可用不可见
BLINDFOLDED COMPUTING
| DataFun.

可证明安全的隐私计算

洪澄 阿里安全双子座实验室

目录 CONTENT

隐私计算：Privacy-preserving computation

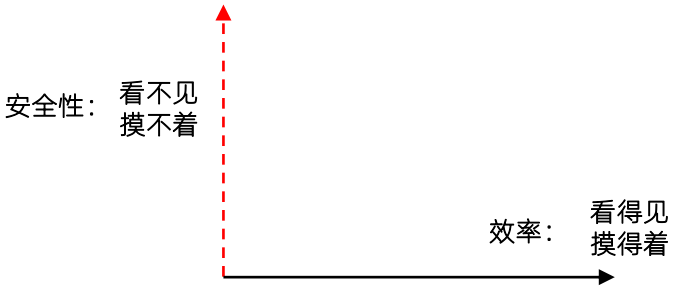
- 视隐私保护程度的不同，计算的效率也有所不同
- 例：广域网下，多方合作训练一个LR模型，1024行64列，迭代1次
  - SecureML[SP17]需要~100秒
  - ABY3[CCS18]需要0.3秒
  - Blaze[NDSS20]需要2秒
  - Helen[SP20]需要~15分钟

Q：“请问现在隐私计算能做到比明文慢多少倍？”

A：

- 只有同时给出安全性和效率两方面的数据，才是有意义的数据
- 业界现状：
  - 计算的效率是容易衡量的，各家PR都擅长此道
  - 安全指标则甚少有厂商主动提及
  - 客户有如盲人摸象

“比明文仅慢X倍”，X已经从3个量级来到2个量级，甚至3-5倍都有之



- 最高级安全性的代价太高
  - 不需要最高级安全性的场合，可以适当降低安全性以提升效率
  - 但是一定要厘清安全性在哪里进行了取舍，有什么样的风险
- 如何讲清楚一个方案的安全性？
  - 明确定义安全假设：能防什么样的攻击者，不能防什么样的攻击者
  - 明确定义防护效果：有没有中间信息泄露
    - 若有，应清晰的描述泄露内容

“我的方案是安全的”

×

“我的方案在双方都是半诚实的假设下，除了数据行数、列数、最终建模结果之外，没有其他信息泄露”

