公开可验证的模型遗忘方案

翁嘉思[1)] 辜燕云[1)] 刘家男[2)] 李 明[1)] 翁 健[1)]

[1)](暨南大学网络空间安全学院 广州 510632)

[2)](东莞理工学院计算机科学与技术学院 广东 东莞 523808)

摘 要 全球数字化进程的加速伴随着数据主体信息失控现象日益显著。国内外数据安全相关法律相继出台,其中遗忘权(the Right to Be Forgotten)强调了数据主体拥有从数据使用方撤回其数据的权利。模型遗忘(Machine Unlearning)是机器学习领域践行遗忘权的技术,允许模型拥有方(即数据使用方)从已训练的模型中遗忘原本训练数据的指定数据,以满足数据拥有方撤回其数据的需求。现有针对模型遗忘效果的验证方法通常假设存在一个从未使用过被遗忘数据的基准模型,并通过测量遗忘后模型和基准模型的参数分布或输出分布是否足够相似来完成验证。然而,在恶意攻击场景下,模型拥有方容易伪造遗忘后模型的参数和输出分布,且模型参数通常难以归因于特定的训练数据,导致验证方难以有效验证目标模型是否遗忘其数据。本文提出了一种新的公开可验证模型遗忘方案,该方案在数据拥有方和模型拥有方之间执行,并在模型拥有方出现恶意行为时,数据拥有方能够生成任意第三方可验证的不可否认凭证。具体地,数据拥有方先利用动态通用累加器来认证被授权使用的数据或删除不被授权使用的数据;随后,模型拥有方在公开可验证隐蔽模型下证明模型训练使用了被累加数据或没有使用不被累加数据;最后,数据拥有方验证证明的有效性,若发现模型拥有方使用了未授权数据,则其生成公开可验证的凭证来追责模型拥有方的不合法行为。实验评估了不同数据量下证明和验证的计算开销,同时评估了不同数据点删除对模型预测结果的影响。

关键词 机器学习;数据安全;遗忘权;模型遗忘;可验证性

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2025.00477

A Publicly Accountable Machine Unlearning Method

WENG Jia-Si[1)] GU Yan-Yun[1)] LIU Jia-Nan[2)] LI Ming[1)] WENG Jian[1)]

[1)](Department of Cyber Security, Jinan University, Guangzhou 510632)

[2)](Department of Cyber Security, Dongguan University of Technology, Dongguan, Guangdong 523808)

Abstract The rapid pace of global digitalization has brought about increasingly significant challenges related to the loss of control over personal information. In response to these challenges, the laws and regulations for protecting data security have been introduced, both domestically and internationally. Among these, the rule of "the Right to Be Forgotten" first established under the General Data Protection Regulation (GDPR), grants data owners the right to request the removal of their data from data users. Machine Unlearning is a technique in machine learning that embodies this right, enabling the model owner (i.e., the data user) to remove specific data from a trained model to fulfill the data owner's request to withdraw their data.

收稿日期:2024-06-25;在线发布日期:2024-12-04。本课题得到国家自然科学基金青年项目(62302192, 62102166)、国家自然科学基金重点项目(62332007)、国家自然科学基金联合项目(U23A20303)、广东省自然科学基金面上项目(2024A1515010086)、广州市科技计划项目(2024A04J3691,2024A03J0464)、中国博士后科学基金第17批特别资助项目(2024T170348)、江苏省机器学习与网络空间安全跨学科研究工程中心,中央高校基本科研专项资金及东莞市社会发展科技重点项目(20231800940342)的资助。**翁嘉思**,博士,副研究员,主要研究领域为隐私保护的机器学习。E-mai:wengiasi@gmail.com。**辜燕云**,硕士研究生,主要研究方向为机器学习安全。**刘家男**,博士,副研究员,主要研究领域为应用密码学与数据安全。**李 明**,博士,副教授,主要研究领域为区块链安全与数据安全。**翁 健**(通信作者),博士,教授,国家杰出青年科学基金入选者,主要研究领域为公钥密码学。E-mail.cryptweng@gmail.com。

However, implementing and verifying the effectiveness of machine unlearning presents a number of challenges. One critical issue is determining whether the specified data has indeed been removed from the trained model. Current verification methods often rely on the assumption of a baseline model, which is a version of the model that has never been trained on the data in question. Verification is then conducted by comparing the parameter distribution or the output distribution of the unlearned model with that of the baseline model. If the two distributions closely match, it is inferred that the data has been effectively forgotten. But this approach is vulnerable to several limitations. In scenarios where the model owner acts maliciously, they may forge the parameters or the output distribution of the unlearned model, creating the appearance that the data has been removed even when it has not. Furthermore, tracing model parameters back to specific training data is inherently challenging, making it difficult for verifiers to definitively confirm whether the target model has truly forgotten the data in question. These challenges underscore the need for more robust verification mechanisms. To address these limitations, this paper introduces a new publicly verifiable machine unlearning scheme that leverages cryptographic tools. This scheme is designed to enhance transparency and accountability between the data owner (who requests data removal) and the model owner (who is responsible for executing the unlearning process). In this scheme, the data owner employs a dynamic universal accumulator to authenticate authorized data or to exclude unauthorized data. The model owner generates a proof under a publicly verifiable covert security model, demonstrating that the training process has either included only the accumulated data or excluded the unaccumulated data. The data owner then verifies the validity of the proof. If the data owner discovers that the model owner has used unauthorized data, they can generate undeniable and publicly verifiable evidence to hold the model owner accountable. This evidence can be reviewed by any third party, ensuring that the process is transparent and that malicious actions are deterred. To evaluate the effectiveness of the proposed scheme, testing experiments were conducted. These experiments assessed the computational overhead associated with proof generation and verification for varying data volumes. They also examined the impact of removing specific data points on the model's prediction results. The findings demonstrate the scheme's practicality and its potential to address key challenges in verifying machine unlearning. By integrating cryptographic tools and machine unlearning, this approach represents a significant advancement in ensuring compliance with data privacy rights in the context of machine learning.

Keywords machine learning; data security; the right to be forgotten; machine unlearning; verifiability

1 引 言

基于人工智能(Artificial Intelligence, AI)的应用服务正改变着人们生活的方方面面,例如智能助手、聊天机器人、自动驾驶、人脸识别、智能家居等。据工信部统计数据,2023年我国人工智能核心产业规模已达到5000亿元[1]。放眼全球,许多产业界巨头如OpenAI、谷歌、华为等均提供了AI应用服务。在此类服务中,云服务提供商可以在数据主体(也称数据拥有方)同意的情况下收集数据,并使用所收集的数据训练机器学习模型,完成训练后通过公开可访问接口来提供模型的预测服务。然而,研究人员发现训练数据的相关隐私信息可能会在模型预测服务的过程中被泄露。这是因为训练数据的抽象特征会在训练过程中形成模型的参数权重,从而"记忆"了训练数据,并在模型预测服务的输入输出中泄漏训练数据的相关隐私信息[2-6]。因此,数据拥有方也许会顾虑个人隐私信息泄露的隐患,从而请求从云服务提供商的模型中撤回或遗忘其个人数据。除了隐私泄漏问题,数字版权侵权也可能导致模型被销毁或导致云服务提供商面临巨额罚款。譬如,《纽约

时报》于2023年12月起诉OpenAI未经授权使用其文章训练大模型，并要求销毁所有包含《纽约时报》文字作品的生成式人工智能模型或其他大语言模型和训练数据集。

面对数据主体个人信息的失控，全球许多国家出台了有关个人信息删除义务的法律法规，使得数据主体的遗忘请求获得法律保护。这最早可以追溯到被遗忘权（the Right to Be Forgotten）。该概念最开始出现于2010年法国的一项立法案中，之后在2016年欧盟通过的《通用数据保护条例》（GDPR）第17条被确认了法定地位。被遗忘权强调了数据主体对其个人数据充分的控制权，个人具有要求删除已过时或者撤销已授权的个人数据权利。在我国，数据安全相关法律法规文件均规定了公民有权要求信息处理者删除其个人信息的请求。随着AI应用所使用的数据量和范围日益扩增，个人信息卷入AI应用的负面影响逐渐暴露出来，AI领域的数据删除正确性也逐渐引起世界各监管机构的重视[7]。譬如，英国信息专员办公室（ICO）要求重新训练AI模型或完全删除AI模型来满足数据删除正确性。

在上述背景下，模型遗忘①（Machine Unlearning）的概念[8]被提出，意指从目标模型中删除其训练数据集的某些数据及其作用对目标模型参数的影响。根据数据遗忘方式的不同，实现模型遗忘的策略可以被划分为两种。一种是算法层面的遗忘策略[9-14]，将要删除的数据从目标模型的训练数据集中移除，并用剩余数据重新训练模型，从而得到一个遗忘模型（Unlearnt Model）。另一种是参数层面的遗忘策略[15-18]，通过直接修改目标模型的参数，直到修改后的模型在参数或输出分布上与完全重新训练后的模型大致相同。

上述算法层面和参数层面的遗忘策略关注于如何高效地遗忘数据，而并不考虑云服务提供商是否会诚实地执行模型遗忘的过程。反观实际服务场景，云服务提供商从目标模型遗忘数据不仅需要消耗计算和时间资源，还可能面临应用服务质量降低的风险，所以云服务提供商与数据主体删除数据的动机存在明显利益冲突，以至于极有可能拒绝回应数据主体的遗忘请求，或者伪造模型遗忘来免受罚款。最近成功发起的参数伪造攻击[19]可能会进一步助长云服务提供商的欺骗动机。具体地，云服务提供商可以使用与原训练数据完全不重叠的数据集伪造一个模型，使其参数分布与目标模型参数分布不可区分，并利用模型分叉攻击将伪造模型作为目标模型来回应数据主体的请求，从而达到无需对目标模型执行模型遗忘操作的目的。因此，考虑到不诚实云服务提供商潜在的欺骗行为，模型遗忘领域亟须提供额外的模型遗忘效果验证技术，使得数据主体或监管机构得以验证云服务提供商是否真正执行了模型遗忘的操作。

虽然目前存在一些模型遗忘验证技术[20-25]，但是这些技术在参数伪造攻击和模型分叉场景下不适用。这主要源于以下几点因素：第一，现有验证指标缺乏说服力。目前模型遗忘策略通常与基于黑盒或白盒探测的验证指标配合执行，具体通过对比遗忘后模型与重新训练后模型的输出分布或参数权重分布来验证遗忘的效果。但是，在参数伪造攻击下，无论是黑盒还是白盒探测的方式，都无法保证比较时所使用的是真实遗忘后的模型还是伪造的模型。第二，现有模型训练证明方法[26]不能直接拓展得到模型遗忘证明方法。模型训练证明方法通过生成含有训练过程一系列中间模型参数的日志来证明模型训练的真实性，但由于中间模型参数可以被伪造，以至于不能推导获得模型遗忘的证明。Zhang等人[27]和Thudi等人[19]的研究已经提供了相关有力的证据，具体展示了预期的模型参数或其分布可利用特殊设计的训练数据来获得。

鉴于现有研究的不足，本文将利用密码学领域的可验证计算（Verifiable Computing）技术[28]，结合算法层面的遗忘策略，提出一种公开可验证方案，允许数据主体和监管机构对模型遗忘效果的真实性进行验证，以推进实现AI领域数据删除正确性。其设计思路源于基于密码学的可验证计算领域“先认证再证明（Authenticate First and Prove Later）”的一系列工作[29-38]。最直观的想法是让数据拥有方使用认证数据结构[39]（Authenticated Data Structure，ADS）对其数据进行可公开的认证，并将数据和ADS发送给云服务提供商；接着云服务提供商提供零知识证明（Zero-Knowledge Proof，ZKP），以证明目标模型确实使用了与ADS完全一致的数据来完成训练；当发起遗忘数据请求后，数据拥有方从可公开的ADS动态删除数据，并验证删除数据的非成员证明；另外验证目标模型执行模型遗忘所涉及计算的正确性，以及该计算的输入数据与更新后的ADS一致；若数据拥有方发现上述数据使用不一致，可以生成公开可验证的凭证来追责云服务提供商的不正确行为。本文具体采用了Boneh等人的动态通用RSA累加器[40]来公开认证训练数据，同时支持可证明的动态

添加和删除，所生成的累加值用于指引底层计算应使用公开认证过的训练数据作为输入；接着，整合Helen[41]的“先承诺再证明”（Commit-and-Prove，CaP）工具和一类内存可延展的交互式ZKP协议[31]来证明被累加数据和底层计算输入数据的一致性。

