Title

Subtitle

Author

Date

Abstract

# 保护用户隐私数据

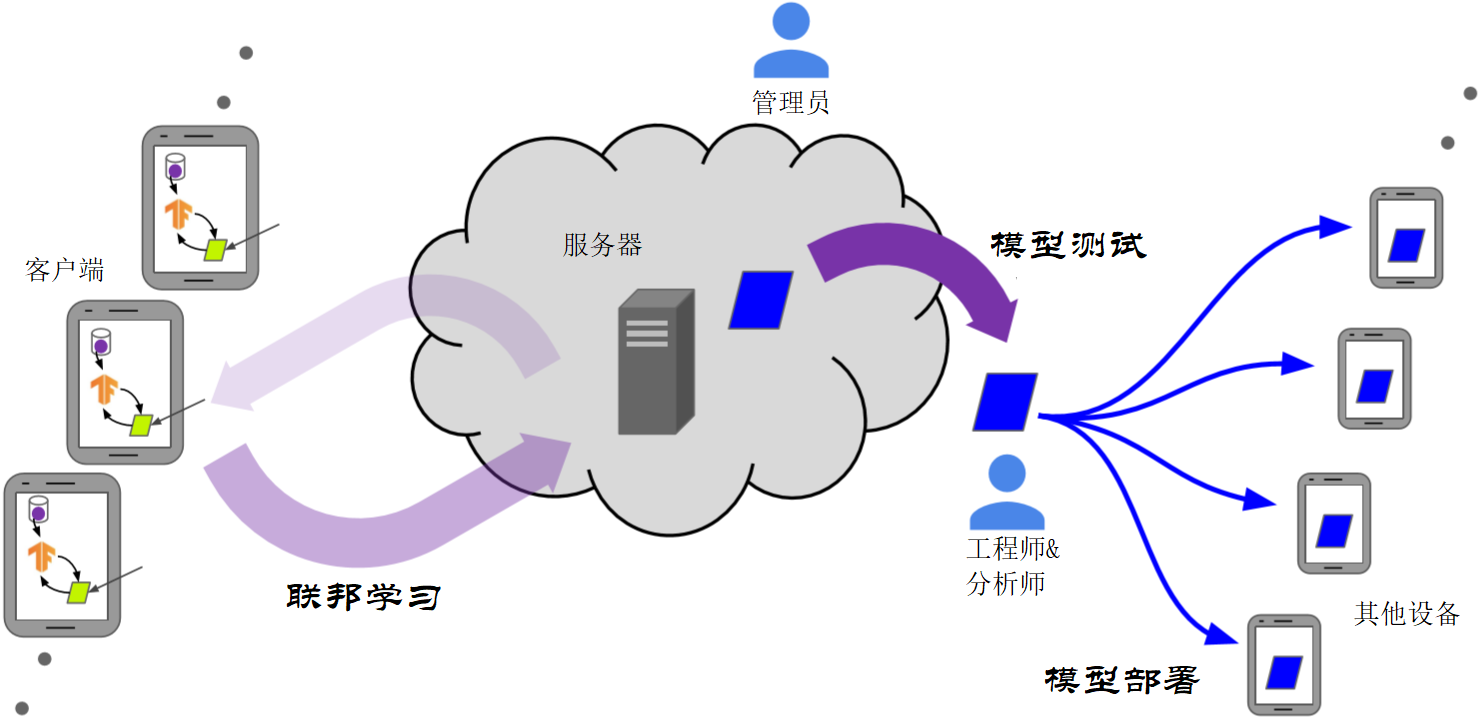
机器学习过程由许多功能不同的角色参与运作。例如，用户可以通过与设备交互来生成训练数据，在机器学习训练过程中，其从这些数据中提取人机交互模式（例如，以训练后模型参数的形式），之后机器学习工程师或分析师可以评估该训练模型的质量，最后可能将该模型部署在最终用户主机上，以支持定制的用户体验（见下图1）。

