多方安全计算概述

A survey on Secure Multi-Party Computation

徐晨阳，李松达，张韵琪，王琬晴，章露，潘天天，杨旭晓，刘思琪，陈琪豪

华东师范大学软件工程学院

2020

華東師範大學
EAST CHINA NORMAL UNIVERSITY

Outline

从姚氏百万富翁谈起

有两个百万富翁 $Alice$, Bob，各自拥有的资产都在百万级别，记为 x, y，单位为百万

现在他们聚集在同一个舞台，攀比心切，想比较出谁更富有，并且不希望让彼此以及任意第三方知道各自的具体资产情况

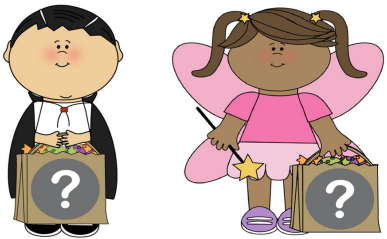

图: Yao's Millionnare

从姚氏百万富翁谈起

假设有 9 个一模一样 $(indistinguishable)$ 的盒子，盒子里藏了对应的数值 $i, 1 \leq i \leq 9$，$Alice$ 先手，在盒子里面放置苹果、橘子和梨

- 若 $Alice$ 资产大于数值，即 $x > i$，$Alice$ 就在 box_i 中放梨子
- 若 $Alice$ 资产等于数值，即 $x = i$，$Alice$ 就在 box_i 中放橘子
- 若 $Alice$ 资产小于数值，即 $x < i$，$Alice$ 就在 box_i 中放苹果

$Alice$ 放置完毕后，给每一个盒子上锁，并发给 Bob，Bob 选出与之资产对应的盒子并上锁，抛弃剩余 8 个盒子后发给 $Alice$，二人同时解锁，查看盒子里水果的情况

- 若盒子中是梨子，则 $x > y$
- 若盒子中是橘子，则 $x = y$
- 若盒子中是苹果，则 $x < y$

姚氏百万富翁的启示——两方安全计算

上述解释存在一个问题，在无可信第三方的前提下，*Alice* 上锁后，*Bob* 无法选出与之资产对应的盒子

但是这个例子却引出了两方安全计算 (*Secure Two Party Computation*) 的几个基本概念

- 不经意传输 (*Oblivious Transfer, OT*)。*Alice* 向 *Bob* 传输 n 个信息，*Bob* 选择其中一个信息 i。对 *Alice* 而言，他不知道 *Bob* 选择了哪个信息；对 *Bob* 而言，他没有获得除信息 i 之外的任何信息
- 诚实模型 (*Credit Model*)。分为完全遵守协议规定的诚实模型，遵守协议规定的同时但试图推导额外信息的半诚实模型，以及不遵守协议规定的恶意模型

进一步，姚期智 (*Andrew Yao*) 提出了基于混淆电路 (*Garbled Circuit*) 的安全计算方法，能够处理任何可计算的函数，因此两方安全计算的概念逐渐被推广到多方安全计算，更多的方法层出不穷

多方安全计算

非形式化地解释，多方安全计算 ($Secure\ Muliti\ Party\ Computation$, $SMPC$) 指，针对某一特定函数 f，在无可信第三方的情况下，多方参与计算 f 的问题，并且在计算的过程中不应当泄露任意一方的隐私，可以初步理解为安全的分布式计算

SMPC 实例

例如统计医院病人患癌总数，在不泄露个人患癌信息的前提下进行安全计算，用 1，0 表示每个人患癌与否，即安全计算

$$f(a_1,\ a_2,\ ..,\ a_n) = \sum_{i=1}^{n} a_i$$

这样的不透露个人隐私的数据可以拿来做模型的预测与训练，达成学术目的的同时，也避免了社会的舆论

接下来我们介绍多方安全计算领域，几个常见的计算方法与协议，他们的应用场景各有千秋

秘密分享 *Secret Sharing*

秘密分享 (*Secret Sharing*) 指的是将秘密 (*secret*) 分布在一群参与者中，其中每个参与者拥有一个秘密碎片 (*share*)