此外，本文设计公开可验证隐蔽安全（Publicly-Verifiable Covert Security，PVC security）模型下的ZKP协议，应用PVC安全模型的动机在于，云服务提供商可能更关心因作弊被抓而导致法律或声誉损失，而不是成功作弊所获得的利益。现实案例如，2022年初法国数据隐私监管机构CNIL发现谷歌滥用Cookies，对其处以1.5亿欧元的巨额罚款，并要求在三个月内完成整改，否则继续处以每天10万欧元的罚款。因此，提供模型遗忘的公开可验证性具有现实应用意义。

本文的主要贡献总结如下：

（1）给出了公开可验证模型遗忘问题的定义，在已有工作只考虑诚实云服务提供商场景下的数据删除正确性不可区分性定义基础之上，提供了一个公开可验证的功能函数，允许数据拥有方验证模型遗忘的正确性，或者以一定概率抓获云服务提供商不正确的行为，并向第三方提供公开可验证的证明。

（2）提出了基于动态通用RSA累加器、交互式ZKP协议和公开可验证隐蔽安全模型的公开可验证模型遗忘方案，这是作为上述公开可验证模型遗忘问题定义的一个具体实例。

（3）设计了模型遗忘模拟实验，评估了不同数据规模情况下证明和验证的计算开销，同时评估了不同数据点删除对模型预测结果的影响。

本文第2节介绍了模型遗忘方法和模型遗忘验证方法的相关工作；第3节介绍了方案设计所需的预备知识，涉及模型遗忘定义、承诺、累加器、零知识证明、公开可验证的隐蔽安全等密码知识；第4节陈述方案设计的挑战和解决思路；第5节和第6节分别介绍公开可验证模型遗忘的问题定义和具体设计；第7节和第8节分别分析了安全性和展示了相关实验结果；最后，本文在第9节进行总结与展望。

2 相关工作

2.1 模型遗忘方法

模型遗忘的一种最简单实现方式是从训练数据集移除指定遗忘的数据，并使用剩余数据重新训练一个机器学习模型以替换原来的机器学习模型。显然重新训练的方式，尤其对于参数量大的模型来说，开销太大。因此，近年来许多学者提出相比于重新训练更快的近似模型遗忘（Approximate Unlearning）方法[15-18]，例如，应用差分隐私技术来隐藏某个待删除数据点留在模型参数的影响，最后通过对比遗忘后模型与重新训练模型的参数或输出分布来检验所提方法的有效性。另一类被称为确切模型遗忘（Exact Unlearning）的方法[9-14]，从训练阶段开始就对训练数据进行分组，并且每组分别训练一个子模型，当接收预测数据时，通过聚合这多个子模型的预测输出作为最终预测输出。这种方式由于将某组数据限定在某一个子模型，当需要遗忘该组数据时，只需重新训练子模型而非整个模型，所以提高了模型遗忘效率。

2.2 模型遗忘验证方法

当前少量工作[20-25]关注模型遗忘效果的可验证方法，这些方法通常利用成员推断（Membership Inference）或者后门嵌入（Backdoor Insertion）技术来让验证方通过黑盒询问模型的方式来探测某些指定删除的训练数据是否还作用在遗忘后机器学习模型参数上，从而验证数据是否被遗忘。具体来说，基于成员推断的验证方法让验证方使用一系列与训练数据相似的测试数据样本来询问一个遗忘机器学习模型，并获得一系列模型输出，从而来审计该模型是否已经遗忘了某些训练数据。正如Sommer等人[21]所述，这类基于成员推断的方法存在验证准确性较低等缺陷，因此他们提出了一种基于后门嵌入的方法，让验证方先将后门嵌入在模型的训练数据集中，之后用嵌入了同样后门的测试数据询问模型，通过观察模型输出来判断该模型是否删除了指定部分训练数据。不同于上述针对中心化机器学习场景的工作，Gao等人[22]针对联邦学习场景提出一种通用的数据遗忘验证方法，该方法对多种基于参数拟合的模型遗忘都有效。本文与现有工作相比，考虑了更强的敌手场景，敌手可能发起模型分叉攻击或者模型伪造攻击来替换被询问的模型，前一种攻击可以分叉一个不使用指定训练数据集训练后的模型，后一种攻击可以伪造一个与使用了指定训练数据集训练后的模型参数相同分布的模型，这些攻击可能使得现有基于黑盒询问模型来探测数据遗忘效果的方式无效。此外，现有方法需要验证方承担较高的计算开销，如在训练数据嵌入后门或训练影子模型（Shadow Model），这可能不适用于轻量级客户端作为验证方的场景。

类似本文的工作，如Weng等人[42]和Eisenhofer

等人[43]的方案，也考虑了强敌手场景，但是它们与本文要么在安全假设上不同，要么在模型遗忘定义上不同。他们都先形式化定义了可验证模型遗忘问题，并分别采用了可信执行环境和简洁非交互式ZKP技术的实现方式，如表1所示。虽然Weng等人提供高效的可验证功能，但需假设信赖硬件厂商。Eisenhofer等人定义了训练阶段而非预测阶段的模型遗忘请求，不能保证预测阶段的模型是模型遗忘之后的模型，因而可能遭受前文提及的模型分叉攻击，与本文对模型遗忘的定义范围不同。此外，该工作不仅要求每一轮训练需生成证明，还要求验证方需验证每一轮证明的有效性，导致证明和验证开销大。

表1　基于可验证计算的模型遗忘验证方法对比

方法	技术路线	训练-预测全流程	验证开销
文献[42]的方法	可信执行环境	是	低
文献[43]的方法	非交互式ZKP	否	高
本文方法	交互式ZKP	是	较低

3　预备知识

本节先介绍模型遗忘的概念，然后介绍本文所应用的密码学术语，包括承诺（Commitment）、累加器（Accumulator）、零知识证明ZKP和公开可验证的隐蔽安全性。

3.1　模型遗忘定义

模型遗忘是一种从已训练的机器学习模型消除或遗忘其训练数据集中部分数据留在模型上影响的技术。目前学术界将这类技术的实现方式可以划分为三种：重新训练（Retraining）、近似模型遗忘和确切模型遗忘。由于重新训练的方式计算开销大，所以通常不被采用。下文的定义主要介绍后面两种实现方式。

定义1．模型遗忘．给定一个包含N个样本的训练数据集$D=\{d_i\}_{i\in\{1,2,\cdots,N\}\setminus u}\cup\{d_u\}$。$D_{-u}$代表从数据集$D$移除了样本$d_u$的训练数据集，包括$N-1$个样本。机器学习模型$M$是使用训练数据集$D$训练而成的模型。$\mathcal{D}_M$代表机器学习模型$M'$的模型参数分布，$M'$是通过一种模型遗忘算法A遗忘了样本$d_u$后更新的机器学习模型。$\mathcal{D}_{M''}$代表机器学习模型$M''$的模型参数分布，$M''$是使用数据集$D_{-u}$训练而成的模型。如果$\mathcal{D}_{M'}$近似相等于$\mathcal{D}_{M''}$，则称算法A是近似模型遗忘算法。如果$\mathcal{D}_{M'}$等于$\mathcal{D}_{M''}$，则称算法A是确切模型遗忘算法。

3.2　承　诺

一个非交互式承诺方案涉及一个概率多项式时间*Setup*算法，该算法接收输入安全参数1^k，输出一个承诺密钥ck。此承诺密钥ck关联一个消息空间$\mathcal{M}$，一个承诺空间$\mathcal{C}$，一个承诺打开空间$\mathcal{O}$和以下两个多项式时间算法：

- $Commit(ck,m,r)\to(com,d)$：接收承诺密钥ck，随机数r以及消息$m\in\mathcal{M}$，该算法输出一个承诺值$com\in\mathcal{C}$及其打开值$d=(m,r)\in\mathcal{O}$.
- $Open(ck,com,d)\to 1\text{ or }0$：接收承诺密钥$ck$，承诺值$com$及其打开值$d=(m,r)$，如果$d$是$com\in\mathcal{C}$的有效打开值，则输出1；否则，输出0.

3.3　累加器

累加器[44-46]能将一个集合的多个元素压缩成一个较短的累加值，并能为每一个被压缩元素提供该元素与累加值的成员关系证明。通用累加器则既能为被压缩元素生成成员关系证明，又能为没有被压缩的元素生成非成员关系证明。动态通用累加器则是指具备动态更新功能的通用累加器，具体来说，动态通用累加器能够动态地添加和删除一个元素，同时更新相应元素与累加值的成员关系证明。

本文回顾Boneh等人2019年提出的基于RSA假设实现的动态通用累加器[40]。为了便于叙述，使用$\mathbb{G}$代表某种未知阶群，$\xleftarrow{\$}$表示随机均匀选取，$g$表示$\mathbb{G}$的一个生成元。$\mathbb{Z}$代表素数域，$X$代表当前时刻的一个元素集合，$\mathrm{acc}\in\mathbb{G}$代表当前时刻的累加值。$H_{\mathrm{prime}}$定义为一个抗碰撞（Collision Resistant）、不可分割（Division Intractable）的哈希函数，将集合X的某个元素映射为$\mathbb{Z}$的一个素数。下文描述累加器的基本算法，如果在未知阶群$\mathbb{G}$中RSA假设成立，这些算法是安全的：

- $Setup(1^k)\to(\mathbb{G},g)$：通过$\mathbb{G}\xleftarrow{\$}Gen(1^k)$和$g\xleftarrow{\$}\mathbb{G}$生成公开参数$\mathbb{G},g$.
- $Add(\mathrm{acc},X,x)\to\mathrm{acc}$：接收输入acc，$X$和$x$，当$x\in X$时，输出当前时刻累加值acc；当$x\notin X$时，通过$X\cup\{x\}$将$x$添加到集合$X$中，并计算$\mathrm{acc}\leftarrow\mathrm{acc}^{H_{\mathrm{prime}}(x)}$，输出acc.
- $Del(\mathrm{acc},X,x)\to\mathrm{acc}$：接收输入acc，$X$和$x$，当$x\notin X$时，输出当前时刻的累加值acc；当$x\in X$时，通过$X\leftarrow X\setminus\{x\}$将$x$从集合$X$中移除，并计算$\mathrm{acc}\leftarrow\mathrm{acc}^{H_{\mathrm{prime}}(x)^{-1}}$，输出acc.
- $MemProofCreate(\mathrm{acc},X,x)\to\omega_x$：接收输入

acc，X和x，计算$\omega_x \leftarrow g^{\prod x' \in X \wedge x' \neq x H_{\text{prime}}(x')}$，输出$\omega_x$作为元素$x$的成员证明.

- *VerMemProof*(acc，ω_x，x)→1 or 0：接收输入acc，ω_x和x，如果$(\omega_x)^{H_{\text{prime}}(x)}$=acc，输出1；否则，输出0.
- *NonMemProofCreate*（acc，X，x）→u_x：接收输入acc，X和x，选择参数$a,b \in \mathbb{Z}$使得$\prod_{x' \in X} H_{\text{prime}}(x') + bH_{\text{prime}}(x) = 1$，输出$u_x \leftarrow \{a, g^b\}$作为元素$x$的非成员证明.
- *VerNonMemProof*(acc，u_x，x)→1 or 0：接收输入acc，u_x和x，如果$acc^a (gb)^{H_{\text{prime}}(x)} = g$，输出1；否则，输出0.

