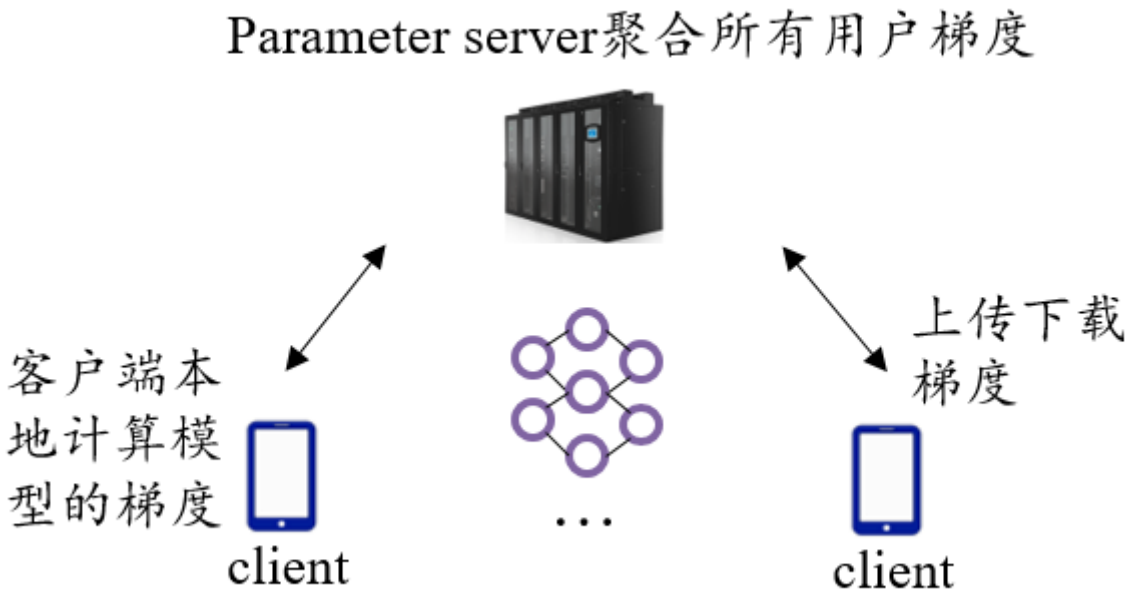
横向联邦学习下隐私保护安全聚合：问题，方法，与展望

本文总结面向横向联邦学习的主要安全聚合技术路线和经典方法，对各条技术路线所处理的问题和经典方法的核心思想做一些梳理，并提出一点个人浅见。

0. 横向联邦学习（Horizontal Federated Learning, HFL）

联邦学习是从机器学习角度出发，试图解决数据隐私保护、数据孤岛的一种分布式机器学习架构。其核心架构和分布式机器学习类似：客户端具有数据，在本地训练模型；通过参数服务器进行梯度（或者模型参数）的聚合（例如加法聚合）；最终客户端下载聚合结果在本地更新模型。



然而，和分布式机器学习相比，FL主要有以下不同：

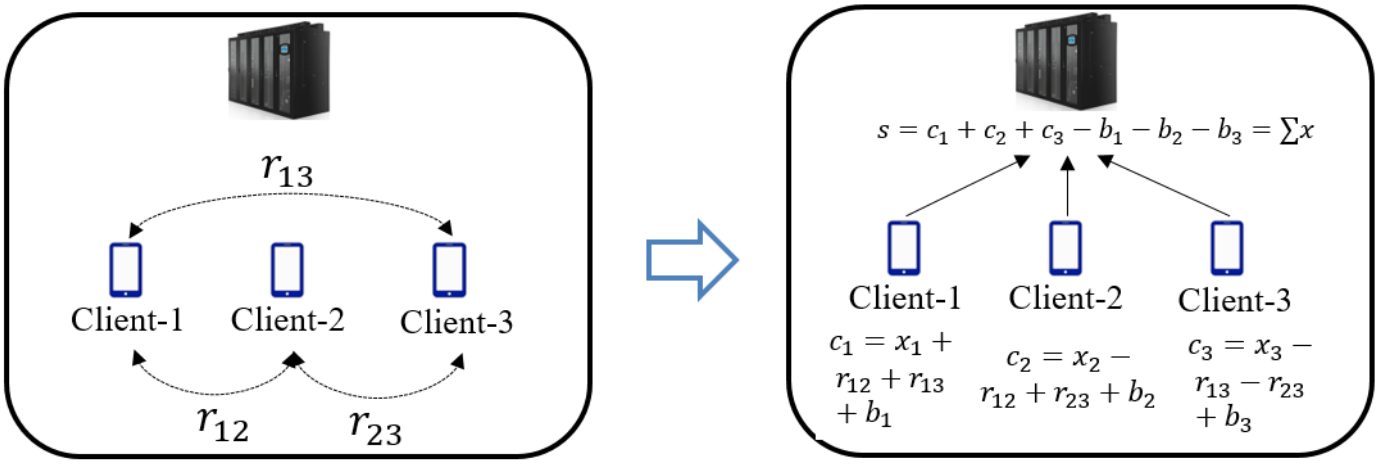
1. 分布式机器学习主要关注并行计算、快速训练，而FL更多考虑在数据隐私安全等问题前提下的效率提升；
2. 分布式机器学习下可以集中化处理数据，进而分发数据进行分布式训练；但是在FL中，由于现实场景、法律法规等原因，FL中的数据分布更加不均衡，数据处理受到隐私保护的限制也更加复杂，从而使得FL的模型训练优化更具有挑战性；
3. 分布式机器学习下计算节点比较稳定，大多在局域网内进行；FL下设备更加多样，并且难控制，面向广域网下的FL更加常见，因此FL的计算和通信资源更加受限。目前FL的场景业内公认的分类是cross-silo 和 cross-device，前者基本不需要考虑设备不稳定问题，而后者则需要认真考虑终端设备资源受限的影响。