若要重构秘密，则需要聚集一定数量的秘密碎片，少于这个数量都无法重构秘密，显然单个秘密碎片没有任何意义

从一个故事谈 *Secret Sharing*

探险家们被困沙漠，物资匮乏，补给所剩无几，大家经过商讨后决定将剩下的食物与水全部放入一个保险箱中，并且每人每天需要出去搜索获得资源。

但是问题来了，如何分配打开保险箱的钥匙？

图: A story related to Secret Sharing

从一个故事谈 *Secret Sharing*

若交由一人保管，如果他带着物资跑路，剩下的所有人都将遇难。

若交给部分人保管，且他们全部到齐才能打开补给箱，如果有一人丢掉钥匙，所有人都将遇难。

比较好的办法是交给部分人保管，且部分人到齐才能打开补给箱。

从一个故事谈 *Secret Sharing*

我们对上面的故事进行抽象

给定秘密 M，将其划分为 n 个 $share$，设定一个阈值 t 满足 $1 < t < n$

只需要凑齐 n 个 $share$ 里面的 t 个，我们便可以重构秘密 M，这便是 $Secret\ Sharing$ 的思想

Secret Sharing 的简单实现

给定秘密 S_0，生成 $t-1$ 个随机数 $S_1,\ S_2,\ ..,\ S_{t-1}$

构造 $t-1$ 次常系数多项式，将秘密 S_0 藏在其中

$$f(x) = \sum_{i=0}^{t-q} S_i x^i = S_0 + S_1 x + .. + S_t x^{t-1}$$

生成 n 个随机数 x_i，并且得到对应的 $y_i = f(x_i)$，将 n 个数对 $(x_i,\ y_i)$ 分发给 n 个人，只要凑齐其中 t 个人便可以重构多项式，从而得到秘密 $S_0 = f(0)$

Secret Sharing 的简单实现

例如使用拉格朗日插值 $(Lagrangian\ interpolation)$，对每一个数对 $(x_i,\ y_i)$ 构造插值基函数

$$\ell(x) = \prod_{i=0,\ i\neq j}^{t-1} \frac{x - x_i}{x_j - x_i} = \frac{x - x_0}{x_j - x_0} \cdot \frac{x - x_1}{x_j - x_1} \cdots \frac{x - x_{t-1}}{x_j - x_{t-1}}$$

观察知道

$$\ell(x) = \begin{cases} 1 & x = x_j \\ 0 & x \neq x_j \end{cases}$$

进而构造拉格朗日多项式

$$L(x) = \sum_{j=0}^{t-1} y_j \ell_j(x)$$

事实上，$L(x) = f(x)$，因此直接计算 $L(0)$ 即可，利用拉格朗日插值恢复秘密的时间复杂度是 $O(t^2)$

Shamir's Scheme

在整数域 $\mathbb{Z}$ 上构造多项式仍然可以被破解，*Shamir's Scheme* 提出了在素数域 $\mathbb{Z}_p$ 上构造，给出了更高的安全保证

