔树树根 R_0 至参与训练的客户端; 在第 t 轮次, 客户端 C_i 根据私有数据 D_i 训练本地模型, 生成更新 $\Delta W_{t,i}$; 客户端 C_i 计算 $h_{t,i} = H(\Delta W_{t,i})$, 并将 $h_{t,i} = H(\Delta W_{t,i})$ 发送至服务器 S ; 服务器 S 聚合更新, 计算 $W_t = W_{t-1} + \sum_i \Delta W_{t,i}$, 将 $h_{t,i}$ 作为叶节点加入 MT_{t-1} , 重新计算树根 R_t ; 服务器 S 使用时间戳 TS_t 对 R_t 签名, 生成 $S: \text{Sig}_t = \text{Sign}(R_t || TS_t)$, 并广播 (R_t, Sig_t) 给客户端 C_i 。

4.2 VeriFU遗忘与承诺证明阶段

参与训练的客户端可以在上传模型参数后的任意时刻发起遗忘请求, 服务器对遗忘请求做出快速响应, 遗忘客户端模型参数后, 生成遗忘证明并交由客户端验证。遗忘与承诺证明阶段流程如图3所示。

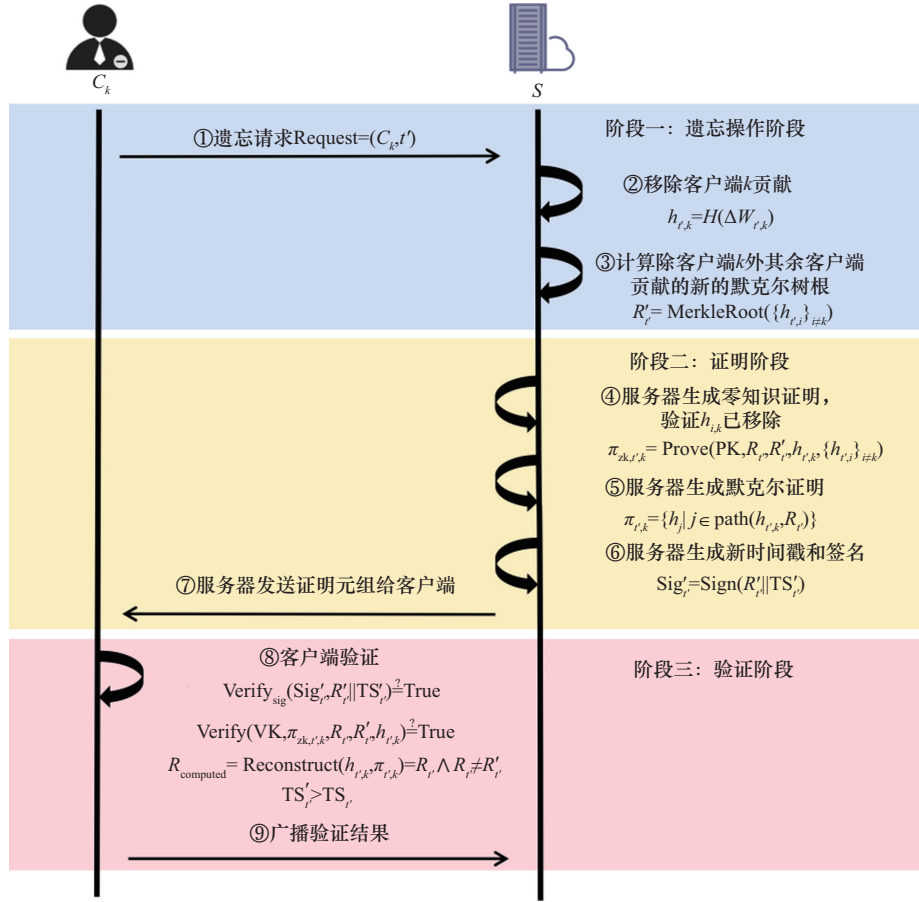


图3 遗忘与承诺证明阶段流程

详细过程如下。

客户端 C_k 提出遗忘请求 $\text{Request} = (C_k, t')$ 。

服务器 S 执行遗忘操作，从 $\text{MT}_{t'}$ 中移除 $h_{t',k} = H(\Delta W_{t',k})$ ，生成更新后的默克尔树 $\text{MT}'_{t'}$ ，计算新根默克尔 $R'_t = \text{MerkleRoot}(\{h_{t',i}\}_{i \neq k})$ 。

服务器生成 ZKP，验证 $h_{t',k}$ 已移除，具体如下。

公开输入： $R_t, R'_t, h_{t',k}$ 。

私有输入： $\{h_{t',i}\}_{i \neq k}$ 。

证明生成：

$\pi_{zk,t',k} = \text{Prove}(\text{PK}, R_t, R'_t, h_{t',k}, \{h_{t',i}\}_{i \neq k})$ 。

服务器生成默克尔证明 $\pi_{t',k}$ ，包含从 $h_{t',k}$ 到 R_t 的兄弟路径 $\pi_{t',k} = \{h_{t',j} \in \text{path}(h_{t',k}, R_t)\}$ 。