√

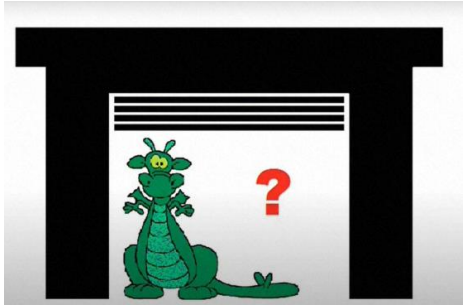
- 反例：Dragon in my garage

我的仓库里有一条喷火龙

Q：你的龙为什么看不见？
A：因为它是透明的

Q：你的龙为什么没有脚印？
A：因为它是飞着的

Q：你的龙为什么摸不着？
A：因为它是等离子态的



我有一个隐私计算解决方案

Q：你的方案可能会泄露XX统计信息？
A：我们认为XX统计信息不影响方案的安全性

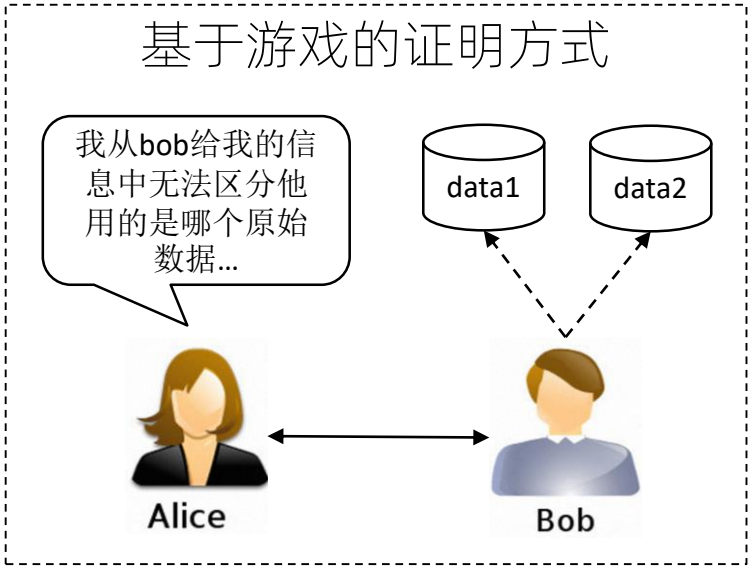
Q：你的方案需要额外的第三方？
A：我们认为只要严格审计是可以接受第三方的

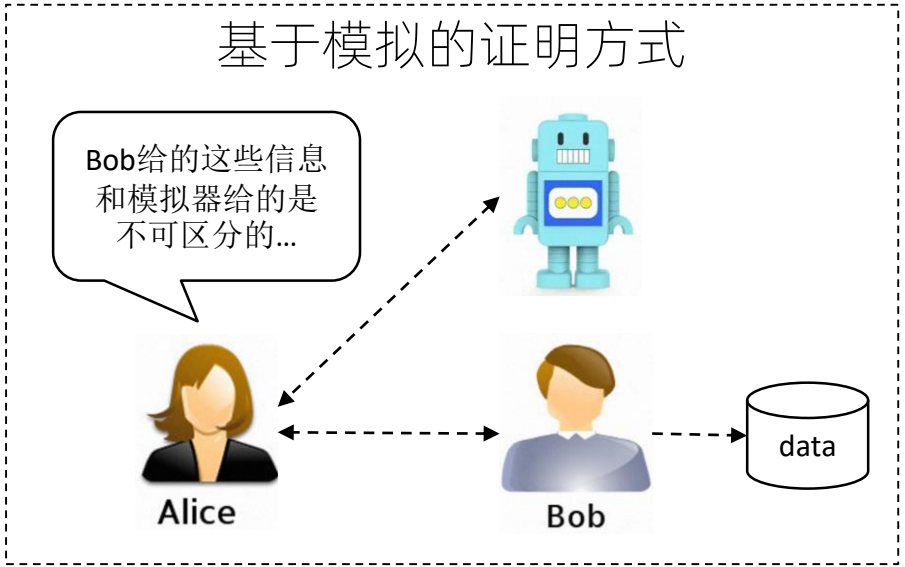
Q：你们的自研算法有严格的安全证明吗？
A：有专利，论文还在投稿中

- 安全性需要正向定义：
  - 需要描述“龙”到底能在什么环境下做到什么，才有办法证明它的存在