在一个理想的世界里，系统中的每一个参与者只会学到扮演他们角色所需的信息。例如，如果分析师只需要确定某个特定的质量度量是否超过了所需的阈值，以便授权将模型部署到最终用户，那么在理想化的世界中，该度量值是分析师可以获得的唯一信息；该分析师既不需要访问训练数据，也不需要访问模型参数。类似地，最终用户体验到的可能只需要由经过训练的模型提供的预测，而不需要其他任何内容。

此外，在理想的世界中，系统中的每个参与者都能够轻松、准确地推断出自己和他人的哪些个人信息可能通过参与系统而泄露，参与者将能够利用这一推理结果，就是否参与以及如何参与做出明智的选择。

创建一个具有上述所有理想隐私属性的系统本身将是一项令人望而生畏的壮举，若其还可以实现其他令人满意的属性，则更是难上加难，比如所有参与者的易用性、最终用户体验的质量和公平性（以及影响体验结果的模型），智能地使用通信和计算资源、抵御攻击和失败的能力等。

与其追求无法企及的完美，我们不如另辟蹊径——整个系统由模块化单元组成，这些单元可以相对独立地进行学习和改进，同时我们也要注意，我们最终必须根据上述我们的理想隐私目标，测量整个系统的隐私属性。本节中，我们将提出目前无论是通过单个模块实现方法还是通过整个系统实现方法，还不了解如何同时实现所有目标的领域，作为开放性研究问题。



*图4-1 联邦学习模型的生命周期及联邦学习系统中的不同参与者*

*（与第7页图相同）*

联邦学习提供了一个有吸引力的结构，可以将整个机器学习工作流程分解成我们想要的可实现的模块单元。联邦学习模型的一个主要优点是它可以通过数据最小化为参与的用户提供一定程度的隐私：设备从不发送原始用户数据，只将对模型的更新（例如梯度更新）发送到中央服务器。这些模型更新更侧重于要完成的学习任务而非关注原始数据（即，与原始数据相比，它们严格不包含关于用户的附加信息，而且通常不包含其他意义），并且单个更新只需要由服务器暂时保存。

虽然这些特性可以在集中所有训练数据的基础上提供显著的实用性隐私改进，但是在这个基线联邦学习模型中仍然没有隐私的正式保证。例如，可以构造这样的场景，在该场景中，原始数据的信息从客户端泄漏到服务器。比如，知道以前的模型和用户的梯度更新将允许某一方推断该用户持有的训练示例。因此，本节调查现有的结果，并概述了设计可以提供严格的隐私保障的联邦学习系统的开放性的挑战。我们更专注于联邦学习和分析设置中特定的问题，而不考虑了在一般的机器学习设置中也会出现的问题。

除了针对用户隐私的攻击之外，还有其他种类的对联邦学习的攻击；例如，对方可能会试图阻止模型被学习，或者他们可能会试图使模型产生偏向对方的训练结果。

我们稍后在第5章讨论这些类型的攻击的讨论。本章的其余小节梗概如下。第4.1节讨论了我们希望提供抵御的各种威胁模型。第4.2节列出了一套核心工具和技术，可用于针对第4.1节中讨论的威胁模型提供严格的保护。第4.3节假定可信服务器的存在，并讨论在对抗对手和/或分析师提供保护方面的公开问题和挑战。第4.4节讨论了在没有完全可信服务器的情况下的开放性问题和挑战。最后，第4.5节讨论了关于用户感知的开放性问题。

## 参与者，威胁模型与深层隐私

在联邦学习中，对隐私风险的规范处理需要一种整体的、跨学科的方法。对于一些风险类型，可以通过将现有技术扩展到指定场景中从而保护隐私和减轻风险，而其他更复杂的风险类型则需要跨学科的协同努力。