此外，令$h = H_{\text{prime}}(x)$，如果x是当前累加值acc的成员元素，那么记为$h \in$ acc，否则记为$h \notin$ acc。

3.4 零知识证明

零知识证明系统允许证明方在不泄露秘密值的情形下向验证方证明关于秘密值的论断真实性。通过将复杂问题归约至NP完全问题，通用零知识证明系统支持通用计算，存在不同的底层构造技术。目前应用于处理复杂计算（如本文涉及的机器学习模型训练过程）时，这类技术还存在证明内存开销大、计算和通信效率低等问题。本文关注一类证明高效的交互式零知识证明协议[31]，其基础构件是基于子域向量的不经意线性函数计算（Sub-Field Vector Oblivious Linear Evaluation，sVOLE）构造的消息认证码。定义素数域$\mathbb{F}_q$及其扩展域$\mathbb{F}_{q^k}$，下文介绍该基础构件$\mathcal{F}_{sVOLE}^{q,k}$在一个证明方和一个验证方交互下实现的基本功能：证明方输入秘密数据$x \in \mathbb{F}_q$（若无，$x \xleftarrow{\$} \mathbb{F}_q$），证明方获得秘密数据的认证码$y = (a \cdot x + b) \in \mathbb{F}_{q^k}$，同时验证方获得其中的全局密钥$a \in \mathbb{F}_{q^k}$和本地密钥$b \in \mathbb{F}_{q^k}$。上述基础构件在交互式零知识证明协议中可用于高效地生成认证数据$[x]$，此处的认证数据$[x]$具体指证明方持有的(x, y)和验证方持有的b。

具体来说，基于上述基础构件的交互式零知识证明包含两个阶段的协议：

（1）离线协议（Offline Protocol）：给定一个待证明的通用计算电路，证明方和验证方交互生成该电路相关的认证数据。

（2）在线协议（Online Protocol）：对于电路输入的所有证据（Witnesses）以及电路执行后每一个乘法门的输出值，证明方使用离线阶段的认证数据生成它们的秘密承诺值。为了验证电路执行的正确性，验证方与证明方交互并利用离线阶段的认证数据验证此阶段所有秘密承诺值的正确性。

3.5 公开可验证的隐蔽安全

公开可验证隐蔽安全的概念建立在隐蔽安全模型上，提供了公开可验证性[47-49]。隐蔽模型的安全性介于半诚实模型和恶意模型之间，该模型一定程度上平衡了安全性和效率，相比于半诚实模型安全性更高，同时相比于恶意模型执行效率更高。具体来说，隐蔽模型下的敌手会像恶意敌手一样偏离协议的执行，但同时诚实参与方具备一定概率识别出其偏离行为，这一概率称为威慑因子。隐蔽敌手通过理性地权衡损失和收益决定是否偏离协议。更进一步地，公开可验证的隐蔽模型增强了隐蔽模型的威慑能力，当诚实参与方识别出偏离协议的参与方时，诚实参与方能够利用公开可验证的凭证揭发偏离协议的参与方行为，从而使任意第三方验证凭证的有效性。该模型特别适用于参与方因害怕作弊被抓会导致信誉受损，并且在不可否认证据情况下公开受到审判的场景。

4 方案设计的挑战和解决思路

本文方案设计基于“先认证再证明”的思想，先由数据拥有方生成数据的认证值，再由云服务提供商提供关于其模型训练确实使用了与认证值完全一致的数据；当发起遗忘数据请求后，数据拥有方从认证值中删除数据，并验证云服务提供商模型训练的输入数据与更新后的认证值是否一致；若数据拥有方发现上述数据使用不一致，可以生成公开可验证的凭证来追责云服务提供商的不正确行为。具体地，使用D, M代表训练数据集和模型，$d_u \in D$是被遗忘数据，$f(\cdot)$、$f'(\cdot)$、$H(\cdot)$分别表示训练算法、模型推理算法和可公开的认证数据结构（如Pedersen承诺、Merkle树、RSA累加器等），a、a'指数据拥有方在时间$t, t'(t < t')$提供的测试数据样本，上述过程可表示为如下的组合论断：$M_t := f(D) \wedge h_t := H(D) \wedge p = f'(M_t, a)$和$M_{t'} := f(D \backslash d_u) \wedge h_{t'} := H(D \backslash d_u) \wedge p' = f'(M_{t'}, a')$。上述论断有三方面要求：

（1）$f(\cdot), f'(\cdot), H(\cdot)$执行正确；

（2）同一时间上的$f(\cdot)$和$H(\cdot)$使用了一致的输入数据，即D或$D \backslash d_u$；

（3）$f'(\cdot)$的输入模型是同一时间上$f(\cdot)$的输出模型。

目前基于密码学的可验证计算采用了两种技术

路线来证明上述论断(2)的要求：第一种是把公开ADS的验证过程表述为底层证明系统的一部分关系描述，然后与$f(\cdot)$和$f'(\cdot)$计算正确性证明的关系描述一起作为底层证明系统的输入，最后生成证明[29-31]。第二种是基于解耦的技术线路，采用独立于底层证明系统的机制来证明ADS所认证的数据与底层计算$f(\cdot)$之间输入数据的一致性[32-38]。

4.1 问题挑战

前述的技术路线不能直接用于高效地解决本文场景问题，主要源于模型遗忘场景具备以下两个场景特点。一方面，训练时需要认证更新的数据规模大。机器学习模型常常需要大规模训练数据来完成训练以保证模型准确性，而遗忘请求通常情况下只涉及一小部分数据点的遗忘，剩余用于更新模型的训练数据规模依然很大。这就要求方案设计需考虑能够高效地支持大规模数据的认证及认证更新。另一方面，训练时需要多次迭代重复计算且每次迭代的输入数据不确定。训练算法通常需要多次迭代的重复计算，根据随机梯度下降(Stochastic Gradient Descent，SGD)优化算法，每次迭代计算的实际输入是从训练数据集中随机采样的部分数据，因此输入数据是不确定的。考虑这种特点，不能直接使用已有技术路线来证明训练算法的实际输入与先前经过认证的数据是否一致。此外，云服务提供商常常使用黑盒预测服务方式，其重复计算的中间结果(如中间模型参数)对数据拥有方是秘密的，这也增加了证明难度。

基于上述特点，如果采用前面所述第一种技术路线，需将大规模认证数据的验证表达成关系描述，这显然会引起大量的证明生成开销。第二种技术路线要么只支持某种特定的计算[32-33]，要么其认证数据结构仅支持追加操作[34]，要么是针对简洁非交互式零知识证明系统(Zero-Knowledge Succinct Non-interactive Argument of Knowledge，zk-SNARK)的定制化构造[35-38]，而考虑到zk-SNARK证明系统的证明生成效率和内存扩展性能难以支持大量重复计算及数据量大的场景(见表2)，所以这类证明系统并非本文技术路线的首选。

为了实现高效公开可验证的模型遗忘方案，本文先考虑在执行算法层面模型遗忘的前提下降低遗忘成本，从而为进一步降低认证更新和计算证明开销提供一个好的起点。因此，从分组训练的思想[9]出发，将训练数据集分成多个不重叠的组，每个组训练一个模型，最后将这些模型的输出合并以提供最终的预测服务，使得每个数据点仅影响其所属组内的模型参数。同时，进一步将组内数据集划分为多个相等大小的数据片段，并增量地添加它们到组内模型训练中，且在添加新数据片段之前保存训练的中间模型，这使得每个数据片段对组内模型参数的影响有迹可循。基于这一起点，认证更新和计算证明可以分组实现，由于遗忘某一数据点只会影响其所属组，所以认证更新和计算证明开销相比于整体训练的开销低。

表2 Groth16 zk-SNARK方案[50]实现矩阵相乘开销例子

矩阵相乘维数	公共参考字符串存储开销	证明生成时间开销(秒)
5×5	18.85 KB	0.0135
10×10	121.70 KB	0.0548
20×20	854.62 KB	0.2893
30×30	2.84 MB	0.8683
50×50	12.15 MB	3.2480
100×100	94.95 MB	24.5670
200×200	750.49 MB	202.9710

然而，这一好的起点还不足以缓解问题挑战，原因有两个方面：首先，即便将常见简单的SVHN数据集划分为10组，每组依然有数万个数据片段，因而生成被遗忘数据不再认证数据结构的证明开销依然很大，更不用说其他复杂大规模的训练数据集。其次，组内训练依然有多次重复计算、输入数据不确定和中间结果不可见的特性。

4.2 解决思路

针对前述问题挑战，本文以分组训练作为起点，妥适地整合了Boneh等人的动态通用RSA累加器[40]、Helen的“先承诺再证明”工具[41]以及一类内存可延展的交互式ZKP协议[31]，提供了一个公开可验证的模型遗忘方案，并将其应用于一类线性机器学习模型训练与预测场景。之所以使用RSA累加器而非Merkle树作为可公开的认证数据结构是因为它删除数据前后的累加值是概率多项式时间不可区分的，满足下文第5.3小节定义二的要求。总的来说，本文方案具体通过三个方面来实现：

第一，解决由于SGD训练每次迭代输入数据具有随机性而导致每次迭代需证明输入数据一致性问题。本文先通过对输入数据进行矩阵乘法的预处理操作，将其打包成固定的输入组件，并生成承诺；然后，采用另一种优化算法对这一输入组件进行迭代训练，确保每次迭代的输入数据是固定的而非随机的。接着，通过以下方式生成证明：(1) 利用Helen的CaP工具证明了迭代计算的正确性，确保每次迭代都使用了承诺后的输入组件；(2) 应用ZKP协议

证明预处理时矩阵乘法的计算正确性；(3) 证明上述(2)输入数据与先前RSA累加器认证的数据是一致的。由于(3)涉及大规模训练数据而难以高效进行，以下两个方面围绕其高效实现展开。

第二个方面，本文证明(3)的方案设计需支持大规模输入数据，并且要同时兼容所采用ZKP协议[31]和RSA累加器[40]对各自输入数据的不同处理构造。具体地，本文ZKP协议的秘密认证输入是关于输入数据在素数扩展域上的消息认证码，而可公开认证的RSA累加器先将输入数据哈希映射到一个大素数域中，然后在一未知阶群内进行哈希值指数乘法运算。这两个部分对输入数据的处理差异使得证明它们的一致性并不容易。一种直接的想法是将对于他们一致性的验证编译成通用证明zk-SNARK的底层电路来证明，但在n个输入情况下其未知阶群内的操作将导致近$1.8\times n$百万个约束需要证明，考虑到本文场景很大，这种实现方式并不实际。因此，本文没有将一致性验证过程中所涉及的未知阶群内操作编译成底层电路来证明，而是让验证方(也是数据拥有方)先检查输入数据是否与证明方的秘密认证输入保持一致，然后由验证方生成对应输入数据在大素数域上的哈希值来询问该哈希值是否存在于可公开认证的RSA累加器。

最后，在上述一致性证明中进一步引入公开可验证的隐蔽安全模型。这受启发于Damgård等人的通用编译器[49]，以实现具有威慑因子的随机选择输入数据的一致性证明。借助PVC安全特性，本文实现由验证方在上述(2)中随机选择部分秘密认证输入，然后生成相应输入数据的哈希值来询问其是否存在于RSA累加器。一旦验证方发现云服务提供商使用了不一致输入数据的欺骗行为，能够提供不可否认的公开可验证凭证来追责它的欺骗行为。

5 公开可验证的模型遗忘问题定义

本节定义公开可验证的模型遗忘问题，涉及一个执行模型遗忘的云服务提供商和一个请求数据遗忘的数据拥有方。本文考虑云服务提供商是不诚实的，当数据拥有方请求遗忘数据时，该云服务提供商可能不遵循数据拥有方的请求来偏离模型遗忘的行为。因此，本文提供公开可验证的功能，允许数据拥有方验证模型遗忘的正确性，或者以一定概率抓获云服务提供商不正确的行为，并向第三方提供公开可验证的证明。下文将依次介绍公开可验证模型遗忘的基本流程定义、威胁假设、模型遗忘正确性定义、主要算法定义和安全目标。

为了方便阅读，表3给出了本节主要使用的符号含义。

表3 第5节的符号含义

符号	含义
$\mathcal{F}_{AP}$	“先认证再证明”的功能函数
ε	执行环境
(π,π_D,π_V)	交互协议
P_s	云服务提供商
P_d	数据删除请求者
k	认证数据时算法使用的安全参数
M、D	模型、被用于模型训练的数据
ρ	认证模型时算法使用的安全参数
H_M、H_D	模型摘要、数据摘要
d_u	被遗忘数据
$H_{D\backslash d_u}$	除去被遗忘数据的数据摘要
M'、D_r	更新后的模型、用于模型训练的剩余数据
H_{D_r}	剩余数据摘要
F	训练过程涉及的计算
G	预测过程涉及的计算
p	预测结果
pk, sk	云服务提供商的公钥和私钥
cert	用私钥sk生成的凭证
c	数据拥有方提出的挑战
$proof$	正确性证明
$\perp$	中止符号

5.1 基本流程定义

公开可验证模型遗忘的基本流程包含六个步骤，由云服务提供商和数据拥有方交互完成，如图1所示。

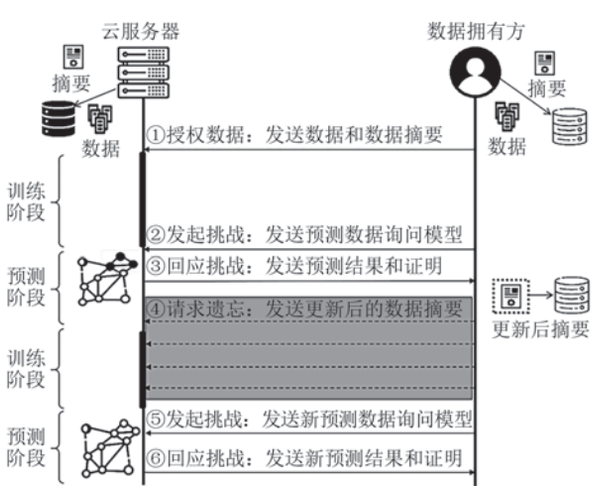

图1 基本流程

①授权数据：数据拥有方向云服务提供商授权使用其数据训练一个机器学习模型，通过发送数据

和关于数据的摘要（如哈希值、承诺值或累加值）来完成。云服务提供商接收并存储数据和摘要，然后使用数据训练模型，并部署模型提供预测服务（如API查询接口）。值得注意的是，数据拥有方也本地保存数据摘要，便于后续追踪数据。