服务器 S 生成新时间戳 TS'_t 和签名 $\text{Sig}'_t = \text{Sign}(R'_t || \text{TS}'_t)$ ，并发送证明元组 $(R'_t, \pi_{zk,t',k}, \pi_{t',k}, \text{Sig}'_t, \text{TS}'_t)$ 给客户端 C_k 。

客户端 C_k 验证过程如下。

签名验证： $\text{Verify}_{\text{Sig}}(\text{Sig}'_t, R'_t || \text{TS}'_t)$ 。

ZKP 验证： $\text{Verify}(\text{VK}, \pi_{zk,t',k}, R_t, R'_t, h_{t',k})$ 。

默克尔证明验证：

$R_{\text{computed}} = \text{Reconstruct}(h_{t',k}, \pi_{t',k})$ ，

确认 $R_{\text{computed}} = R_t$ 且 $R'_t \neq R_t$ 。

时间戳验证： $\text{TS}'_t > \text{TS}_t$ 。

若所有验证通过，返回 True；否则，返回 False。

需特别指出的是，从默克尔树中移除客户端贡献的哈希值等同于全局模型完成了遗忘，依赖于 ZKP 算术电路对聚合逻辑的强约束。本文方案的 ZKP 电路不仅验证叶节点的移除，更严格约束了新的全局模型参数必须且仅能由剩余有效叶节点重新聚合生成。即 ZKP 电路证明了等式 $W'_t = \text{Agg}(\{\Delta W_{t,i}\}_{i \in [1,N] \setminus \{k\}})$ 严格成立，其中的 $\text{Agg}()$ 表示聚合函数。因此，只要 ZKP 验证通过，即在数学上保证了当前的全局模型参数中已删除了遗忘客户端的贡献，实现了树状态与模型状态的强绑定。

5 理论分析

VeriFU旨在保护客户端“遗忘权”，保证遗忘过程的真实性、公平性、合法性。将本文方案建模为交互式证明系统 (P, V) ，其中 P （服务器）生成遗忘证明， V （客户端）验证。声明 stmt 为不含客户端 V 的贡献且满足一致性规则， w 为有效见证， $\text{Pr}()$ 表示反映定事件发生的概率。

5.1 安全性分析

(1) 完备性

$$\Pr(\langle P(\text{stmt}, w), V(\text{stmt}) \rangle = 1 \mid \text{stmt 为真}) = 1 \quad (4)$$

如果服务器 S 诚实执行了遗忘操作并更新了默克尔树，则生成的证明必能以概率 100% 通过验证。方案基于 Groth16 协议构建，根据算术电路的可满足性，只要输入合法，生成的 ZKP 证明即有效，同时通过 RSA 签名确保真实性。

(2) 可靠性

$\Pr(\langle P^*(\text{stmt}), V(\text{stmt}) \rangle = 1 \mid \text{stmt 为假}) \leq \epsilon \quad (5)$
其中， ϵ 表示可忽略的概率上界。假设哈希函数 $H(\cdot)$ 具有抗碰撞性，且 ZKP 协议满足可靠性，则恶意服务器在未真实删除数据的情况下，伪造有效证明的概率是可以忽略的。本文方案中默克尔树的安全性依赖于哈希函数的抗碰撞性，若服务器未移除叶节点 L_i 或违反一致性规则，则其除非找到哈希碰撞，否则无法生成一个新的根节点 R' 使得 $\text{Verify}(\text{VK}, \pi_{\text{zk}, i', k}, R_{i'}, R_{i'}, h_{i', k}) = \text{True}$ 。此外，ZKP 的可靠性确保服务器无法在不拥有有效见证 w 的情况下生成有效证明。

(3) 零知识性

零知识性要求存在模拟器 S ，使视图不可区分。

$$\text{View}_{V^*}(\langle P, V^* \rangle) \approx S(\text{stmt}) \quad (6)$$

由于 VeriFU 仅公开默克尔树证明和 ZKP 证明，ZKP 确保 V^* 仅获知证明真伪，而不泄露参数信息，攻击者无法从证明中反推其他客户端的贡献信息。

(4) 抗重放攻击

证明元组 $(R_{i'}, \pi_{\text{zk}, i', k}, \pi_{i', k}, \text{Sig}_{i'}, \text{TS}_{i'})$ 中包含时间戳 $\text{TS}_{i'}$ 和对根哈希、时间戳的 RSA 签名。客户端在验证时执行以下检查。

签名验证: $\text{Verify}_{\text{Sig}}(\text{Sig}_{i'}, R_{i'} \parallel \text{TS}_{i'})$ 。