隐私不是二进制量，甚至不是标量。这种规范处理的第一步是准确描述不同的参与者（见第1节图1，为方便起见，在第35页重复）及其在模型中扮演的角色，最终确定相关的威胁模型（见表7）。例如，我们希望将服务器管理员的视图与使用所学模型的分析师的视图区分开来，因为可以想象，设计用于针对恶意分析师提供强大隐私保证的系统可能不会提供任何恶意攻击的机会。这些参与者行为模式很好映射到其他文献中讨论的威胁模型上；例如，在Bittau等人中。[67，第3.1节]，“编码器”对应于客户机，“洗牌者”通常对应于服务器，“分析器”可能对应于服务器或分析师完成的后处理。

例如，一个特定的系统可能提供差异性的隐私保证[[1]](#footnote-1)，比如向服务器管理员提供的特定参数为ε，而分析师观察到的参数结果可能具有更高的保护性ε’<ε。

此外，这一保证可能仅适用于能力受到特定限制的对手，例如可以观察服务器上发生的所有事情（但不能影响服务器的行为）的对手，这类对手同时完全控制占客户端总数比例为γ的客户端（“完全控制”即可以观察他们可获知的所有数据并以任意方式影响其行为）；对手也可能被认为无法破解在特定安全级别σ下，实例化的加密机制。为了对抗实力突破这些限制的对手，在服务器管理员的看来可能仍然需要差异性隐私保证，但认为分析师观察的结果在较弱的隐私保护级别ε’>ε。

正如我们在本例中看到的，精确地指定系统的假设和隐私目标，以及诸如差异性隐私保证、老实但好奇行为等安全性概念，可以很容易地通过到几个参数（ε、ε’、ε0、γ、σ等）进行具体实例化。

实现联邦学习所需的所有隐私属性通常需要将下述的许多工具和技术组合到端到端系统中，包括以下两种：多种分层策略都是为了保护系统的同一部分（例如，在可信执行环境（TEE）中运行安全多方计算（MPC）协议的一部分，使对手更难对该组件产生足够大的损害）以及使用不同的策略来保护系统中不同的部分（例如，使用MPC保护模型更新的聚合，然后在服务器之外共享聚合更新之前使用隐私披露技术）。

因此，在两种技术都无法提供其预期隐私保护能力的情况下，我们提倡构建这样一种优美的联邦系统，即尽可能降低隐私性。例如，在TEE中运行MPC协议的服务器组件可能允许维护隐私，即使TEE安全性或MPC安全性假设中的一个（但不是两个）在实践中不成立。另一个例子是，要求客户端向服务器端TEE发送渐变更新，而要求客户端将原始训练示例发送到服务器端TEE将被强烈不推荐。因为一旦TEE的安全性失效，前者的隐私性期望将更优美地降级。我们将这种优美降级的原则称为“深度隐私”，类似于成熟的深度防御网络安全原则[311]。

## 工具与技术

一般来说，联邦学习计算的目的是让分析师或工程师通过计算请求获得结果，这可以看作是对分布式客户机数据集上的函数*f*的评估（通常是机器学习模型训练算法，但可能更简单，例如基本的数据统计）。有三个隐私方面需要解决。

首先，我们需要考虑*f*是如何计算的，以及在这个过程中中间结果的信息流是什么，它主要影响对恶意客户端、服务器和管理参与者的敏感性。除了设计系统中的信息流（例如提前数据最小化）外，包括安全多方计算（MPC）和可信执行环境（TEEs）等安全计算相关技术对于解决这些问题特别重要。这些技术将在第4.2.1节中详细讨论。

其次，我们必须考虑该计算哪些内容。换言之，*f*本身的计算结果会向分析师和域内参与者透露了参与客户的多少信息。这与隐私保护披露技术，特别是差异隐私（DP）是高度相关的，将在第4.2.2节中详细讨论。

最后，可验证性也是需要考虑的问题，即客户机或服务器能够向系统中的其他人证明他们已忠实地运行了所需的指令，而不泄露他们运行过程中的潜在隐私数据。验证技术，包括远程认证和零知识证明，将在第4.2.3节中讨论。