②发起挑战：数据拥有方使用测试数据询问云服务提供商的预测服务，以挑战当前预测服务模型使用的训练数据集是否含有步骤①授权的数据。

③回应挑战：云服务提供商返回预测结果和证明，以响应数据拥有方的挑战。

④请求遗忘：数据拥有方请求云服务提供商从当前提供预测服务的模型遗忘部分步骤①授权的数据，该步骤通过更新本地保存的数据摘要，并将更新后的摘要发送给云服务提供商来完成。

⑤发起挑战：与步骤②类似，数据拥有方使用新的测试数据询问当前预测服务模型，以确认当前模型使用的训练数据集是否含有步骤④请求遗忘的数据。

⑥回应挑战：云服务提供商返回新的预测结果和证明，以响应数据拥有方步骤⑤的挑战。

上述基本流程隐含了一个条件，即数据遗忘请求需要建立在被遗忘数据已经被当前机器学习模型用于训练的前提之下。该流程不限定于某一种模型遗忘算法，对于第2.1节所介绍的多种遗忘学习算法是通用的，但在具体验证方法构造方面，后文第6节的方案针对基于分组思想的确切模型遗忘算法来设计。另外，本文与同领域文献[9,11,13-14]一样，考虑数据点层面的数据删除，并且为了便于定义问题，只考虑了遗忘一个数据点的情况。据Bourtoule等人[9]的文献中，只考虑了少量数据点被删除的情况，即总数据集大小的0.003%，并提到在实际中，遗忘请求的数量通常很少，约为总数据集大小的0.0001%。其他复杂情况，如涉及重叠数据特征或标签的数据遗忘、适应性地删除一系列数据点[10]等可建立在上述基本流程来进一步讨论和限定。

5.2 威胁假设

本文假设云服务提供商是不诚实的，可能采取任意手段偏离模型遗忘的过程，以欺骗数据拥有方遗忘了指定数据而事实上没有遗忘数据。具体地，本文考虑云服务提供商可能声称当前预测服务模型的训练数据集使用了（或没有使用）数据拥有方授权数据，但实际上没有使用（或已经使用）；云服务提供商也可能没有正确地执行模型遗忘过程；或者云服务提供商没有使用模型遗忘之后的模型提供后续的预测服务，而继续沿用之前的模型。另一方面，本文假设诚实的数据拥有方，其授权的数据不包含任何毒害数据（Poisoning Data）和后门（Backdoors）。关于数据遗忘，本文考虑目标模型曾经使用过请求遗忘的数据，且现在不再使用该数据。由于物理介质上的内存删除已被先前的工作所解决，因此本文不考虑这方面，本文也不考虑遗忘备份数据和模型。此外，文中假设云服务提供商与数据拥有方使用安全认证的通信通道。

5.3 模型遗忘正确性定义

本文的模型遗忘正确性定义先继承了Garg等人[51]所陈述的诚实场景下数据删除正确性的不可区分性定义，即当数据确实被删除时，任何计算区分器无法区分被删除数据曾存在并在之后被删除还是它从未存在过。然后，在云服务提供商引入一个实现“先认证再证明”的功能函数$\mathcal{F}_{AP}$，使得数据拥有方或第三方监管机构能够公开验证模型遗忘的正确性情况，同时任何计算区分器无法区分被删除数据是模型遗忘数据还是从未用于模型训练。

Garg等人工作[51]在通用可组合框架下定义了一个云服务提供商P_s、一个数据删除请求者P_d、除去P_s和P_d之外的执行环境$\mathcal{E}$、P_s和P_d之间使用数据d_u的交互协议π以及删除d_u的交互协议π_D。此外，定义P_s的状态*state*包含了内存数据以及由内存数据派生的数据。其中派生的数据在本文指训练数据和模型参数。Garg等人的数据删除正确性定义要求P_s的状态*state*和环境$\mathcal{E}$在现实世界执行的联合分布与在理想世界执行的联合分布是计算上不可区分的。这意味着在真正执行数据删除的情况下，任意计算区分器无法区分被删除数据是被P_s收集之后再被删除，还是从未被收集过。

在上述定义基础之上，本文引入“先认证再证明”的功能函数$\mathcal{F}_{AP}$，该功能函数在某一时间点上定义了两个阶段：认证阶段和证明阶段。认证阶段对训练数据集中的每一个数据生成承诺；证明阶段证明被认证过的训练数据被用于训练模型或未被认证的数据没被用于训练模型。其中认证阶段支持数据的成员和非成员证明查询，以说服验证者训练数据被授权使用或没有被授权使用；此外，也支持数据动态添加和删除，同时可高效地更新相应成员证明和非成员证明。本文在Garg等人定义下借助功能函数$\mathcal{F}_{AP}$来定义模型遗忘正确性。具体地，定义$\mathcal{F}_{AP}$－hybrid混合模型，所有参与实体（P_s和P_d）除

了之前的交互，还可以与可信的功能函数$\mathcal{F}_{AP}$进行交互（如图2所示）。如果能够在$\mathcal{F}_{AP}$－hybrid混合模型中设计一个实现功能函数$\mathcal{F}_{AP}$的协议Prot_{AP}，则P_s的状态*state*和环境$\mathcal{E}$在$\mathcal{F}_{AP}$－hybrid混合模型执行的联合分布与在理想模型执行的联合分布是计算上不可区分的，见定义2。

定义2. 模型遗忘正确性．定义云服务提供商P_s、数据删除请求者P_d、环境$\mathcal{E}$及P_s和P_d之间的交互协议(π,π_D,π_V)，其中π_V表示验证数据遗忘的交互协议，定义理想模型执行$\mathrm{IDEAL}^{\mathrm{P}_s,\mathcal{F}_{AP}}_{\mathrm{P}_d,\mathcal{E}}$的联合分布$(state^{ideal,k}_{\mathrm{P}_s,\mathcal{F}_{AP}},\ view^{ideal,k}_{\mathcal{E}})$和$\mathcal{F}_{AP}$－hybrid混合模型执行的联合分布$(state^{hybrid,k}_{\mathrm{P}_s,\mathrm{Prot}_{AP}},\ view^{hybrid,k}_{\mathcal{E}})$，如果对于安全参数$k\in\mathbb{N}$、概率多项式时间的$\mathcal{E}$和$\mathrm{P}_d$，任意计算区分器以可忽略的概率区分上述两个联合分布，则云服务提供商P_s以及P_s与P_d的交互协议过程(π,π_D,π_V)满足模型遗忘正确性定义，表示为

$$\begin{aligned}&\left|\Pr\left[\mathcal{D}\left(state^{hybrid,k}_{\mathrm{P}_s,\mathrm{Prot}_{AP}},\ view^{hybrid,k}_{\mathcal{E}}\right)=1\right]-\right.\\&\left.\Pr\left[\mathcal{D}\left(state^{ideal,k}_{\mathrm{P}_s,\mathcal{F}_{AP}},\ view^{ideal,k}_{\mathcal{E}}\right)=1\right]\right|\leqslant\\&negl(k).\end{aligned}\tag{1}$$

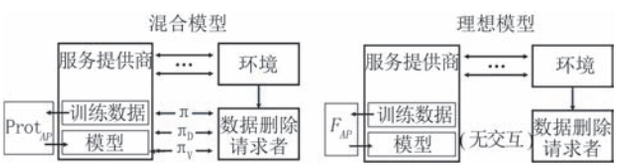

图2 混合模型执行与理想模型执行

5.4 主要算法定义

此处定义由云服务提供商和数据拥有方执行的一组算法（*AuthData*, *DelAuthData*, *AuthModel*, *TrackData*, *Judge*, *Verify*），以用于实现上一小节的功能函数$\mathcal{F}_{AP}$。在介绍之前，定义云服务提供商的公钥*pk*和私钥*sk*。

（1）$AuthData(k,D:=\{d_1,\cdots,d_n\})\to H_D$：数据拥有方执行该算法，接收安全参数$k$和数据$D:=\{d_1,d_2,\cdots,d_n\}$，输出可公开认证的数据摘要$H_D=\{h_1,h_2,\cdots,h_n\}$.

（2）$DelAuthData(H_D,d_u)\to H_{D\backslash d_u}$：数据拥有方执行该算法，接收数据摘要$H_D$和被遗忘数据$d_u$，输出更新后的数据摘要$H_{D\backslash d_u}=H_D\backslash\{h_u\}$.

（3）$AuthModel(\rho,M)\to H_M$：云服务提供商执行该算法，接收模型M和安全参数ρ，输出认证后的模型摘要H_M.

（4）$TrackData\left(c,D_r,H_{D_r},M,H_M,F/\!\!\wedge G\right)\to$ cert or $(p,M',H_{M'},proof)$：云服务提供商接收数据拥有方的挑战c（包含了预测数据）、输入数据D_r及其摘要H_{D_r}、输入模型M及其摘要H_M，执行计算$F/\!\!\wedge G$，F表示训练计算，G表示预测计算，获得预测结果p；验证方检查云服务提供商使用数据的正确性，若发现欺骗行为，则输出不可否认凭证cert，并中止；否则，输出预测结果p、更新后模型M'及其$H_{M'}$，以及正确性证明*proof*.

（5）$Judge(pk,\mathrm{cert})\to pk$ or $\perp$：第三方审计方使用*pk*验证cert的有效性，若有效，输出*pk*；否则，输出$\perp$.

（6）$Verify(proof,c,p,H_{D_r},H_M,H_{M'})\to 0$ or 1：若*Judge*算法被调用且输出*pk*，该算法不被调用；否则，数据拥有方验证*proof*的有效性，以验证预测p由摘要$H_{M'}$的原像M'通过预测计算G生成，且摘要$H_{M'}$的原像M'由数据D_r和模型M通过计算F生成，若验证无效，则输出0；否则，输出1。

5.5 安全目标

本文的安全目标关注于实现公开可验证的模型遗忘。这一目标在于保证训练数据按照数据拥有方的意愿被使用。验证方（即数据拥有方或数据删除请求方）能够验证其授权或者未授权数据是否被用于训练当前预测模型。当验证方发现证明方的不正确行为时，可生成公开可验证且不可否认的凭证，向第三方证明证明方确实执行了不正确行为。

6 公开可验证的模型遗忘具体设计

本文利用Weng等人的内存可延展交互式ZKP协议[31]、Boneh等人的公开可认证RSA累加器[40]以及Damgård等人的公开可验证隐蔽安全通用编译器[49]来设计具体的公开可验证模型遗忘方案。方案设计中面临的主要挑战是如何高效地证明RSA累加器所认证的数据与ZKP协议的实际输入数据是一致的。挑战源于既要支持大规模且动态删除的训练数据集，又要兼容ZKP协议和RSA累加器处理输入数据的不同构造（分别是素数扩展域上的消息认证码和哈希值的指数乘）。本文通过两个步骤解决这一挑战。步骤一，由证明方证明所使用ZKP协议的秘密认证数据与指定输入数据的一致性，之后由验证方（也是数据拥有方）生成这些输入数据的哈希值，并询问累加器，验证其是否被累加。步骤二，在公开可验证隐蔽（PVC）安全模型下执行上述

ZKP协议。利用PVC安全的优势，一旦数据拥有方发现RSA累加器所认证的数据和ZKP协议实际使用的数据不一致，能够生成一个公开可验证的凭证追责云服务提供商。

表4是本节所使用的主要符号含义。

表4　第6节的符号含义

符号	含　义
$g(\cdot)$	将训练数据打包成固定输入组件的计算
$\mathbf{b}$、$\mathbf{A}$、$\mathbf{B}$	$g(\cdot)$的输出
c_b、c_A、c_B	$\mathbf{b}$、$\mathbf{A}$、$\mathbf{B}$的承诺
$\mathcal{P}$	证明方
$\mathcal{V}$	验证方
$(pk_{\mathcal{P}}, sk_{\mathcal{P}})$	证明方的公钥和私钥
$(pk_{\mathcal{V}}, sk_{\mathcal{V}})$	验证方的公钥和私钥
$\mathcal{C}$	待证明的电路
I_w	电路$\mathcal{C}$的秘密输入个数
I_c	电路$\mathcal{C}$的乘法门数量
$3\mathcal{L}$	认证三元组个数
$\mathcal{S}_{\mathrm{OT}}$	不经意传输协议的执行过程
h_i	累加器中数据d_i的哈希值
Π_{OFF}	离线协议
Π_{ON}	在线协议
$\{[\lambda_i]\}_{i\in[\mathrm{I}_w]}$	离线协议生成，用于在线协议计算秘密输入的认证数据
$\sigma_i^{\mathrm{OFF},\mathcal{P}}$	证明方离线阶段生成的签名
$\sigma_i^{\mathrm{OFF},\mathcal{V}}$	验证方离线阶段生成的签名
$\sigma_i^{\mathrm{ON},\mathcal{P}}$	证明方在线阶段生成的签名
$\sigma_i^{\mathrm{ON},\mathcal{V}}$	验证方在线阶段生成的签名