有兴趣同学在学习了抽象代数的基本知识后可以进一步学习参考资料，我们在此不再赘述

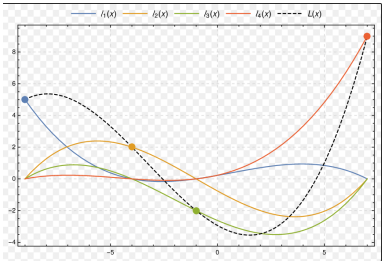

图: Lagrange Interpolation In $\mathbb{Z}$

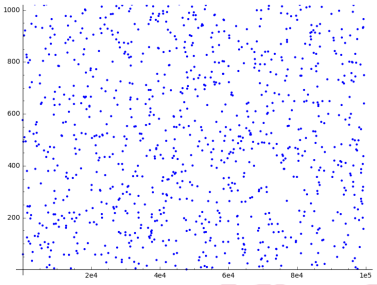

图: Lagrange Interpolation In $\mathbb{Z}_p$

同态加密 *Homomorphic Encryption*

回想一下数学上的同态 $(Homomorphism)$，简单解释就是

$$f(x \cdot y) = f(x) \cdot f(y),\ f(x + y) = f(x) + f(y)$$

非形式化地解释同态加密 $(Homomorphic\ Encryption)$，将密文拆散之后加密再合成，等价于直接对密文加密

同态加密的用武之地

考虑如下场景，多方共同传输信息，需要将信息全部传输到公共服务器，合成后统一进行加密，如果服务器被攻击，那么多方隐私都会被泄露

因此，考虑多方选择合适且支持同态的加密方案，各自加密信息，将密文传送到服务器上再合成，便可以有效的降低隐私泄露的风险

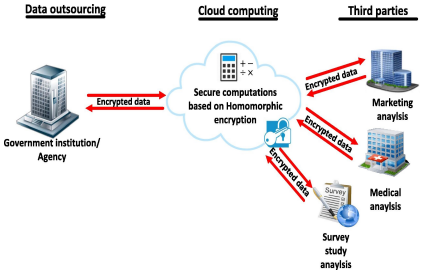

图: Homomorphic Encryption

零知识证明 *Zero Knowledge Proof*

非形式化地解释，考虑一个事件有两位参与者 A，B，其中 A 掌握了秘密 C

若 A 能够在不透露任何信息 M 的前提下，向 B 证明自己掌握了秘密 C，我们就把这个证明过程称为**零知识证明** (*Zero Knowledge Proof*, *ZKP*)

图: Zero-Knowledge Proof

我们为何需要 ZKP?

传统数学证明中，我们通过基本的定理推导出结论，一步一步搭建理论大厦

问题在于，我们证明结论的过程可能揭示了一定的信息

例如 A 向 B 证明大整数 n 是合数，一种显然的证明方式是给出具体的 p，q 使得 $n = pq$

又例如 A，B 是好友，却又都爱上了 C，A 想要告诉 B 自己恋爱了，显然的方式是告诉 B 自己爱上了 C

所以，有时候 A 期望向 B 证明一个结论，却并不希望告诉 B 其他信息，因此逐步发展出了零知识证明的概念

ZKP 实例

E.g.1 大整数分解 $(Big\ Integer\ Factorization)$

Desc. 给定大整数 $n = pq$，其中 p，q 是两个素数，考虑 A 知道 p，q，如何向 B 零知识证明自己能够将 n 大整数分解？

Sol. B 随机选择一个整数 x，计算

$$y \equiv x^4 \ (mod\ n)$$

并将 y 发送给 A，若 A 知晓 n 的分解情况，便可以利用中国剩余定理计算出

$$z \equiv x^2 \ (mod\ n)$$

并将 z 发送给 B，验证是否满足

$$z^2 \equiv y \ (mod\ n)$$

此过程重复多次，直到 B 相信 A 可以对 n 进行大整数分解，此过程素因子 p，q 未被泄露

ZKP 实例

E.g.2 洞穴问题 (*Cave Problem*)

Desc. 给定地图如下，*Bob* 知晓门的密码并向 *Alice* 零知识证明

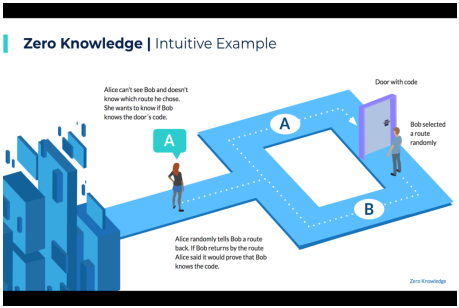

图: Cave Problem

ZKP 实例

Sol. Bob 向 $Alice$ 进行多次游说，每次游说 $Alice$ 站在分岔口，Bob 以 $\frac{1}{2}$ 概率随机站在门的 A，B 口，游说生成时保证 $Alice$ 看不到 Bob 的位置