时间戳验证: $\text{TS}_{i'}' > \text{TS}_{i'}$ 。

若服务器重复发送旧证明，则时间戳验证失败，证明被拒绝。该机制有效防止重放攻击，同时时间戳与签名结合确保证明的时效性和不可否认性。

综上，VeriFU 在 ZKP 的密码学保证基础上，满足完备性、可靠性和零知识性。同时，通过时间戳与签名机制有效抵御重放攻击。零知识性确保遗忘验证过程不泄露其他客户端隐私数据，有效保护其他客户端隐私数据。可靠性依赖 ZKP 电路的严格约束和 RSA 签名的不可伪造性，整体安全性在恶意服务器模型下成立。

5.2 复杂度分析

VeriFU 包含 N 个客户端，系统训练 T 轮，在第 F 轮客户端提出遗忘请求，执行遗忘操作并进行遗忘验证。本文方案侧重点为遗忘验证，因此本节主要是对遗忘验证相关流程复杂性进行分析。

(1) 时间复杂度

假设每个参数更新可分割成大小为 B 的块， N 个客户端的总块数量为 K ，构建默克尔树时，叶节点数量即块数量 K ，树高为 $\lceil \log_2 K \rceil$ ，因此一次默克尔树的更新节点操作复杂度为 $O(\log_2 K)$ ，此外，每次更新默克尔树后需要重建树，默克尔树每次节点合并需要计算一次哈希，总哈希次数为 $K - 1$ ，复杂度为 $O(K)$ ，因此，一次默克尔树的更新操作复杂度为 $O(K \cdot \log_2 K)$ ， T 个轮次下的默克尔树的操作复杂度为 $O(T \cdot K \cdot \log_2 K)$ 。

时间戳使用 `time.time()` 生成，复杂度为 $O(1)$ ，时间戳长度为常数。签名使用 RSA 生成签名，输入根哈希和时间戳，签名操作复杂度也为 $O(1)$ 。

ZKP 系统构造采用 Groth16 协议，根据 Li 等^[36]、Groth 等^[38]对 Groth16 协议的分析，生成证明的时间复杂度依赖于算术电路规模 $|C|$ ，因此证明时间复杂度为 $O(|C|)G_0$ ，其中 G_0 表示群上操作，而验证仅需固定数量的配对操作^[39]，因此，验证时间复杂度为 $O(1)$ 。

签名验证复杂度为 $O(1)$ ，默克尔树验证与路径长度相关为 $O(\log_2 K)$ 。

综上，遗忘验证相关步骤的总时间复杂度为：

$$O(T \cdot K \cdot \log_2 K + 1 + 1 + 1 + |C|G_0) \quad (7)$$

等价于：

$$O(T \cdot K \cdot \log_2 K + |C|G_0) \quad (8)$$

(2) 空间复杂度

默克尔树有 K 个叶节点, 总节点数为 $2K - 1$, 假设每个节点 L 字节, 默克尔树的空间复杂度为:

$$O((2K - 1) \cdot L) = O(K \cdot L) \quad (9)$$

由于时间戳、签名长度以及 ZKP 大小均为常数级别, 因此其空间均为 $O(1)$ 。

综上, 遗忘验证总空间复杂度为:

$$O(K \cdot L + 1 + 1 + 1) = O(K \cdot L) \quad (10)$$

6 性能分析

6.1 实验设置

为全面评估 VeriFU 方案, 所设计的实验包括: 验证有效性实验、计算效率实验、默克尔树运行操作实验以及对比实验。实验物理环境为 Ubuntu20.04, 服务器拥有 1 块 RTX 3080 显卡, 共 10.5 GB 显存, 使用 Intel Xeon 处理器, 代码环境使用 Python3.9, 模型训练和评估基于 PyTorch2.1.1 和 CUDA11.8。ZKP 通过 Libsnark 编写生成, 默克尔树哈希、签名等加密操作使用 Cryptography 库。实验使用 MNIST、Fashion-MNIST 和 CIFAR-10 数据集^[40-41], 模型分别采用 LeNet-5、ConvNet 以及 ResNet-18, 3 个数据集和模型总结见表 2, 实验参数见表 3。

表 2 3 个数据集和模型

数据集	类别数/个	样本数/个	分辨率	模型
MNIST	10	70 000	28 像素×28 像素	LeNet-5
Fashion-MNIST	10	70 000	28 像素×28 像素	ConvNet
CIFAR-10	10	60 000	32 像素×32 像素	ResNet-18

表 3 实验参数

参数	值	说明
学习率	0.01	—
本地训练轮数	10 轮	客户端本地训练轮次
批量大小	64	训练与测试批量
总训练轮次	100 轮	全局训练轮次
客户端总数	10 个	默认为 10 个, 可扩展至 100 个客户端
遗忘客户端比例	10%	—