目录 CONTENT

- 可证明安全：密码学领域评估安全性的黄金准则
- 两种安全证明方式

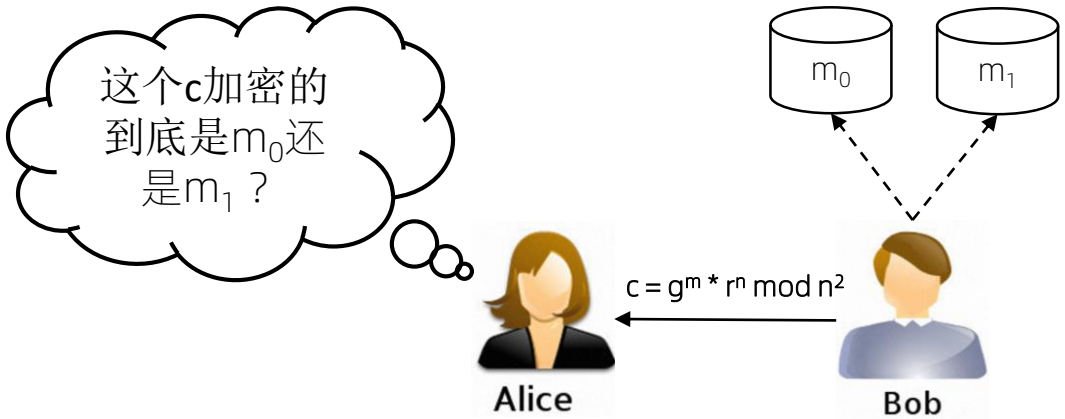




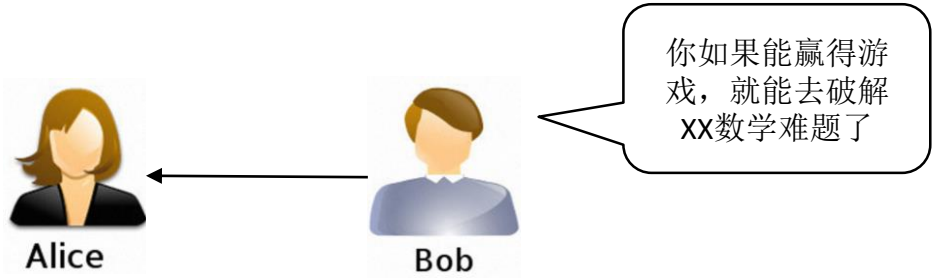
- 基于游戏的证明方式举例：Paillier
  - Alice选择 m_0 , m_1
  - Bob选择 $c \in \{ Enc(m_0), Enc(m_1) \}$
  - Alice猜测c的明文，猜对则赢得游戏

密钥生成：
- 生成两个大质数p,q, n=p*q , λ=lcm(p-1,q-1), g=n+1
- g, n是公钥，λ是私钥

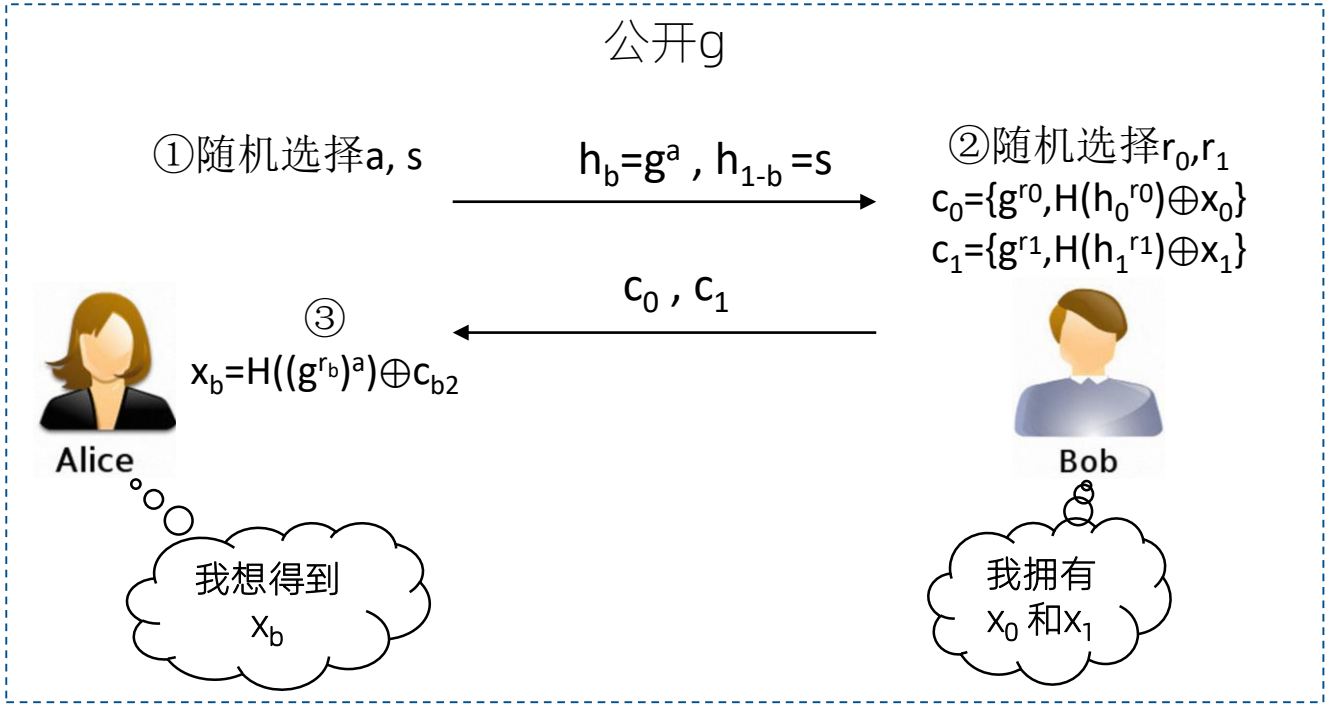
加密m
- 选择随机数r, 计算$c = g^m * r^n \bmod n^2$