本文针对明文HFL下存在隐私泄露的问题，主要梳理面向隐私保护的安全聚合方案，并对涉及到的其他方面的问题（例如鲁棒性）等做一些简单介绍。

1. 成对加性掩码路线

Google在CCS'17提出的基于成对加性掩码的安全聚合方案是经典方案之一，该方案基于客户端之间协商的成对加性掩码，在安全聚合中利用加法或者减法茫化真实输入。而成对加减掩码在最终的加法聚合中可以成对抵消，从而实现了安全聚合。然而，该方案主要面向cross-device问题，因此需要补充额外的技术处理设备掉线带来的影响：

1. 为了解决设备掉线带来的正确性问题，该方案额外引入了Shamir's 秘密分享技术将茫化种子安全分享到所有用户，从而所有用户构成一个全连接图。以便在线客户端恢复掉线客户端种子并将其茫化作用抵消；
2. 为了解决设备资源受限带来的隐私泄露问题，该方案引入了双盲（double-masking）技术。



该方案的具体过程和分析我们在之前的博客 Secure Aggregation有详细论述。

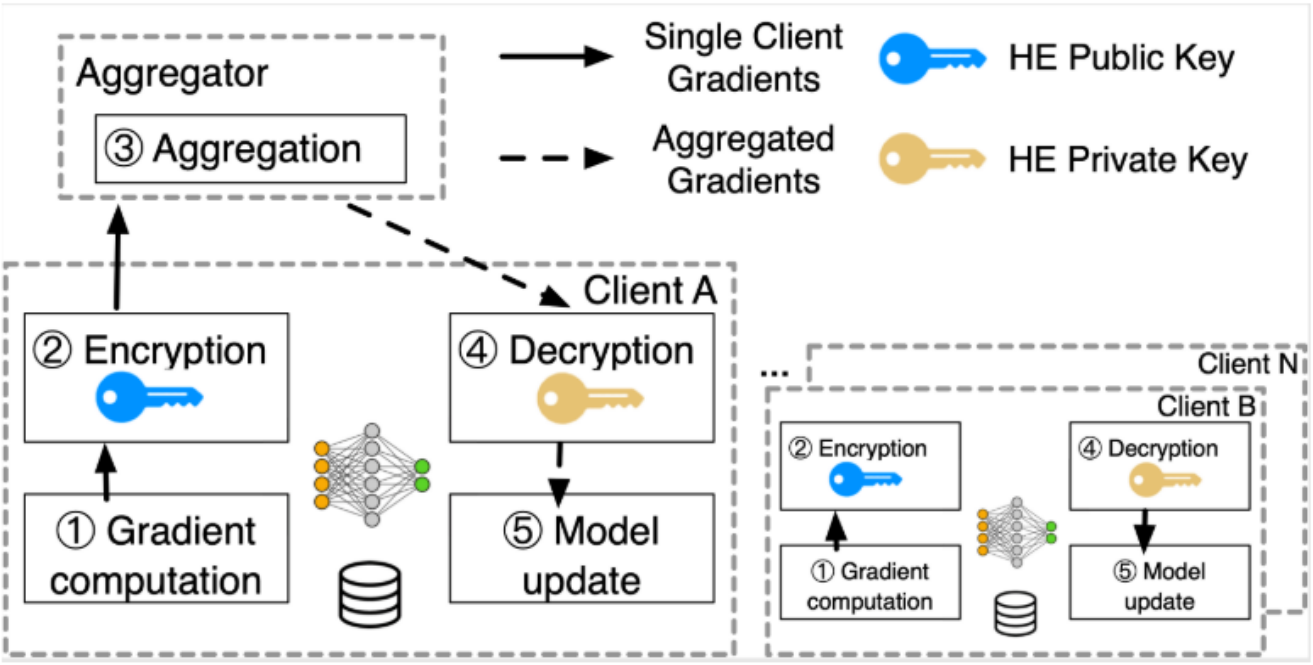
然而，该方法也存在着很大的性能问题，尤其是大规模的使用Shamir's 秘密分享带来的计算和通信开销。为了进一步优化性能，CCS'20将每一个客户端的邻居节点限制在 $O(\log_2 N)$ 以提升性能，并分析了在该情况下的隐私性等问题。该论文的详细解读可以参考博客
https://zhuanlan.zhihu.com/p/403179338
除此之外，还有许多工作在CCS'17的方案上继续展开，例如为了抵抗恶意服务器，VerifyNet 进一步引入了双线性技术等。