6.2 实验分析

基于上述设置, 本文首先对方案进行了验证有效性实验, 遗忘有效性实验旨在验证贡献移除和模型一致性, 以此验证模型遗忘与验证操作真实有效。实验通过对 3 个数据集进行独立同分布 (IID) 和不同参数 α 的非独立同分布 (Non-IID) 的数据划分, 每个客户端数据量均等, 且客户端数据之间没有重叠, 有效性实验数据分布如图 4 所示。在不同实验设置下进行 FL 训练, 训练 100 轮后, 遗忘客户端发起遗忘请求。为了量化方案的有效性, 本文实验采用个体梯度减法 (IGS)^[20] 作为遗忘算法, 即服务器通过计算并直接从全局模型中减去目标客户端的历史累计梯度以精确剔除其贡献。在此基础上, 服务器通过更新默克尔树移除其参数哈希, 利用 ZKP 生成证明, 并重新聚合全局模型。为评估一致性, 额外训练一个基线模型, 即不包含遗忘客户端数据重新训练的模型, 作为预期状态参考。有效性实验数据分布见表 4, 无论是在 IID 场景还是 Non-IID 场景下, 执行遗忘操作后, 模型在遗忘数据集上的测试准确率均出现显著下降。实验结果表明, 数据异构程度越高, 遗忘效果越显著。具

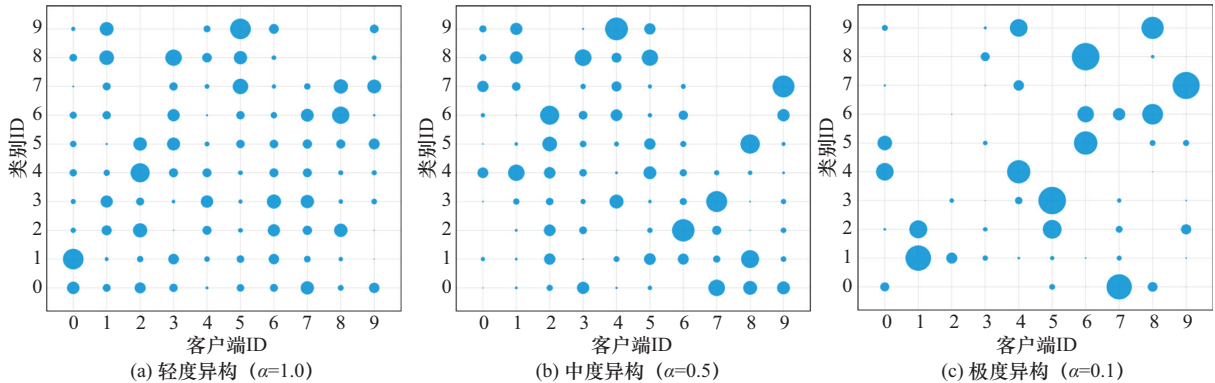


图 4 有效性实验数据分布

表4 有效性实验数据

数据集	数据分布情况	遗忘前准确率	遗忘后保留数据集准确率	遗忘后遗忘数据集准确率	保留数据集遗忘前后准确率差值 $\Delta Acc1$	遗忘数据集遗忘前后准确率差值 $\Delta Acc2$
MNIST	IID	98.93%	97.91%	96.53%	1.02%	2.40%
	Non-IID $\alpha = 1.0$	98.64%	88.51%	63.14%	10.13%	35.5%
	Non-IID $\alpha = 0.5$	97.97%	86.85%	49.28%	11.12%	48.69%
	Non-IID $\alpha = 0.1$	91.75%	59.93%	33.51%	31.82%	58.24%
Fashion-MNIST	IID	94.41%	90.96%	83.90%	3.45%	10.51%
	Non-IID $\alpha = 1.0$	93.65%	79.28%	59.06%	14.37%	34.59%
	Non-IID $\alpha = 0.5$	92.86%	70.08%	42.54%	22.78%	50.32%
	Non-IID $\alpha = 0.1$	87.89%	46.03%	29.13%	41.58%	58.76%
CIFAR-10	IID	84.50%	79.48%	70.57%	5.02%	13.93%
	Non-IID $\alpha = 1.0$	75.39%	64.22%	50.19%	11.17%	25.20%
	Non-IID $\alpha = 0.5$	73.32%	53.18%	36.75%	20.14%	36.57%
	Non-IID $\alpha = 0.1$	63.11%	37.12%	27.39%	25.99%	35.72%

体而言, Non-IID 场景下的准确率降幅显著高于 IID 场景, 这主要归因于 Non-IID 数据的强异质性, 使得模型对特定客户端的独特特征依赖性更强, 一旦移除该贡献, 其推理能力便会大幅衰退。此外, 模型在遗忘数据集上的性能衰退幅度远高于保留数据集, 充分证明了该方案在有效剔除目标知识的同时, 较好地维持了模型的全局可用性。

计算效率实验评估证明生成和验证与通信开销如图5所示, 测量 VeriFU 生成和验证时间, 记录证明大小及广播至所有客户端的总通信量, 采用轻量化电路设计和批量广播优化效率, 每种规模重复 100 次取平均值, 指标包括生成时间、验证时间与通信开销。实验数据表明, 客户端数增加使所提方案生成时间有略微提升, 在验证时间上接近一致, 因而证明该方案适合大量客户端情况下的 FU。通信开销随着客户端数量增加呈线性增长, 但在 100 个客户端参与时仍控制在较小的通信开销内。