6.1　构造组件

本文方案设计主要包含4个构造组件，如图3所示。首先是①支持成员关系查询的RSA累加器，数据拥有方使用RSA累加器生成训练数据集的累加值，被累加的数据意味着数据被授权用于模型训练。其次是②证明预处理步骤正确执行的ZKP协议，这将于下一小节介绍。接着是③由Helen借鉴而来的方法，利用“先承诺再证明”的工具来证明多次迭代训练和预测阶段的计算按预期正确执行。最后是④证明RSA累加器所认证的数据与ZKP协议的实际输入数据是一致的。

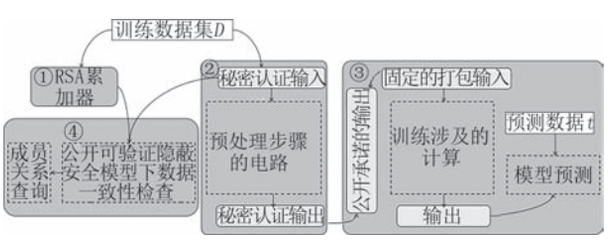

图3　构造组件概览

由于本文方案主要适用于处理线性计算，因此适用于类似LASSO的正则化线性模型，这类模型已经被广泛应用于风险分析、癌症诊断等实际场景。神经网络模型的训练过程涉及求导、非线性计算等复杂操作，而且训练迭代次数更高，本文方案不适用。如果存在能高效支持神经网络模型训练的零知识证明协议，可以在本文第5节的定义基础之上设计适用于神经网络模型的公开可验证模型遗忘方案。

6.2　预处理步骤

此处介绍如何将训练数据打包成固定的输入组件，使得训练过程中每次迭代的训练数据都是相同的，从而训练期间只需检查一次数据一致关系，而非每次迭代都检查。一开始，为了降低认证更新和计算证明开销，本文选择分组训练作为起点，正如前文所述，每组训练时需多次迭代计算，由于SGD的随机性质，每次迭代所选取的训练数据都是从训练数据集中随机选择的。这种随机性不利于6.1节模块④的高效实现，因为这导致每次迭代都要检查数据是否一致。为了解决这一问题，将SGD优化算法替换为交替方向乘子法（Alternating Direction Method of Multipliers，ADMM），这使得能够通过简单无需迭代的计算（记为$g(\cdot)$）来打包训练数据，之后的迭代训练过程都使用打包后数据。通过这一处理，数据一致关系检查变成检查函数$g(\cdot)$的输入数据是否与RSA累加器中的数据一致。

接下来介绍计算函数$g(\cdot)$主要涉及的矩阵相乘操作。$D:=\{d_1, d_2, \cdots, d_m\}$用于表示一组训练数据集，其中$m$表示该组数据点的最大索引，每个数据点包含$k$个特征。$m\times k$维数矩阵$\boldsymbol{X}$和$m$维向量$\boldsymbol{y}$分别表示这组数据点特征和其相应标签，它们将作为函数$g(\cdot)$的输入，其中$m>k$。具体计算函数$g(\cdot)$涉及计算$\boldsymbol{b}=\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$，$\boldsymbol{A}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\rho\boldsymbol{I}\right)$，$\boldsymbol{B}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\rho\boldsymbol{I}\right)^{-1}$，其中$\rho$是可设置的常量参数，$\boldsymbol{I}$是维数为$k$的单位矩阵，$\boldsymbol{b}$、$\boldsymbol{A}$及$\boldsymbol{B}$是$g(\cdot)$的输出。

6.3　公开可验证的证明协议

现在介绍证明预处理步骤正确执行和数据一致性的ZKP协议（图3中的②和④），该协议在公开可验证的隐蔽安全模型下运行，记为$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$。$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$不仅要求证明方$\mathcal{P}$证明$g(\cdot)$的正确执行，还需证明随机选中的秘密认证输入（即输入数据的消息认证码）与相应累加器中累加数据的一致性。验证方$\mathcal{V}$

具有一定概率发现证明方$\mathcal{P}$的不正确行为。若验证方$\mathcal{V}$发现不一致，则生成公开可验证的凭证，用于追责证明方$\mathcal{P}$的不正确行为。

从技术角度来看，$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$主要结合了Weng等人的交互式ZKP协议[31]和Damgård等人的公开可验证隐蔽安全通用编译器[49]来构造。所涉及的ZKP协议遵循恶意安全模型下的安全两方计算范式，包含离线协议和在线协议，只不过考虑只有证明方拥有计算时的秘密输入。在此基础上，根据编译器的定义，本文在离线阶段引入不经意传输(Oblivious Transfer，OT)协议和签名方案，目的在于实现验证方安全地随机抽取一部分输入数据，以便用于之后检查所选输入数据与RSA累加器中哈希值之间的一致性，并且在检查出不一致时，能够生成不可否认的公开可验证凭证。结合修改后的离线协议和原来的在线协议，本文获得公开可验证的隐蔽安全模型下运行的协议$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$。具体而言，OT协议在证明方和验证方之间执行，以获取验证方希望检查的随机输入。由于本文考虑的是诚实验证方，因此不要求验证方在OT协议中提前生成其随机输入的承诺。引入OT协议可以防止证明方在其不期望的输入被检查时进行恶意中止。此外，要求证明方使用签名方案对其发送的所有消息进行签名，以作为作弊证据被公开并由任何第三方验证。

为了便于理解协议，先介绍相关符号定义。$(pk_{\mathcal{P}}, sk_{\mathcal{P}})$，$(pk_{\mathcal{V}}, sk_{\mathcal{V}})$分别代表证明方的公钥和私钥；$\mathcal{C}$表示待证明的电路，$I_w$表示秘密输入的个数，$I_c$表示乘法门数量，$3l$表示认证三元组个数；$D:=\{d_1, d_2, \cdots, d_m\}$是证明方随机置乱了索引的秘密输入数据，将作为电路的输入，满足关系$\mathcal{C}(D)=1$。累加器中数据哈希值表示为$h_i=H_{\mathrm{prime}}(d_i)$。$\mathcal{S}_{\mathrm{OT}}$代表OT协议的执行过程。

$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$协议包含离线协议$\Pi_{\mathrm{OFF}}$、在线协议$\Pi_{\mathrm{ON}}$和*Judge*算法：

(1)在离线协议Π_{OFF}阶段，证明方$\mathcal{P}$和验证方$\mathcal{V}$调用第2.4节介绍的$\mathcal{F}_{\mathrm{sVOLE}}^{\mathbb{q},\mathrm{k}}$，生成$I_w+I_c+3\ell$个认证数据，包含$I_w$个$\{[\lambda_i]\}_{i\in[I_w]}$用于在线协议计算秘密输入、$I_c$个用于乘法门计算以及$\mathcal{L}$个乘法三元组；证明方$\mathcal{P}$和验证方$\mathcal{V}$执行OT协议$\prod_{\mathrm{OT}}^{(I_w,\left\lceil\frac{I_w}{2}\right\rceil-1)}$，其中证明方输入索引$\{1, 2, \cdots, I_w\}$，验证方获得随机索引$\{\hat{i}\}_{i\in[I_v]}$，$I_v=\left\lceil\frac{I_w}{2}\right\rceil-1$；证明方对所生成的认证数据$\{[\lambda_i]\}_{i\in[I_w]}$和执行流程$\mathcal{S}_{\mathrm{OT}}$生成签名$\{\sigma_i^{\mathrm{OFF},\mathcal{P}}\leftarrow Sign_{\mathrm{sk}_{\mathcal{P}}}(\mathcal{S}_{\mathrm{OT}}, [\lambda_i])\}_{i\in[I_w]}$，验证方根据索引$\{\hat{i}\}_{\hat{i}\in[I_v]}$对认证数据和执行流程生成签名$\{\sigma_{\hat{i}}^{\mathrm{OFF},\mathcal{V}}\leftarrow Sign_{\mathrm{sk}_{\mathcal{V}}}(\mathcal{S}_{\mathrm{OT}}, [\lambda_{\hat{i}}])\}_{\hat{i}\in[I_v]}$。

(2)在线协议Π_{ON}阶段，证明方$\mathcal{P}$和验证方$\mathcal{V}$使用上一阶段的$\{[\lambda_i]\}_{i\in[I_w]}$生成证明方的秘密输入数据$d_i$的认证数据，记为$\{[d_i]\}_{i\in[I_w]}$，在这之前证明方随机置乱秘密输入数据的索引；基于秘密输入数据，证明方执行电路$\mathcal{C}$，最后输出认证结果以供验证方验证；此外，验证方和证明方交互使用上一阶段的乘法三元组验证乘法计算的正确性；与此同时，证明方对秘密输入认证数据$\{[d_i]\}_{i\in[I_w]}$以及相应秘密数据的哈希值$\{h_i=H_{\mathrm{prime}}(d_i)\}_{i\in[I_w]}$生成签名$\{\sigma_i^{ON,\mathcal{P}}\leftarrow Sign_{sk_{\mathcal{P}}}(\mathcal{S}_{OT}, h_i, [d_i])\}_{i\in[I_w]}$，验证方生成签名$\{\sigma_{\hat{i}}^{ON,\mathcal{V}}\leftarrow Sign_{sk_{\mathcal{V}}}(\mathcal{S}_{OT}, h_{\hat{i}}, [d_{\hat{i}}])\}_{\hat{i}\in[I_v]}$。

(3)最后，*Judge*算法接收$\mathrm{cert}:=(\sigma_i^{OFF,\mathcal{P}}, \sigma_{\hat{i}}^{OFF,\mathcal{V}}, \sigma_i^{ON,\mathcal{P}}, \sigma_{\hat{i}}^{ON,\mathcal{V}}, \mathcal{S}_{OT}, [\lambda_{\hat{i}}], h_{\hat{i}}, [d_{\hat{i}}])$，以及秘密输入数据的认证数据$[d_{\hat{i}}]$的打开值$d_{\hat{i}}$，其中$\hat{i}\in[\mathrm{I}_v]$，验证签名的有效性，若任一签名无效，则中止；否则使用$h_{\hat{i}}$询问其是否是当前累加值acc的成员元素，若非acc认证授权的数据，则输出$pk_{\mathcal{P}}$，表明证明方$\mathcal{P}$使用了未授权数据，被抓获的概率为$\frac{1}{2}-\frac{1}{I_w}$。

6.4 整体构造

基于前面介绍的预处理步骤和$\prod_{\mathrm{PVC}}^{\mathrm{ZKP}}$协议，现在结合RSA累加器以及Helen的“先承诺再证明”工具实现3.3节定义的算法(*AuthData*, *DelAuthData*, *AuthModel*, *TrackData*, *Judge*)，给出如图4所示的整体构造，从而实现公开可验证的模型遗忘。整体构造包括(a)用于压缩训练数据，支持公开查询的认证数据结构，由*AuthData*和*DelAuthData*算法实现；(b)训练数据被正确预处理为$\boldsymbol{b}$、A和B的公开可验证证明，由*TrackData*算法实现；(c)基于“先承诺再验

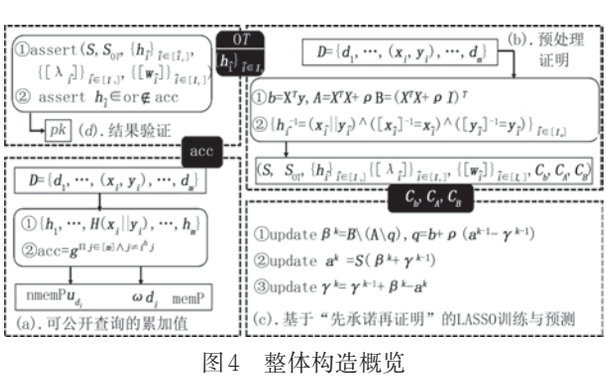

图4 整体构造概览

证”工具证明LASSO确实使用了c_b、c_A和c_B进行训练，并用训练后模型进行预测，由*TrackData*算法实现；(d)结果验证以检查输入数据一致性。

(1)$AuthData(k,D)\rightarrow$acc：数据拥有方调用累加器的*Setup*算法，生成参数$\mathbb{G},g$，然后将数据集$D:=\{d_1,\cdots,(\boldsymbol{x}_i,y_i),\cdots,d_m\}$压缩成累加值$\text{acc}\leftarrow g^{\Pi_{i\in\{1,2,\cdots,m\}}H_{\text{prime}}(d_i)}$，随后将$D$和acc发送给服务端。