浅见：

基于成对加性掩码的安全聚合技术更适合轻量级客户端的安全聚合，只需要一个服务器，而且允许部分客户端和服务器合谋，安全性好。不过，该方案需要客户端-服务器之间多轮通信，而且只支持加法聚合。

2. 同态加密路线

将同态加密技术应用到FL中以保证梯度（或权重）的隐私最早是在Phong等人在TIFS‘17提出。利用同态加密的加同态性质可以很直接的实现密文下的梯度加法聚合。然而，同态加密巨大的计算开销是制约其发展的重要瓶颈。而且密文膨胀也会带来巨大的通信负载。为了提供足够的计算资源，基于同态的方案一般面向cross-silo场景。



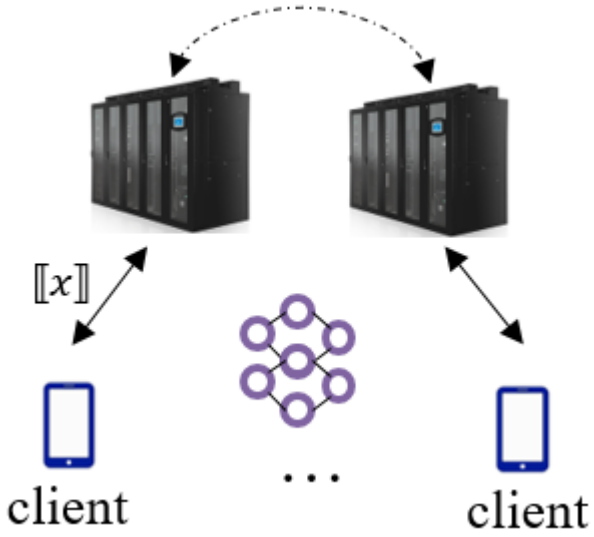
为了提升TIFS'17的方案效率，USENIX ATC'20上Zhang等人发表的BatchCrypt将量化技术和同态加密结合，从而提升同态密文批处理编码梯度的效率，进而减少加密等操作的次数和密文数量，从而提升系统性能。该方案的细节可以参考博文 https://zhuanlan.zhihu.com/p/326712188。

浅见：

基于同态加密的安全聚合技术目前受计算限制比较大，但是方案构造简单友好。不过目前该技术思路下的聚合方案还是大多面向加法聚合，并且该方案下客户端不能和服务器合谋。虽然目前也有多密钥同态方案可以借鉴，但是性能会更受限制。

3. 秘密分享路线

基于秘密分享的方案令客户端将梯度或权重以秘密分享的形式传送到多个不完全合谋的服务器上，多个服务器之间利用安全多方计算方案实现安全聚合。



和之前的技术路线相比：

1. 该技术路线理论上可以支持任意聚合函数的计算；
2. 另一个不同则是本技术路线需要多个满足合谋限制的服务器，这可能会给实用带来限制。不过，本路线也允许部分客户端和部分服务器合谋；
3. 从性能上看，基于秘密分享的方案计算轻量级（尤其是客户端计算），而且通信轮数少，架构简单。不过客户端-服务器通信量与服务器数量成正比。

最早提出该方案雏形的是Prio，之后我们做了一系列微小改进工作，相关的论文可以参考博客 https://zhuanlan.zhihu.com/p/294805865 ， https://zhuanlan.zhihu.com/p/295464454，https://zhuanlan.zhihu.com/p/416656329。
其中，前两项还是针对加法聚合的改进优化，第三项工作我们则面向针对鲁棒性聚合方案实现了隐私保护安全聚合改造。

浅见：

基于秘密分享的安全聚合能够得到较好的整体性能，但是在面对复杂聚合函数的时候开销还是很大。

4. 展望

上述梳理了一些面向横向FL的安全聚合技术方案，不同的方案各有优劣。然而，对于FL和安全聚合，我们也需要认识到：

1. 虽然每次聚合能够保护客户端的私有输入，但是聚合结果还是要公开给用户。结果的公开对于隐私的泄露是多少需要注意，之前一些研究也在探究FL中聚合结果造成的隐私泄露。
2. 除此了横向FL，面向纵向FL的安全聚合研究还比较少，虽然也有一些技术试图解决这个问题，例如PrivColl，但是性能通信等需要进一步的提升探索；
3. FL在分布式的场景下，更容易受到计算节点的恶意攻击。面向恶意攻击的高效安全聚合方案也是迫切需要的。