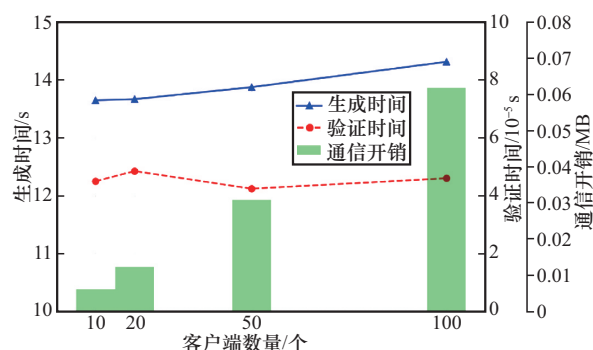


图5 计算效率实验评估证明生成和验证与通信开销

本文进一步基于 VeriFU 中的默克尔树, 针对不同客户端数量, 在 FU 场景下评估了其性能表现, 具体测量了构建树、单次添加数据、单次删除数据、生成成员证明、验证成员证明和频繁更新操作的时间。其中, 频繁更新操作分为多次添加操作和删除操作, 采用随机比例, 确定更新操作次数, 确保动态性。默克尔树运行操作时间开销如图6所示, 实验结果表明, 构建树、单次添加数据、单次删除数据和生成成员证明的操作时间随客户端数量的增加呈现显著增长, 相比之下, 验证成员证明的时间开销增长趋势较为平缓, 与理论复杂度 $O(\log_2 K)$ 相符。然而, 频繁更新操作的时间会随着客户端数量增加而快速增长, 在大规模且客户端频繁更新的场景下可能会形成性能瓶颈, 未来优化可以通过增量更新策略或稀疏默克尔树等方法提升大规模场景下的性能。

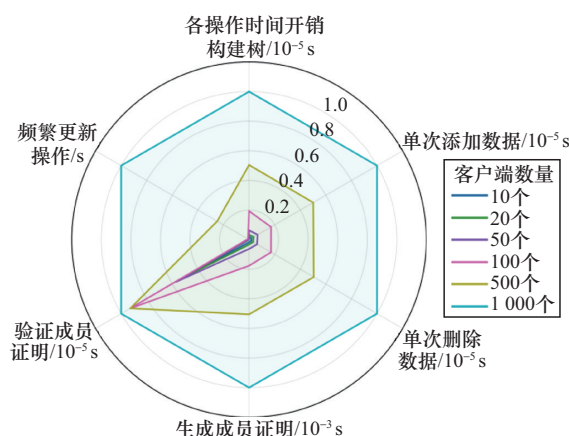


图6 默克尔树运行操作时间开销

Groth16、BCTV14与GM17协议证明、验证操作时间开销对比实验结果如图7所示,实验结果中Groth16在证明生成时间以及证明验证时间上明显优于BCTV14和GM17。此外,从图7可以看出,VeriFU使用的ZKP随着客户端数量增加并未显著提升在证明生成时间和验证时间上的开销,这与Groth16生成证明的时间复杂度依赖于算术电路规模 $|C|$ 、完全不依赖公开输入与证人数量且验证仅需固定数量的配对操作相符。因此本文方案使用Groth16可以确保在客户端数量大规模扩展时仍保持常数级证明生成和验证开销。

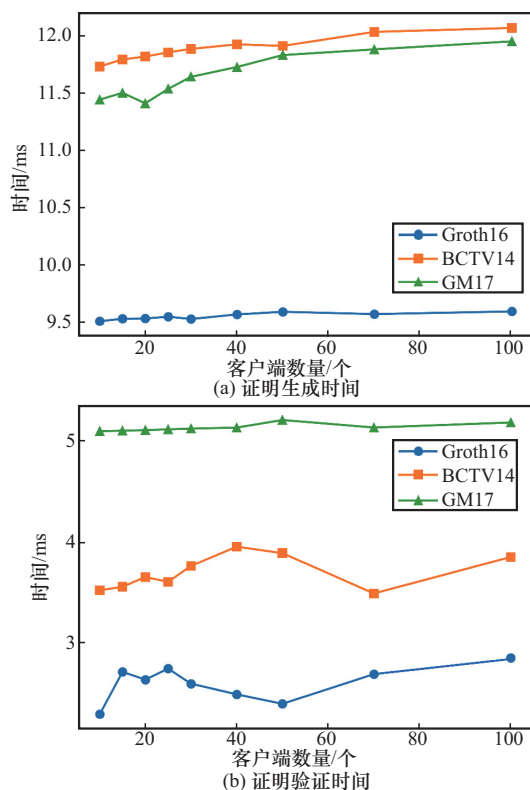


图7 Groth16、BCTV14与GM17协议证明、验证操作时间开销对比实验结果

更进一步,实验对比了基于默克尔树、哈希链和RSA累加器添加数据、生成成员证明与验证成员证明的频次,实验结果如图8所示。图8(c)中,尽管RSA累加器方案的验证时间复杂度为 $O(1)$,但由于其底层依赖高昂的模幂运算,单次验证耗时远高于默克尔树方案。相比之下,默克尔树的轻量级哈希运算实现了RSA累加器方案约13.2倍的性能提升。此外,哈希链方案的验证开销随数据规模线性增长,当处于大规模数据场景时其验证开销较大。