(2)$DelAuthData(\text{acc},d_u)\rightarrow\text{acc}'$：数据拥有方调用累加器的*Del*算法，接收当前累加值acc和被遗忘数据$d_u=(\boldsymbol{x}_u,y_u)$，执行$\text{acc}'\leftarrow\text{acc}^{H_{\text{prime}}(d_u)^{-1}}$，输出更新后的累加值$\text{acc}'$。

(3)$AuthModel(\rho,M)\rightarrow com_M$：云服务提供商调用承诺*Setup*算法和*Commit*算法，接收模型M和安全参数ρ，输出承诺后的模型com_M。

(4)$TrackData(c,D_r,\text{acc}',M,com_M,F/\!/G)\rightarrow\text{cert or}(p,M',com_{M'},proof)$：云服务提供商和数据拥有方交互执行该算法，具体如下：

首先，数据拥有方发起随机性挑战$c:=(\mathcal{S}_{\text{OT}},t)$，其中$\mathcal{S}_{\text{OT}}$是第4.3节协议中的OT协议执行过程，隐含了采样秘密输入认证数据的随机数，t代表测试数据；

然后，云服务提供商输入数据集$D:=\{d_1,d_2,\cdots,d_m\}\backslash d_u$，执行4.2节和4.3节数据预处理过程$g(\cdot)$，输出$\boldsymbol{b}$、A以及B，$\prod_{\text{PVC}}^{\text{ZKP}}$的输出$\left(\sigma_i^{\text{OFF},\mathcal{P}},\sigma_i^{\text{OFF},\mathcal{V}}\sigma_i^{\text{ON},\mathcal{P}},\sigma_i^{\text{ON},\mathcal{V}},\mathcal{S}_{\text{OT}},\left[\lambda_{\hat{i}}\right],h_{\hat{i}},\left[d_{\hat{i}}\right]\right)$，其中签名$S:=\left(\sigma_i^{\text{OFF},\mathcal{P}},\sigma_{\hat{i}}^{\text{OFF},\mathcal{V}},\sigma_i^{\text{ON},\mathcal{P}},\sigma_{\hat{i}}^{\text{ON},\mathcal{V}}\right)$；

验证方验证$\left[\lambda_{\hat{i}}\right],h_{\hat{i}},\left[w_{\hat{i}}\right]$以检查云服务提供商使用数据的正确性，若发现欺骗行为，则输出不可否认凭证$\text{cert}:=(\sigma_i^{\text{OFF},\mathcal{P}},\sigma_{\hat{i}}^{\text{OFF},\mathcal{V}},\sigma_i^{\text{ON},\mathcal{P}},\sigma_{\hat{i}}^{\text{ON},\mathcal{V}},\mathcal{S}_{\text{OT}},\left[\lambda_{\hat{i}}\right],h_{\hat{i}},\left[d_{\hat{i}}\right])$，并中止；

否则，云服务提供商调用Helen的“先承诺再证明”工具和*AuthModel*算法，先对$\boldsymbol{b}$、A及B生成承诺c_b、c_A和c_B，并证明基于此承诺和初始模型M执行LASSO训练过程F(参见图4)，输出训练后的模型M'及其承诺$com_{M'}$和证明π_1；

云服务提供商接收挑战c中的测试数据t，再调用Helen的“先承诺再证明”工具和*AuthModel*算法，证明基于上述训练完成模型$com_{M'}$和t执行了预测过程G，输出预测结果p和证明π_2。

最后该算法输出$(p,M',com_{M'},proof:=(\pi_1,\pi_2))$。

(5)$Judge(pk_{\mathcal{P}},\text{cert})\rightarrow\perp\text{ or }pk_{\mathcal{P}}$：

首先，数据拥有方解析cert获得签名S，并验证签名S的有效性，若有效则继续，否则，中止。

其次，验证$\left[\lambda_{\hat{i}}\right],h_{\hat{i}},\left[w_{\hat{i}}\right]$，若$\left[d_{\hat{i}}\right]$、$h_{\hat{i}}$分别是$d_u$的认证数据和哈希值，而$d_u\notin\text{acc}'$，则输出$pk_{\mathcal{P}}$.

(6)$Verify(proof,t,p,\text{acc}',com_M,com_{M'})\rightarrow 0\text{ or }1$：若*Judge*算法被调用且输出$pk_{\mathcal{P}}$，该算法不被调用；否则数据拥有方验证$proof:=(\pi_1,\pi_2)$的有效性，以验证预测$p$是否由摘要$com_{M'}$的原像$M'$通过预测计算$G$生成，且摘要$com_{M'}$的原像$M'$是否通过计算$F$以数据$D$和模型$M$作为输入来生成。若*proof*有效，则输出1，否则，输出0。

7　安全性分析

本文假设验证方(也是数据拥有方)是诚实的，而证明方(即云服务器)会发起恶意欺骗行为，将以一定概率被数据拥有方抓获而受到处罚。此外，也假设方案所采用的承诺、消息验证码、签名、累加器以及ZKP协议满足相应安全性。在这些安全组件上应用Damgård等人的通用编译器[49]，根据文献[49]的定义2和定义3，本节主要分析公开可验证模型遗忘所依赖的三个安全特性，包括隐蔽安全性(Covert Security)、公开可验证性(Public Verifiability)和防诋毁性(Defamation-Freeness)。

具体来说，云服务器被认为是比恶意敌手弱的隐蔽敌手(Covert Adversary)，在本文中，云服务器的秘密认证输入可能使用了未被授权的数据。针对I_w长的秘密认证输入，数据拥有方随机选择其中的$\left\lceil\frac{I_w}{2}\right\rceil-1$进行检查，计算验证方以$\frac{1}{2}-\frac{1}{I_w}$的概率被抓获：

$$1-\frac{\binom{I_w-1}{\left\lceil\frac{I_w}{2}\right\rceil-1}}{\binom{I_w}{\left\lceil\frac{I_w}{2}\right\rceil-1}}=1-\frac{\frac{(I_w-1)!}{\left(\left\lceil\frac{I_w}{2}\right\rceil-1\right)!\left(\left\lceil\frac{I_w}{2}\right\rceil\right)!}}{\frac{I_w!}{\left(\left\lceil\frac{I_w}{2}\right\rceil-1\right)!\left(\left\lceil\frac{I_w}{2}\right\rceil+1\right)!}}=\frac{1}{2}-\frac{1}{I_w}.\tag{2}$$

基于通用编译器[49]获得的$\prod_{\text{PVC}}^{\text{ZKP}}$存在$\frac{1}{2}-\frac{1}{I_w}$的震

慑因子抵御作恶云服务器。在公开可验证方面，考虑到云服务器在离线协议执行时签名了认证数据 $\{[\lambda_i]\}_{i\in[I_w]}$ 和执行流程 $\mathcal{S}_{\mathrm{OT}}$，即 $\{\sigma_i^{\mathrm{OFF},\mathcal{P}}\leftarrow Sign_{\mathrm{sk}_{\mathcal{P}}}(\mathcal{S}_{\mathrm{OT}},[\lambda_i])\}_{i\in[I_w]}$，在线协议执行时签名了秘密输入的认证数据 $\{[d_i]\}_{i\in[I_w]}$ 以及相应秘密数据的哈希值 $\{h_i=H_{prime}(d_i)\}_{i\in[I_w]}$，即 $\{\sigma_i^{ON,\mathcal{P}}\leftarrow Sign_{sk_{\mathcal{P}}}(\mathcal{S}_{OT},h_i,[d_i])\}_{i\in[I_w]}$，因此若云服务器作恶且数据拥有方发现了 $\left[\lambda_i\right],h_i,\left[w_i\right]$ 的不一致性，云服务器所签名的消息可被公开验证。在防诋毁性方面，假如云服务器没有欺骗，执行 *Judge* 算法的一方不能产生一个有效的证书来诋毁云服务器存在欺骗行为，这是因为该方不能在没有云服务器私钥的情形下伪造其有效的签名。

8 实验与结果

本文使用了Python编程语言来实现实验所涉及的算法，并且在计算机配置为64位操作系统、13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz处理器、16 GB内存的笔记本上完成实验评估。

本文模拟了模型遗忘场景。实现了基于ADMM算法的LASSO模型，并使用了公开可下载的糖尿病数据集Diabetes进行模型训练，然后从训练数据集中随机选择1～5个不同的数据点进行删除，并重新训练同一模型，重复50次。模型训练完成后，观察并对比删除了数据点的模型预测结果与未删除数据的模型预测结果。评估预测性能采用了常用的回归模型评估指标，用于评估预测值与真实值的差异，包括标准误差(Standard Error，SE)、平均绝对误差(Mean Absolute Error，MAE)、均方根误差(Root Mean Square Error，RMSE)。这三个误差值越小，说明预测值越接近于真实值，预测准确率越高。

本文在RSA累加器的实现中，设置了3072位长的密钥长度，累加器的素数域哈希 H_{prime} 长度为128位。此外，还实现了基于子域向量的不经意线性函数计算消息认证码，从而实现不同数据规模下的秘密认证数据及相应的验证算法，其中素数域的素数大小为128位。

本文使用了RSA累加器而不是Merkle树作为可公开的认证数据结构是因为它删除数据前后的累加值是概率多项式时间不可区分的，严格遵循第5.3小节中定义二的内容。若不遵循该定义采用Merkle树，在删除少量数据点的情况下，则累加器生成成员证明和删除数据的操作速度更快。基于此考虑，本文也实现了Merkle树作为累加器的功能。

基于上述设置，本文首先通过随机选择不同数据进行遗忘来评估数据删除对模型预测性能的影响；然后，评估不同数据规模下预处理时用输入数据生成秘密认证数据以及验证认证数据有效性的开销；接着评估1000-7000个数据点规模下累加器添加、删除、查询操作以及成员证明和验证的开销。

首先，评估数据遗忘对预测结果的影响。实验对删除数据操作重复执行了10次，如图5所示，横坐标表示重复执行的次数序号，纵坐标表示评估指标。注意，实验中遗忘数据是通过不使用所遗忘数据用于模型训练的。以下是实验评估结果对比。在未删除数据时，SE、MAE和RMSE的三个误差值分别为0.0947、1.7889和2.2249。在随机重复50次删除1-5个数据点过程中，图5(a)的标准误差在0.0877～0.1022之间波动，图5(b)的平均绝对误差在1.7072～2.0728之间上下变化，以及图5(c)的均方根误差在2.1106～2.5199之间波动。在固定所删除数据点数量下的10次重复执行中，SE、MAE和RMSE三个误差值有时降低或

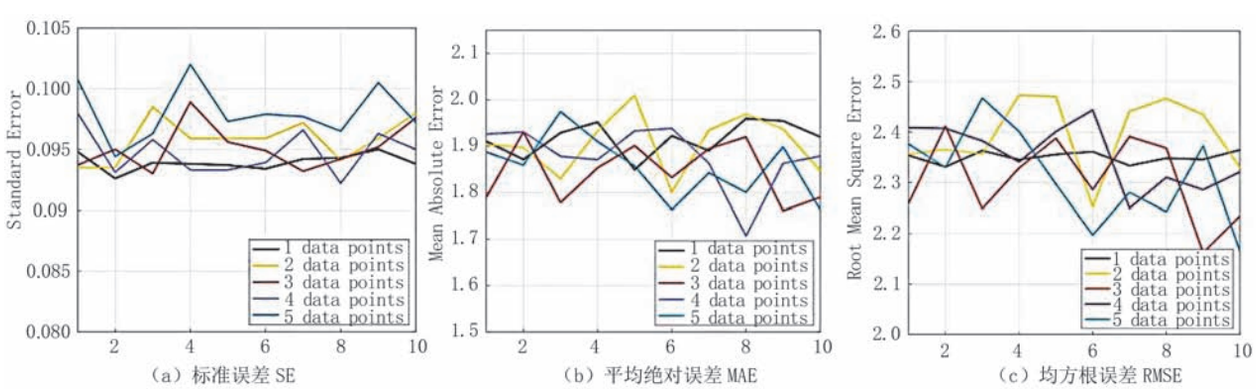

图5 不同数量的数据点删除情况下的模型预测性能(图中横坐标表示重复执行的次数序号)

有时升高，误差值降低表明模型准确率提高而相反则表明模型准确率降低。从图中可知，删除不同的数据点对模型预测误差值存在不同的影响，表明了模型遗忘后的模型预测准确性可能与删除哪些数据点有关。