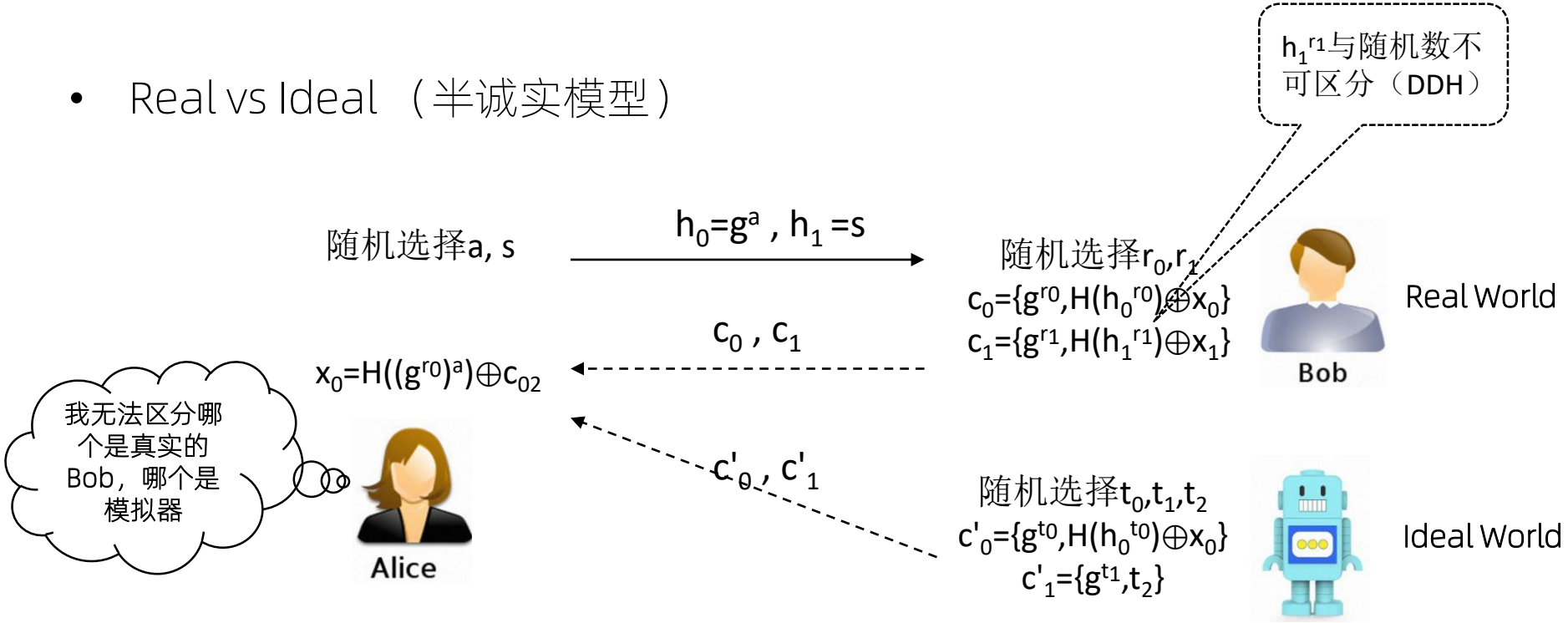
- 反证法：假设Alice能以不可忽略的优势(>50%的概率)赢得游戏
  - 设Alice能成功判断出c是m_0的密文，
  - 因为$d = c*g^{-m_0} = r^n \bmod n^2$，所以她也能判断出d是一个n次幂
  - 而判别一个数是不是mod n^2 上的n次幂，这个问题称为DCR问题（decisional composite residuosity problem），目前认为是困难的，与大数分解接近
  - 矛盾，证毕