因此,默克尔树方案在验证证明表现上最优。如图8(b)所示,RSA累加器方案生成成员证明的开销巨大,在 10^5 数据规模情况下证明开销高达约10s/次,并不适用于实时性较高的联邦遗忘学习系统。而默克尔树方案证明开销仅为RSA累加器方案的约0.93%。在添加数据方面,本文所提方案的效率低于RSA累加器方案,与哈希链方案接近。总体来看,默克尔树在遗忘证明和验证方面综合性能最佳,适合低时延、高验证需求的应用场景。

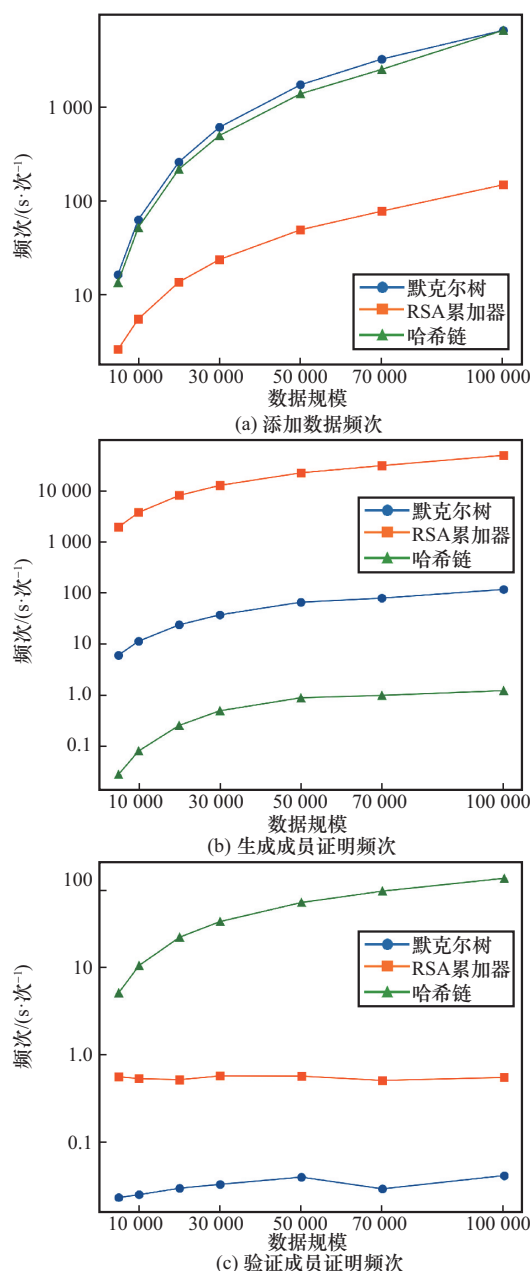


图8 默克尔树、哈希链与RSA累加器添加数据、生成成员证明与验证成员证明频次对比实验结果

为全面评估 VeriFU 作为一个完整验证框架的实用性,本文参考 Gao 等^[20]提出的评估标准,在 CIFAR-10 数据集上对其验证开销进行了对比分析。本文将 VeriFU 与文献[20]中代表性的验证方案进行了实践与空间开销的比较,具体包括:基于后门注入的验证方案 (BN)、基于模型嵌入的验证方案 (ME)、基于边界指纹的验证方案 (BF)、基于错误记忆的验证方案 (EM) 以及基于可验证记忆的验证方案 (FM)。验证开销对比实验结果见表 5。结果表明,本文方案的综合验证开销显著优于大部分验证方案,尽管 VeriFU 在时间开销上略高于 FM,在空间开销上略高于 BN、EM 与 FM,但这主要源于验证机制的本质差异:FM、EM 仅能提供概率性的推断,并且在非独立同分布场景下存在显著的误判风险,而 BN 的后门注入操作可能会引入额外的安全隐患。相比之下,VeriFU 依托零知识证明的完备性,在实现严格数学证明验证的同时,依然维持了较高的时间与空间效率,避免了传统遗忘验证因大量模型推理引发的性能瓶颈。

表5 基于 CIFAR-10 数据集的验证开销对比

方法	时间开销/s	空间开销/kB
BN	279.6	4
ME	97.1	12
BF	131.7	1 233
EM	107.2	4
FM	11.7	4
本文方案	13.7	6.2