### 安全计算

安全计算的目标是评估计算分散输入的函数，通过判断其是否仅向预期各方显示计算结果，而不显示任何附加信息（例如各方的输入或任何中间结果）。

**安全多方计算：**安全多方计算（MPC）是密码学的一个子领域，与这样一个问题有关：一组参与方计算其隐私输入通过共识函数得到输出，从而可以只向每个参与方显示期望的输出。这一领域在20世纪80年代由姚[422]开创。由于理论和工程上的突破，该领域已经从单纯的理论研究转向工业上的部署技术[71、70、257、29、169、209、210]。值得注意的是，MPC定义了一组技术，应该更多地被视为安全计算中的领域或安全性的一般概念，而不是技术本身。MPC的一些最新进展可以归因于低级原语的突破，例如不经意传输协议[211]和具有同态性质的加密方案（如下所述）。

表4-1 不同的敌对参与者的多种威胁模型

|  |  |  |
| --- | --- | --- |
| **数据/访问节点** | **参与者** | **威胁模型** |
| 客户端 | 通过系统设计或破坏设备获得客户端设备的最高访问权限者 | 恶意客户端可以检查所参与轮次从服务器接收的全部消息（包括模型迭代），并可以篡改训练过程。老实但好奇的客户端可以检查从服务器接收的所有消息，但不能篡改培训过程。在某些情况下，安全包围/TEEs等技术可能会限制此类攻击者的影响和信息可见性，从而削弱该模型威胁程度。 |
| 服务器 | 通过系统设计或破坏设备获得服务器设备的最高访问权限者 | 恶意服务器可以检查所有轮次发送到服务器的全部消息（包括梯度更新），并可以篡改训练过程。老实但好奇的客户端可以检查发送到服务器的所有消息，但不能篡改培训过程。在某些情况下，安全包围/TEEs等技术可能会限制此类攻击者的影响和信息可见性，从而削弱该模型威胁程度。 |
| 输出模型 | 工程师与分析师 | 恶意分析师或模型工程师可以访问系统的多组输出，例如，使用不同超参数的多个训练运行的模型迭代序列。该向这类参与者发布什么信息是一个重要的系统设计问题。 |
| 部署模型 | 其他设备 | 在跨设备联邦学习场景下，最终模型可能部署到数亿个设备上。访问部分受损的设备可能仍满足黑盒模型，而访问完全受损的设备可以认为是白盒模型。 |

密码学解决方案的共同点是，操作通常在一个有限的字段上完成（即，素数*p*都是整数），这在表示实数时会带来困难。一种常见的方法是调整机器学习模型及其训练程序，即通过标准量化操作并依赖精心设计的量化模式，以确保下（上）溢量在可控范围[172、14、182、77]。

即使在恶意对手面前，任何函数都可以安全计算[183]这一点在几十年间达成了共识。虽然通用解决方案存在，但它们的性能特征常常使它们在实际设置中不适用。因此，研究显著趋势是线性和逻辑斯蒂回归[309，172，302]和神经网络训练和推理[302，14，46]等应用设计定制协议。这些协议通常在孤井互通的设置中进行，或是将计算委托给一组不相互协作的计算服务器的变体模型。将这些协议移植到跨设备设置并不简单，因为它们需要大量的通信。

*同态加密* 同态加密（Homomorphic encryption）方案允许在密文上直接执行某些数学运算，而无需事先解密。同态加密通过使参与者计算函数值，同时保持值隐藏，是一个使MPC成为可能的强大工具。

从一般的全同态加密（FHE）〔176〕到更高水平的变体[79, 160, 80，112 ]，同态加密存在多种实现[3, 350, 4 ]。一些称为部分同态的方案同样具有实际意义，例如包括ElGamal和Paillier，允许同态加法或乘法。加性同态加密称为孤井互通设置中MPC协议的一种成分[309 198]。文献[345]调研了一些同态加密软件库，并简要说明了选择库时应考虑的标准/特性。