其次，评估不同数据规模下生成秘密认证数据及其验证开销。在本文方案中，证明方需生成给定输入数据的秘密认证数据，对于10 000至70 000个数据点，秘密认证数据生成的时间开销呈现线性增长趋势，如图6所示。验证方可以验证证明方所生成的秘密认证数据，判断是否确实是给定输入数据的认证值，本文评估批量验证的时间开销，如图7所示，在验证10 000个数据点时，验证时间为2毫秒，验证70 000个数据点的时间为28毫秒。

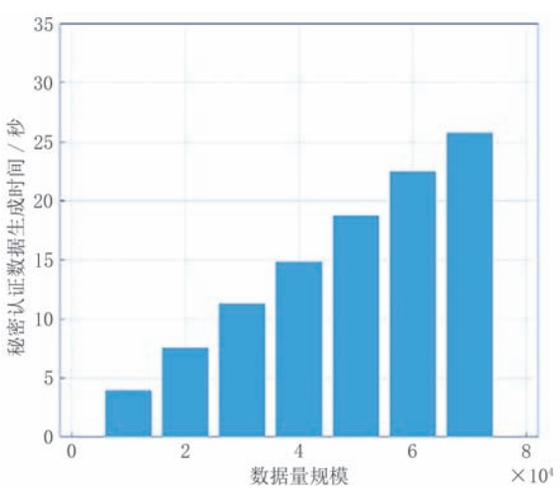

图6　认证时间

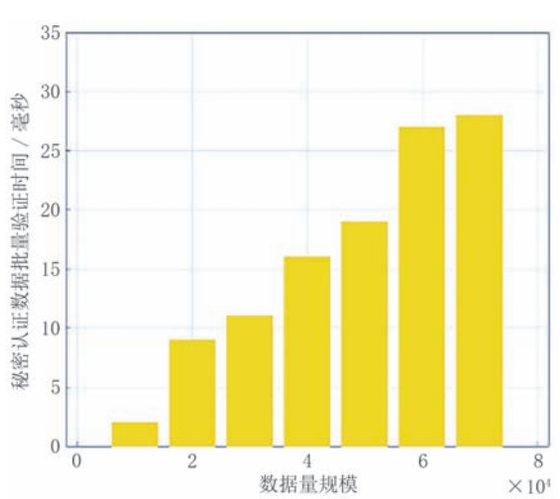

图7　验证时间

本文接着评估累加器相关操作的运行开销，见图8。当数据规模从10 000个数据点增加到70 000个数据点时，累加器添加操作的运行时间几乎呈线性增长，而相应生成一个测试数据的成员证明运行时间也随着数据规模的增加而平缓增加。另一方面，向累加器查询一个测试数据是否被累加的运行时间基本稳定，处于5~6毫秒之间，同时，验证该测试数据的成员证明有效性的运行时间很短，大概1毫秒或更低，如图9所示。

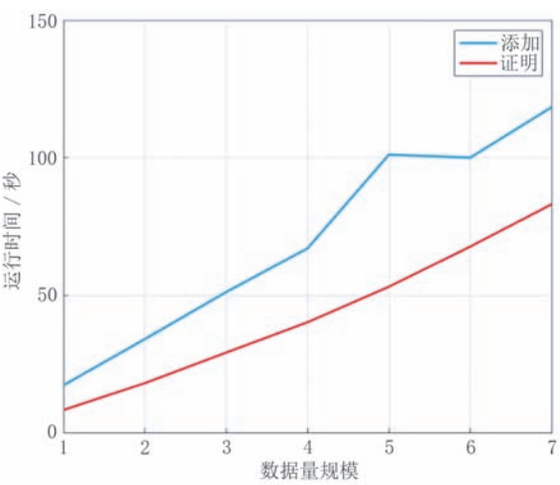

图8　RSA累加器添加数据和生成成员证明的时间

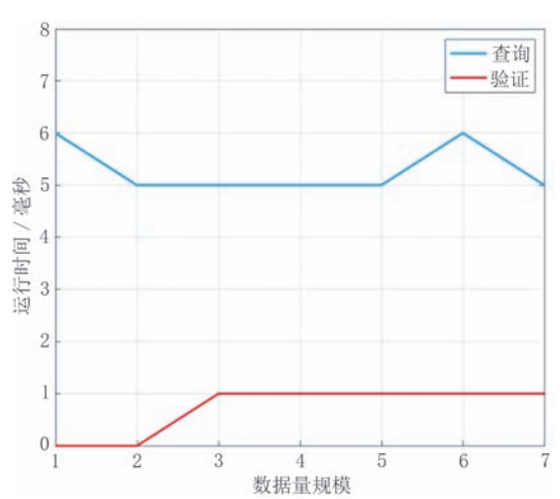

图9　RSA累加器查询数据和验证成员证明的时间

接着，本文针对不同规模数据添加完成的累加器进行删除操作，删除后并询问累加器该数据是否还存于累加值。如图10所示，数据删除和查询的总时间或者数据删除时间与累加器大小正向相关，当数据规模为10 000个数据点时，删除操作时间开销为8.286秒，当70 000个数据点时，时间开销为82.560秒。同时，随着删除数据点数越多，删除操作所耗时间逐渐增加。但对于不同的数据量规模大小，查询操作时间开销基本维持不变。

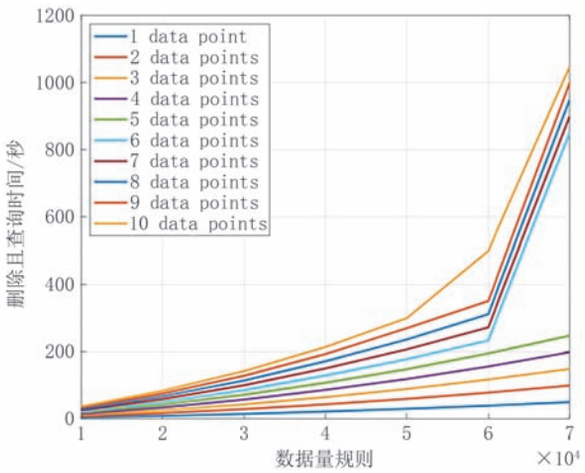

图10 RSA累加器删除且查询数据的时间

实验还模拟了验证方检查数据一致性的计算并评估了相应运行时间开销。在本文方案中，验证方为了检查证明方所使用数据是否为其授权的数据(即累加器内的数据)，需要对证明方的实际输入数据生成相应大素数域上的哈希值，并将生成后的哈希值询问其是否存于累加器。如图11所示，随着数据量规模的增大，验证方计算开销呈现线性增长趋势。

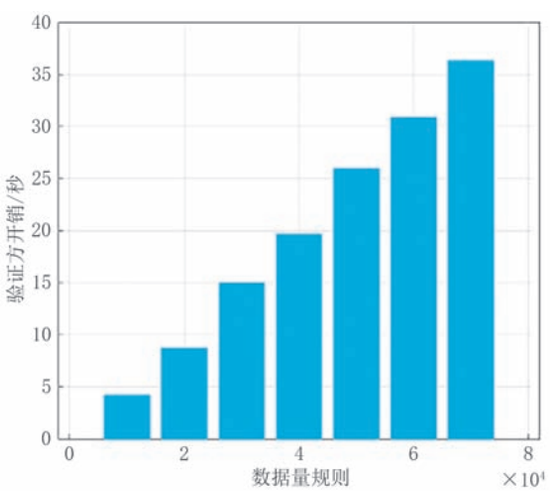

图11 验证方检查数据一致性的时间

此外，实验也使用了Merkle树作为累加器进行实验，表明其添加数据的效率不如RSA累加器的效率，随着数据量增大，其运行时间增长速度远远大于RSA累加器的运行时间增长速度，见图12。

虽然Merkle树添加数据的效率不如RSA累加器的数据添加效率，但在生成成员证明和删除数据的操作方面，其操作速度更快，如表5和表6分别对比了RSA累加器和Merkle树在生成成员证明和删除时的运行时间。

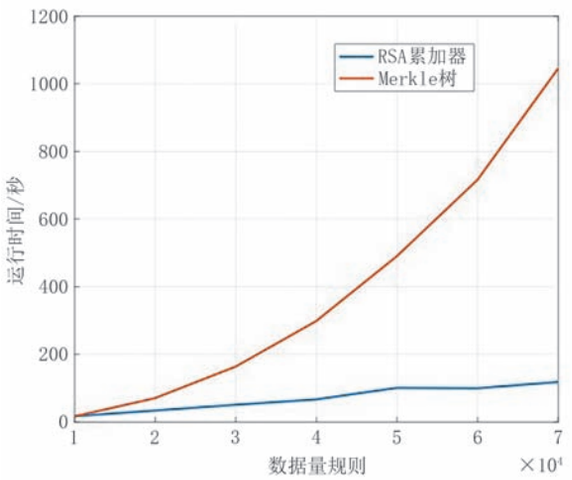

图12 不同累加器添加数据的运行时间对比

表5 RSA累加器和Merkle树在生成成员证明方面的运行时间对比

累计器类型/数据量	RSA累加器	Merkle树
10 000	3.778秒	17毫秒
20 000	8.596秒	16毫秒
30 000	14.794秒	17毫秒
40 000	21.353秒	20毫秒
50 000	29.588秒	18毫秒
60 000	38.950秒	18毫秒
70 000	50.026秒	20毫秒

表6 RSA累加器和Merkle树在数据删除时的运行时间对比

累计器类型/数据量	RSA累加器	Merkle树
10 000	3.754秒	19毫秒
20 000	8.593秒	45毫秒
30 000	14.836秒	70毫秒
40 000	21.402秒	86毫秒
50 000	29.589秒	116毫秒
60 000	39.110秒	136毫秒
70 000	50.562秒	160毫秒

9 总结与未来展望

本文首先定义了公开可验证的模型遗忘问题，该定义不局限于现有的某一类模型遗忘算法。在此基础上，本文针对具体的确切模型遗忘算法和正则化线性模型，提出了一种公开可验证的模型遗忘方案。所提出方案借助可验证计算领域的“先认证再证明”技术，首先对训练数据进行公开认证，然后证明训练过程中使用的数据确实是已认证的公开数

据，并确保经过训练的模型用于提供后续的预测服务。本文方案能够在验证方发现证明方存在不当行为时，生成公开可验证的凭证以追责。现有的大部分工作关注于单个数据点删除的模型遗忘场景，而在实际应用中，遗忘需求可能涉及删除某一标签下的所有数据。为满足这一需求，模型遗忘算法需要扩展至支持多种数据遗忘场景，同时公开可验证的方法也需适应这些场景。此外的未来研究方向之一是开发适配任意模型架构（如Transform架构）的模型遗忘算法及通用的安全可验证模型遗忘方案。

致　谢　衷心感谢编辑和评审专家对本文提出的建设性意见和建议！

参　考　文　献

[1] Artificial Intelligence Industry Situation Analysis Research Group. Analysis of the development situation of China's artificial intelligence industry. China: CCID Think Tank, 2024. http://www.csia-jpw.com/UserFiles/Article/file/638400559487735154798 3703.pdf
（人工智能产业形势分析课题组. 2024年我国人工智能产业发展形势展. 中国：赛迪智库，2024. http://www.csia-jpw.com/UserFiles/Article/file/638400559487735154798 3703.pdf）

[2] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, et al. The secret sharer: Evaluating and testing unintended memorization in neural networks//Proceedings of the 28th USENIX Conference on Security Symposium (USENIX Security 19). ClaraSanta, USA, 2019: 267-284

[3] Reza Shokri, Marco Stronati, Congzheng Song, et al. Membership inference attacks against machine learning models//Proceedings of the 2017 IEEE symposium on security and privacy (IEEE S&P 2017). JoseSan, USA, 2017: 3-18

[4] Florian Tramèr, Fan Zhang, Ari Juels, et al. Stealing machine learning models via prediction APIs//Proceedings of the 25th USENIX Conference on Security Symposium (USENIX Security 16). Austin, USA, 2016: 601-618

[5] Gaoyang Liu, Yutong Li, Borui Wang, et al. Research on membership inference attacks on black box machine learning models. Journal of Information Security, 2021, 6(3): 1-15(in Chinese)
（刘高扬，李雨桐，万博睿等. 黑盒机器学习模型的成员推断攻击研究. 信息安全学报，2021，6(3)：1-15）

[6] Nils Lukas, Ahmed Salem, Robert Sim, et al. Analyzing leakage of personally identifiable information in language models// Proceedings of the2023 IEEE Symposium on Security and Privacy (IEEE S&P 2023).FranciscoSan, USA, 2023: 346-363