7 结束语

本文基于 ZKP 和默克尔树,针对 FU 中客户端“遗忘权”缺乏验证操作研究的问题,提出 VeriFU。此外,通过形式化的数学理论证明 VeriFU 中 ZKP 的完备性、可靠性和零知识性,证明本文方案能够有效抵御伪造证明以及部分删除贡献的攻击。最后,本文对 VeriFU 的有效性、计算开销进行实验分析。实验结果表明,VeriFU 能有效地保证客户端验证操作的准确率和速度,然而,VeriFU 在遗忘验证的维度上仍有进一步拓展的空间。目前,本文所构建的验证框架主要实

现了客户端级别的整体贡献遗忘验证,下面将致力于研究特定样本级或特定类别级的更细粒度的遗忘验证方案。

参考文献:

- [1] 周传鑫,孙奕,汪德刚,等. 联邦学习研究综述[J]. 网络与信息安全学报, 2021, 7(5): 77-92.
Zhou C X, Sun Y, Wang D G, et al. Survey of federated learning research[J]. Chinese Journal of Network and Information Security, 2021, 7(5): 77-92.
- [2] Intersoft Consulting. General data protection regulation (GDPR)[R]. 2016.
- [3] Purtova N. The law of everything. Broad concept of personal data and future of EU data protection law[J]. Law, Innovation and Technology, 2018, 10(1): 40-81.
- [4] Wu C, Zhu S C, Mitra P. Federated unlearning with knowledge distillation[PP]. aeXiv(2022-01-24) [2026-01-29]. arXiv: arXiv. 2201.09441.
- [5] 王鹏飞,魏宗正,周东生,等. 联邦忘却学习研究综述[J]. 计算机学报, 2024, 47(2): 396-422.
Wang P F, Wei Z Z, Zhou D S, et al. A survey on federated unlearning[J]. Chinese Journal of Computers, 2024, 47(2): 396-422.
- [6] Ginart A, Guan M, Valiant G, et al. Making AI forget you: data deletion in machine learning[C]//Proceedings of the 33rd International Conference on Neural Information Processing Systems, 2019: 3518-3531.
- [7] Sun X Q, Yu F R, Zhang P, et al. A survey on zero-knowledge proof in blockchain[J]. IEEE Network, 2021, 35(4): 198-205.
- [8] Merkle R C. A digital signature based on a conventional encryption function[C]//Conference on the Theory and Applications of Cryptographic Techniques on Advances in Cryptology. Berlin: Springer, 1988: 369-378.
- [9] Kreutzer M, Niederhagen R, Shrishak K, et al. Quotable signatures using merkle trees[C]//GI-Jahrestagung 2019. Bonn: GI, 2019: 473-477.
- [10] 刘芳,蔡永泉,王朝翔,等. Merkle树遍历机制的改进及应用研究[J]. 计算机工程与应用, 2012, 48(14): 98-101, 167.
Liu F, Cai Y Q, Wang C X, et al. Research of traverse and application of Merkle authentication tree[J]. Computer Engineering and Applications, 2012, 48(14): 98-101, 167.
- [11] Kuznetsov O, Rusnak A, Yezhov A, et al. Efficient and universal Merkle tree inclusion proofs via OR aggregation[J]. Cryptography, 2024, 8(3): 28.
- [12] Jia H R, Yaghini M, Choquette-Choo C A, et al. Proof-of-learning: definitions and practice[C]//Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP). Piscataway: IEEE Press, 2021: 1039-1056.