表4-2 不同技术及其特性描述

|  |  |
| --- | --- |
| **技术** | **特性描述** |
| 差异隐私（本地、中心、混编、聚合、混合模型） | 从包含用户的数据集的输出分析中可以了解到的个人信息量。具有差分隐私的算法必然包含一定数量的随机性或噪声，可以对其进行调整以掩盖用户对输出的影响。 |
| 安全多方计算 | 两个或多个参与者协作，通过密码学模拟完全可信的第三方，第三方满足：  •计算所有参与者提供的输入的函数；  •向选定的部分参与者显示计算结果，同时任一方没有进一步学习。 |
| 同态加密 | 允许一方在不具有纯文本访问权限的情况下，不解密密文下对密文执行数学运算，从而计算出它们的数据的函数。尽管计算成本更高，任意复杂度的数据函数都可以通过这种方式计算（“完全同态加密”）。 |
| 可信执行环境（安全环境） | 可信执行环境提供了在远程计算机上可靠地运行代码的能力，即使不信任计算机的所有者/管理员。这是通过限制任何一方（包括管理员）的能力来实现的。尤其是，可信执行环境具有以下性质373]：  •一致性：除非程序显式发布消息，否则程序的执行状态始终不可见；  •完整性：除非程序显式接收输入，否则程序的执行不会受到影响；  •可测量/认证性：可信执行环境可以向远程方证明什么程序（二进制）正在执行，以及它的起始状态是什么，定义了一致性和完整性的初始条件。 |

考虑在联邦学习设置中使用同态加密，会遇到谁该持有该模式的密钥这一问题。虽然每个客户机加密其数据并将其发送到服务器端进行同态计算的想法很有吸引力，但服务器不应该能够解密单个客户机的提交数据。克服这一问题的一个简单方法是依赖一个持有密钥并解密计算结果的外部非合谋方。然而，大多数同态加密方案要求密钥经常更新（例如，由于易受选择密文攻击[102]）。此外，使用信任的非共谋方不在标准的联邦学习设置中。

解决此问题的另一种方法是依赖于分布式（或阈值）加密方案，其中密钥在各方之间分发。Reyzin等人。[336]和Roth等人。[341]提出在跨设备设置中计算总和的这种解决方案。他们的协议使用了加性同态方案（分别是基于ElGamal和基于格的方案的变体）。

**可信执行环境：**可信执行环境（TEEs，也称为安全环境）可以将联邦学习过程的一部分转移到云中的可信环境中，而该环境的代码可以被证明和验证。可信执行环境拥有几个关键性质，使他人相信，一段程序已被忠实而保密地执行[373]：

•一致性：除非程序明确发布消息，否则程序的执行状态仍是保密的。

•完整性：除非程序显式地接收输入，否则程序的执行不会受到影响。

•可测量/认证性：可信执行环境可以向远程方证明什么程序（二进制）正在执行，以及它的起始状态是什么，定义了一致性和完整性的初始条件。

可信执行环境已经在实现不同体系结构上被，包括英特尔的SGX处理器[208，116]、ARM的TrustZone[2，1]和Sanctumon RISC-V[117]，它们在上述关键性质上的性能各不相同。

当前的安全环境在内存方面受到限制，只提供对CPU资源的访问，即它们不允许在GPU或机器学习处理器上进行处理（Tram`er和Boneh[382]探索如何将可信执行环境与GPU结合起来进行机器学习推断）。此外，对于可信执行环境（特别是那些在共享微处理器上操作的可信执行环境）来说，完全排除所有类型的侧信道攻击也是一项挑战[391]。

虽然安全环境为运行在其中的所有程序提供保护，但在实践中还必须解决其他问题。例如，通常有必要将运行在该环境中的代码构造为一个不受数据影响的过程，这样它的运行时和内存访问模式就不会显示它正在计算的数据的信息（参见示例[67]）。此外，可测量/证明性通常只证明某个特定的二进制文件正在运行；系统架构师需要提供一种方法来证明该二进制文件具有所需的隐私属性，这可能需要使用来自开源代码的复现过程来重构该二进制文件。