[7] Beizheng Lin. Not deletion but forgetting: Explanation and reconstruction for the obligation to delete personal information in AI large models. Journal of Xi'an Jiaotong University (Social Science Edition), 2024:1-16(in Chinese)
（林北征. 没有删除，只能遗忘：AI大模型个人信息删除义务的解释与重构. 西安交通大学学报（社会科学版），2024：1-16）

[8] Yinzhi Cao, Junfeng Yang. Towards making systems forget with machine unlearning//Proceedings of the 2015 IEEE Symposium on Security and Privacy (IEEE S&P 2015). San Jose, USA, 2015: 463-480

[9] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, et al. Machine unlearning//Proceedings of the 2021 IEEE Symposium on Security and Privacy (IEEE S&P 2021). San Francisco, USA, 2021: 141-159

[10] Varun Gupta, Christopher Jung, Seth Neel, et al. Adaptive machine unlearning//Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS 2021). Virtual, 2021: 16319-16330

[11] Chong Chen, Fei Sun, Min Zhang, et al. Recommendation unlearning//Proceedings of the ACM Web Conference 2022 (ACM WWW 2022). Virtual Event, Lyon, France, 2022: 2768-2777

[12] Min Chen, Zhikun Zhang, Tianhao Wang, et al. Graph unlearning//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (ACM CCS 2022). Los Angeles, USA, 2022: 499-513

[13] Haonan Yan, Xiaoguang Li, Ziyao Guo, et al. Arcane: An efficient architecture for exact machine unlearning//Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022). Vienna, Austria, 2022: 4006-4013

[14] Hu Yuke, Lou Jian, Liu Jiaqi, et al. ERASER: Machine unlearning in MLaaS via an inference serving-aware approach. https://doi.org/10.48550/arXiv.2311.16136, 2023

[15] Antonio A. Ginart, Melody Y. Guan, Gregory Valiant, et al. Making AI forget you: Data deletion in machine learning// Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver, Canada 2019: 3513-3526

[16] Sekhari Ayush, Jayadev Acharya, Gautam Kamath, et al. Remember what you want to forget: Algorithms for machine unlearning//Proceedings of the 35th International Conference on Neural Information Processing Systems (NeurIPS 2021). Virtual, 2021: 18075-18086

[17] Aditya Golatkar, Alessandro Achille, Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020). Seattle, USA, 2020: 9301-9309

[18] Aditya Golatkar, Alessandro Achille, Stefano Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations// Proceedings of the European Conference on Computer Vision 2020(ECCV 2020). Glasgow, UK, 2020: 383-398

[19] Anvith Thudi, Hengrui Jia, Ilia Shumailov, et al. On the necessity of auditable algorithmic definitions for machine unlearning//Proceedings of the 31st USENIX Security Symposium (USENIX Security 22). Boston, USA, 2022: 4007-4022

[20] Guo Yu, Zhao Yu, Hou Saihui, et al. Verifying in the dark:

verifiable machine unlearning by using invisible backdoor triggers. IEEE Transactions on Information Forensics and Security, 2023, 19:708-721

[21] D. M. Sommer, L. Song, S.Wagh, et al. Towards probabilistic Verification of machine unlearning. https://arxiv. org/abs/2003.04247, 2020

[22] Gao Xiangshan, Ma Xingun, Wang Jingyi, et al. Veri Fi: Towards verifiable federated unlearning. IEEE Transactions on Dependable and Secure Computing, 2024 (early access): 1-16.

[23] Zhou Juexiao, Li Haoyang, Liao Xingyu, et al. Audit to forget: Aunifiedmethod to revoke patients' private data in intelligent healthcare.https://doi.org/10.48550/arXiv.2302.09813, 2023

[24] Shi Weijia, Ajith Anirudh, Xia Mengzhou, et al. Detecting pretraining data from large language models. https://arxiv.org/pdf/2310.16789, 2023

[25] Ayush Alag, Yangsibo Huang, Kai Li. Is EMA robust? Examining the robustness of data auditing and a novel non-calibration extension//Proceedings of the Neural Information Processing Systems 2023 Workshop on Regulatable Machine Learning(NeurIPS 2023 Workshop on Regulatable ML). New Orleans, USA, 2023: 1-9

[26] Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, et al. Proof-of-learning: Definitions and practice// Proceedings of the 2021 IEEE Symposium on Security and Privacy (IEEE S&P 2021). San Francisco, USA, 2021: 1039-1056

[27] Rui Zhang, Jian Liu, Yuan Ding, et al. "Adversarial examples" for proof-of-learning//Proceedings of the 2022 IEEE Symposium on Security and Privacy (IEEE S&P 2022). San Francisco, USA, 2022: 1408-1422

[28] Michael Walfish, Andrew J Blumberg. Verifying computations without reexecuting them. Communications of the ACM, 2015, 58(2):74-84

[29] Benjamin Braun, Ariel J. Feldman, Zuocheng Ren, et al. Verifying computations with state//Proceedings of the 24th ACM Symposium on Operating Systems Principles (SOS&P 2013). Farminton, Pennsylvania, 2013: 341-357

[30] Alex Ozdemir, Riad Wahby, Barry Whitehat, et al. Scaling verifiable computation using efficient set accumulators// Proceedings of the 29th USENIX Security Symposium (USENIX Security 20). Berkeley, USA, 2020: 2075-2092

[31] Chenkai Weng, Kang Yang, Xiang Xie, et al. Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning//Proceedings of the 30th USENIX Security Symposium (USENIX Security 21). virtual, 2021: 501-518

[32] Michael Backes, Dario Fiore, Raphael M. Reischuk. Verifiable delegation of computation on outsourced data//Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (ACM CCS 2013). Berlin, Germany, 2013: 863-874

[33] Dario Catalano, Dario Fiore, Bogdan Warinschi. Homomorphic signatures with efficient verification for polynomial functions// Proceedings of the Advances in Cryptology-CRYPTO 2014: 34th Annual International Cryptology Conference (CRYPTO 2014). Santa Barbara, USA, 2014: 371-389

[34] Michael Backes, Manuel Barbosa, Dario Fiore, et al. Adsnark: Nearly practical and privacy-preserving proofs on authenticated data//Proceedings of the 2015 IEEE Symposium on Security and Privacy (IEEE S&P 2015). JoseSan, USA, 2015: 271-286

[35] Christian Cachin, Esha Ghosh, Dimitrios Papadopoulos, et al. Stateful multi-client verifiable computation//Proceedings of the 16th International Conference on Applied Cryptography and Network Security (ACNS 2018). Leuven, Belgium, 2018: 637-656

[36] Dario Fiore, Cédric Fournet, Esha Ghosh, et al. Hash first, argue later: Adaptive verifiable computations on outsourced data//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (ACM CCS 2016). Vienna, Austria, 2016: 1304-1316

[37] Craig Costello, Cédric Fournet, Jon Howell, et al. Geppetto: Versatile verifiable computation//Proceedings of the 2015 IEEE Symposium on Security and Privacy (IEEE S&P 2015). JoseSan, USA, 2015: 253-270

[38] Matteo Campanelli, Dario Fiore, Anaïs Querol. Legosnark: Modular design and composition of succinct zero-knowledge proofs//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (ACM CCS 2019). London, United Kingdom, 2019: 2075-2092

[39] Roberto Tamassia. Authenticated data structures//Proceedings of the 11th Annual European Symposium on Algorithms (ESA 2003). Budapest, Hungary, 2003: 2-5

[40] Dan Boneh, Benedikt Bünz, Ben Fisch. Batching techniques for accumulators with applications to IOPs and stateless blockchains// Proceedings of the Advances in Cryptology-CRYPTO 2019: 39th Annual International Cryptology Conference (CRYPTO 2019). Santa Barbara, USA, 2019: 561-586

[41] Wenting Zheng, Raluca Ada Popa, Joseph E. Gonzalez, et al. Helen: Maliciously secure coopetitive learning for linear models// Proceedings of 2019 IEEE Symposium on Security and Privacy (IEEE S&P 2019). FranciscoSan,USA, 2019: 724-738

[42] Weng Jiasi, Yao Shenglong, Du Yuefeng, et al. Proof of unlearning: Definitions and instantiation. IEEE Transactions on Information Forensics and Security, 2024, 19: 3309-3323

[43] Eisenhofer Thorsten, Riepel Doreen, Chandrasekaran Varun, et al. Verifiable and provably secure machine Unlearning. https://arxiv.org/pdf/2210.09126, 2022

[44] Josh Benaloh, Michael De Mare. One-way accumulators: A decentralized alternative to digital signatures//Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology (EUROCRYPT 1993). Lofthus, Norway, 1994: 274-285

[45] Niko Baric, Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees//Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT 1997). Konstanz, Germany, 1997: 480-494

[46] Jingtao Li, Ninghui Li, Rui Xue. Universal accumulators with efficient nonmembership proofs//Proceedings of the 5th

Conference on Applied Cryptography and Network Security (ACNS 2007). Zhuhai, China, 2007: 253-269

[47] Gilad Asharov, Claudio Orlandi. Calling out cheaters: Covert security with public verifiability//Proceedings of the 18th International Conference on The Theory and Application of Cryptology and Information Security (CRYPTO 2012). Beijing, China, 2012: 681-698

[48] Cheng Hong, Jonathan Katz, Vladimir Kolesnikov, et al. Covert security with public verifiability: faster, leaner, and simpler//Proceedings of the Advances in Cryptology-EUROCRYPT 2019: 38 th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2019). Darmstadt, Germany, 2019: 97-121

[49] Ivan Damgård, Claudio Orlandi, Mark Simkin. Black-box transformations from passive to covert security with public verifiability//Proceedings of the Advances in Cryptology-CRYPTO 2020: 40th Annual International Cryptology Conference (CRYPTO 2020). Santa Barbara, USA, 2020: 647-676

[50] Jens Groth. On the size of pairing-based non-interactive arguments//Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2016). Vienna, Austria, 2016: 305-326

[51] Sanjam Garg, Shafi Goldwasser, Prashant Nalini Vasudevan. Formalizing data deletion in the context of the right to be forgotten//Proceedings of the 16th Annual International Conference on Theory and Application of Cryptographic Techniques(EUROCRYPT 2020). Virtual, 2020:373-402

WENG Jia-Si, Ph. D., associate researcher. Her research interest is privacy-preserving machine learning.

GU Yan-Yun, M. S. candidate. Her research interest is machine learning security.

LIU Jia-Nan, Ph. D., associate researcher. His research interests include applied cryptography and data security.

LI Ming, Ph. D., associate researcher. His research interests include blockchain security and data security.

WENG Jian, Ph. D., professor. His research interest is public-key cryptography.

Background

This paper addresses the challenge of enabling verifiable machine unlearning in scenarios where the model owner may act maliciously. Previous verification methods have relied on crafted backdoored samples or membership inference attacks, assuming a fully honest environment. However, these methods are insufficient against a malicious model owner who might deviate from the unlearning process, such as model forging attacks, as recent studies have demonstrated. Weng et al., in their IEEE TIFS 2024 paper, introduced a proof of unlearning definition and implemented it using a trusted execution environment (TEE). Similarly, Eisenhofer et al. in their preprint proposed a solution using zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) techniques, although their definition did not encompass the model prediction phase, leaving it vulnerable to model forking by a malicious model owner. In contrast to these previous works, this paper introduces a comprehensive set of algorithms for verifiable unlearning that considers both the training and prediction phases, with the aim of mitigating the risks of model forging and forking attacks. Additionally, the paper presents a new TEE-free verification method by integrating a cryptographic accumulator, an interactive zero-knowledge proof protocol with publicly verifiable cover security. This method is designed to be executed between the data owner and the model owner, allowing the data owner to produce undeniable evidence verifiable by any third party in cases of malicious behavior. This work integrates cryptographic tools and machine unlearning, which can represent a significant advancement in ensuring compliance with "the Right

to Be Forgotten" in the context of machine learning.

The authors have been working on this area for more than 6 years, and published related papers. This new work offers a crypto-enabled verifiable unlearning approach, providing a robust solution to the potential threats posed by model forging and forking attacks.

This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China (Grant Nos. 62302192, 62102166), and the Key Program of the National Natural Science Foundation of China (Grant No. 62332007), and the Joint Funds of the National Natural Science Foundation of China (Grant No. U23A20303), and the Natural Science Foundation of Guangdong Province (No. 2024A1515010086), and the Science and Technology Program of Guangzhou (Nos. 2024A04J3691, 2024A03J0464), and the China Postdoctoral Science Fund-Special Fund (No. 2024T170348), and the Machine Learning and Cyber Security Interdisciplinary Research Engineering Center of Jiangsu Province, and the Fundamental Research Funds for the Central Universities, and the Dongguan Social Development Science and Technology Key Project (No. 20231800940342).