- [13] Qu X D, Wang S L, Hu Q, et al. Proof of federated learning: a novel energy-recycling consensus algorithm[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 32(8): 2074-2085.
- [14] Wang J X, Guo S, Xie X, et al. Federated unlearning via class-discriminative pruning[C]//*Proceedings of the ACM Web Conference 2022*. New York: ACM Press, 2022: 622-632.
- [15] Liu Z Y, Ye H Y, Jiang Y, et al. Privacy-preserving federated unlearning with certified client removal[J]. *IEEE Transactions on Information Forensics and Security*, 2025, 20: 3966-3978.
- [16] Baumhauer T, Schöttle P, Zeppelzauer M. Machine unlearning: linear filtration for logit-based classifiers[J]. *Machine Learning*, 2022, 111(9): 3203-3226.
- [17] Sekhari A, Acharya J, Kamath G, et al. Remember what you want to forget: Algorithms for machine unlearning[J]. *Advances in Neural Information Processing Systems*, 2021, 34: 18075-18086.
- [18] Liu Y, Xu L, Yuan X L, et al. The right to be forgotten in federated learning: an efficient realization with rapid retraining[C]//*Proceedings of the IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. Piscataway: IEEE Press, 2022: 1749-1758.
- [19] Su N X, Li B C. Asynchronous federated unlearning[C]//*Proceedings of the IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*. Piscataway: IEEE Press, 2023: 1-10.
- [20] Gao X S, Ma X J, Wang J Y, et al. VeriFi: towards verifiable federated unlearning[J]. *IEEE Transactions on Dependable and Secure Computing*, 2024, 21(6): 5720-5736.
- [21] Guo C, Goldstein T, Hannun A, et al. Certified data removal from machine learning models[PP]. *arXiv(2023-11-08) [2026-01-29]*. *arXiv*: arXiv.1911.03030.
- [22] Thudi A, Deza G, Chandrasekaran V, et al. Unrolling SGD: understanding factors influencing machine unlearning[C]//*Proceedings of the 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. Piscataway: IEEE Press, 2022: 303-319.
- [23] Sommer D M, Song L W, Wagh S, et al. Athena: probabilistic verification of machine unlearning[J]. *Proceedings on Privacy Enhancing Technologies*, 2022, 2022(3): 268-290.
- [24] Uchida Y, Nagai Y, Sakazawa S, et al. Embedding watermarks into deep neural networks[C]//*Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. New York: ACM Press, 2017: 269-277.
- [25] 李璇, 邓天鹏, 熊金波, 等. 基于模型后门的联邦学习水印[J]. *软件学报*, 2024, 35(7): 3454-3468.
Li X, Deng T P, Xiong J B, et al. Federated learning watermark based on model backdoor[J]. *Journal of Software*, 2024, 35(7): 3454-3468.
- [26] Eisenhofer T, Riepel D, Chandrasekaran V, et al. Verifiable and provably secure machine unlearning[PP]. *arXiv(2025-03-05)[2026-01-29]*. *arXiv*: arXiv. 2210.09126.
- [27] 翁嘉思, 辜燕云, 刘家男, 等. 公开可验证的模型遗忘方案[J]. *计算机学报*, 2025, 48(2): 477-496.
- [28] Weng J S, Gu Y Y, Liu J N, et al. A publicly accountable machine unlearning method[J]. *Chinese Journal of Computers*, 2025, 48(2): 477-496.
- [29] Fraboni Y, Van Waerebeke M, Scaman K, et al. Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization[C]//*International Conference on Artificial Intelligence and Statistics*. PMLR, 2024: 3457-3465.
- [30] Zhu X R, Li G Y, Hu W. Heterogeneous federated knowledge graph embedding learning and unlearning[C]//*Proceedings of the ACM Web Conference 2023*. New York: ACM Press, 2023: 2444-2454.
- [31] Halimi A, Kadhe S, Rawat A, et al. Federated unlearning: how to efficiently erase a client in FL? [PP]. *arXiv(2023-10-20) [2026-01-29]*. *arXiv*: arXiv.2207.05521.
- [32] Gong J N, Kang J, Simeone O, et al. Forget-SVGD: particle-based Bayesian federated unlearning[C]//*Proceedings of the 2022 IEEE Data Science and Learning Workshop (DSLW)*. Piscataway: IEEE Press, 2022: 1-6.
- [33] Wang H Z, Zhu X J, Chen C, et al. Goldfish: an efficient federated unlearning framework[C]//*Proceedings of the 2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Piscataway: IEEE Press, 2024: 252-264.
- [34] Liu G Y, Ma X Q, Yang Y, et al. FedEraser: enabling efficient client-level data removal from federated learning models[C]//*Proceedings of the 2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. Piscataway: IEEE Press, 2021: 1-10.
- [35] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems[C]//*Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*. New York: ACM Press, 1985: 291-304.
- [36] Blum M, Blum M, Feldman P, et al. Non-interactive zero-knowledge and its applications[C]//*Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*. New York: ACM Press, 1988: 103-112.
- [37] 李威翰, 张宗洋, 周子博, 等. 简洁非交互零知识证明综述[J]. *密码学报*, 2022, 9(3): 379-447.
Li W H, Zhang Z Y, Zhou Z B, et al. An overview on succinct non-interactive zero-knowledge proofs[J]. *Journal of Cryptologic Research*, 2022, 9(3): 379-447.
- [38] Naor M, Nissim K. Certificate revocation and certificate update[J]. *IEEE Journal on Selected Areas in Communications*, 2000, 18(4): 561-570.
- [39] Groth J. On the size of pairing-based non-interactive arguments[C]//*Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin: Springer, 2016: 305-326.
- [40] Bagheri K, Kohlweiss M, Siim J, et al. Another look at extraction and randomization of Groth's zk-SNARK[C]//*International Conference on Financial Cryptography and Data Security*. Berlin: Springer, 2021: 457-475.
- [41] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998,

86(11): 2278-2324.

- [41] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE Press, 2016: 770-778.

[作者简介]



江泽豪 (2000-), 男, 福建师范大学硕士生, 主要研究方向为数据安全性与隐私保护、人工智能安全。



熊金波 (1981-), 男, 博士, 福建师范大学教授、博士生导师, 主要研究方向为数据安全性与隐私保护、人工智能安全等。



黄佳毅 (1997-), 男, 福建师范大学博士生, 主要研究方向为数据安全性与隐私保护、人工智能安全。



张媛媛 (1992-), 女, 博士, 福建师范大学讲师、硕士生导师, 主要研究方向为数据安全性与隐私保护、人工智能安全等。



田有亮 (1982-), 男, 博士, 贵州大学教授、博士生导师, 主要研究方向为算法博弈论、密码学与安全协议、大数据安全性与隐私保护等。