$Alice$ 指定一个出口 A 或者 B，若 B 知晓门的密码，便可以开门或者不开门地准确走到指定出口

此过程重复多次，直到 $Alice$ 相信 Bob 知晓门的密码，且此过程密码没有被泄露

ZKP 应用

身份认证方案 $(Identification\ Scheme)$

考虑如下场景

为了实验室安全，仅允许实验室成员拥有门禁

比较容易想到的方案是将每位成员的编号 ID 添加到一个集合，每次刷卡时验证刷卡人 ID 是否属于这个集合

不幸的是，门禁盒暴露在外，考虑敌手 $(adversary)$ 以特殊手段获得内存信息，那么持卡人的信息暴露在外，从而也便于敌手进行攻击

考虑换一种身份认证方案，为门禁维护大整数 n，给每位持卡人私钥 p，q，使得 $n = pq$，每次持卡人只需要向门禁零知识证明 $n = pq$ 即可获得访问权限，敌手对内存的攻击也仅仅知道 n，并不知道具体的分解方式

ZKP 应用

ZKP 其他的经典应用聚焦在

- 数据隐私保护
- 计算压缩与区块链扩容
- 端到端通讯加密

对此内容感兴趣的同学可以查询资料做进一步研究，我们在此不再赘述

从差分攻击谈差分隐私

首先介绍一下差分攻击，考虑如下统计数据库，用 a_i 表示第 i 个人是否单身，用 1 表示单身，用 0 表示非单身

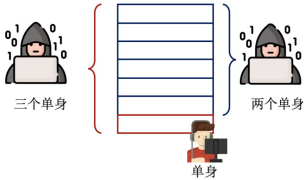

图: Differential Attack

敌手只需要对前缀和做一个差分，就可以知道第 x 个人是否单身，这样一来，第 x 个人的隐私暴露无遗

$$a_x = \sum_{i=1}^{x} a_i - \sum_{i=1}^{x-1} a_i$$

差分隐私 *Differential Privacy*

非形式化地解释，差分隐私 (*Differential Privacy*) 是一种旨在最大化统计数据库的查询准确度的同时，最小化记录的识别的密码学技术

为了阐明差分隐私技术，我们引入相邻数据集的概念

给定数据集 D, D'，两个数据集满足只有一个记录不同，我们就称之为相邻数据集

形式化地表示差分隐私，即从 D, D' 中得到的输出的分布尽可能相似

$$Pr\{\mathcal{D}(x) = O\} \leq e^{\epsilon} Pr\{\mathcal{D}'(x) = O\}$$

用噪声实现差分隐私

为了避免敌手轻易拿到信息，我们在源数据集上添加噪声 $(noise)$，使得对数据集的查询趋向于一个分布

即使有一条记录被修改，对新数据集查询的分布仍然类似于源数据集，所以攻击者很难从差分攻击中得到信息

常用的噪音有拉普拉斯噪声 $(Laplace\ Noise)$ 与高斯噪声 $(Gaussian\ Noise)$，后面的讨论中有详细的介绍，我们在此不再赘述

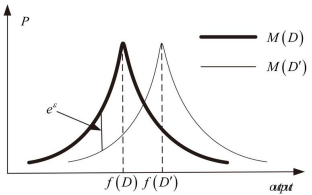

图: Differential Prvacy

多方安全计算的应用

至此，我们介绍了多个多方安全计算的方法。

多方安全计算适用于大型统计问题与模型训练，凡是涉及到多方大数据的计算，并且在意隐私，多方安全计算便可以大显身手，例如安全拍卖 (*Secure auctions*)，安全电子选举 (*Secure electronic voting*)，安全机器学习 (*Secure machine learning*)

作为密码学研究的一个子领域，新的安全计算方法、安全协议层出不穷，应用场景各有千秋，同时也出现了更多的方向等着人们去探索、挖掘，无穷的远方，无数的人们，都与我们相关。

Ending

感谢聆听!