如何在安全环境、云计算资源和客户端设备之间划分联邦学习功能仍然是一个悬而未决的问题。例如，安全环境可以执行安全聚合或混编等关键功能，以限制服务器对原始客户端上传内容的访问，同时此可信计算基础不涉及大多数联合学习逻辑。

**受关注的安全计算问题**：虽然安全多方计算和可信执行环境为分布式隐私数据上的任何函数的保密计算问题提供了一般解决方案，但许多优化可以应用到某些特定功能中。下面描述的任务就是这种情况。

*安全聚合* 安全聚合是指在*n*个客户端和一个服务器的场景中，允许每个客户端提交一个值（通常是联邦学习设置中的向量或张量），这样服务器只学习客户端值的一个聚合函数，通常是这些值的和。

大量文献对单服务器设置（通过成对加法遮蔽[12，188，73]、通过阈值同态加密[356，193，92]、通过一般安全多方计算[86]）以及在多个非合谋服务器设置[71，29，113]中的安全聚合进行了探讨。也可以使用可信执行环境（如上所述）来实现安全聚合，如[269]。

*安全混编* 安全混编是指在*n*个客户端和服务器的场景中，允许每个客户机提交一条或多条消息，这样服务器只从所有客户端学习一个无序的消息集合（multiset），而不需要更多。具体来说，除了消息本身包含的信息之外，服务器无法判断任一条消息的发送者。安全混编可以被视为安全聚合的一个实例，其中值是多集单例，聚合操作为多集求和，尽管通常情况下，为达到安全混编和安全聚合的最佳性能，典型操作制度提供了非常不同的实现

在安全多方计算的背景下，安全混编通常是在混合网络的标题下进行研究[95 251]，也有在可信计算的背景下进行的研究[67]。混合网络中已经存在了以Tor网络[138]的形式进行的大规模部署。

Private information retrieval Private information retrieval (PIR) is a functionality for one client and one server. It enables the client to download an entry from a server-hosted database such that the server gains zero information about which entry the client requested. MPC approaches to PIR break down into two main categories: computational PIR (cPIR), in which a single party can execute the entire server side of the protocol [249], and information theoretic PIR (itPIR), in which multiple non-colluding parties are required to execute the server side of the protocol [106]. The main roadblocks to the applicability of PIR have been the following: cPIR has very high computational cost [361], while the non-colluding parties setting has been difﬁcult to achieve convincingly in industrial scenarios. Recent results on PIR have shown dramatic reductions in the computational cost through the use of lattice-based cryptosystems [16, 313, 17, 25, 175]. It has been shown how to construct communication-efﬁcient PIR on a single-server by leveraging side information available at the user [218]. Recent works propose to leverage client local state to speed up PIR. Patel et al. [319] shows how the latter side information can be obtained and a practical hybrid (computational and information theoretic) PIR scheme on a single server was implemented and validated. Corrigan-Gibbs and Kogan [114] present protocols for PIR with sublinear online time by working in an ofﬂine/online model where, during an ofﬂine phase, clients fetch information from the server(s) independent on the future query to be performed.

42

Further work has explored the connection between PIR and secret sharing in the past [410] with recent connectionstoPIRoncodeddata[139]andcommunicationefﬁcientPIR[66]havingbeenestablished. PIR has also been studied in the context of ON-OFF privacy, in which a client is permitted to switch off their privacy guards in exchange for better utility or performance [306, 423].

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

Heading 7

Heading 8

Heading 9

First Paragraph.

Body Text. Body Text Char. Verbatim Char .  [Hyperlink](http://example.com)  . Footnote. [[2]](#footnote-2)

Block Text.

Table caption.

|  |  |
| --- | --- |
| Table | Table |
| 1 | 2 |

Image Caption

cross-population pattern

人机交互模式

DefinitionTerm

Definition

1. 差异性隐私将在4.2.2节规范化介绍，这里知道更小的ε表示更强的隐私性即可。 [↑](#footnote-ref-1)
2. Footnote Text. [↑](#footnote-ref-2)