- 基于模拟的证明方式举例：OT



- Real vs Ideal （半诚实模型）



- 恶意模型下，基于模拟的证明更加复杂

We begin by proving that conditioned on $\mathcal{S}'$ not outputting $\perp$, it generates output that is identically distributed to V^*'s output in a real proof. That is, for every V^*, every $(G,\psi) \in R_L$ and every $z \in \{0,1\}^*$,

$$\left\{\mathsf{output}_{V^*}(P(G,\psi), V^*(G,z))\right\} \equiv \left\{\mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi) \mid \mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi) \neq \perp\right\}. \quad (5.1)$$

In order to see this, observe that the distribution over the commitments viewed by V^* is identical to a real proof (since they are commitments to a random permutation of a valid coloring). The only difference is that $\mathcal{S}'$ chooses an edge e ahead of time and only concludes an iteration if the query sent by V^* equals e. However, since e is chosen uniformly every time, and since V^* is rewound to the beginning of each iteration until it succeeds (and we condition on it indeed succeeding), these have identical distributions.

Next, we prove that $\mathcal{S}'$ outputs $\perp$ with at most negligible probability. Observe that the commitments provided by $\mathcal{S}'$ reveal no information whatsoever about the choice of e in that iteration (this is due to the fact that the commitments are the same for *every* choice of e). Thus, the probability that a single iteration succeeds is exactly $1/|E|$, implying that $\mathcal{S}'$ outputs $\perp$ for one of the i's in the simulation with probability $\left(1-\frac{1}{|E|}\right)^{n\cdot|E|} < e^{-n}$. There are $n\cdot|E|$ iterations, and so by the union bound, $\mathcal{S}'$ outputs $\perp$ somewhere in the simulation with probability less than $n\cdot|E|\cdot e^{-n}$, which is negligible. This implies that[2]

$$\left\{\mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi) \mid \mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi) \neq \perp\right\} \stackrel{c}{\equiv} \left\{\mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi)\right\}. \quad (5.2)$$

Finally, we prove that the outputs of $\mathcal{S}$ and $\mathcal{S}'$ are computationally indistinguishable:

$$\left\{\mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi)\right\} \stackrel{c}{\equiv} \left\{\mathcal{S}^{V^*(G,z,r_\cdot)}(G)\right\}. \quad (5.3)$$

Intuitively, we prove this via a reduction to the security of the commitment scheme. Specifically, assume by contradiction, that there exists a probabilistic-polynomial time verifier V^*, a probabilistic-polynomial time distinguisher D, and a polynomial $p(\cdot)$ such that for an infinite sequence (G,ψ,z) where $(G,\psi) \in R$ and $z \in \{0,1\}^*$,

$$\left|\Pr\left[D\left(G,\psi,z,\mathcal{S}'^{V^*(G,z,r_\cdot)}(G,\psi)\right)=1\right] - \Pr\left[D\left(G,\psi,z,\mathcal{S}^{V^*(G,z,r_\cdot)}(G)\right)=1\right]\right| \geq \frac{1}{p(n)},$$

where n denotes the number of nodes in G, and R denotes the 3-coloring relation. Without loss of generality, assume that D outputs 1 with higher probability when it receives the output of $\mathcal{S}'$ than when it receives the output of $\mathcal{S}$. We construct a non-uniform probabilistic polynomial-time adversary $\mathcal{A}$ for the commitment experiment $\mathsf{LR\text{-}commit}$ as defined in Section 5.2. Adversary $\mathcal{A}$ receives (G,ψ,z) on its advice tape (for n, where G has n nodes), and works as follows:

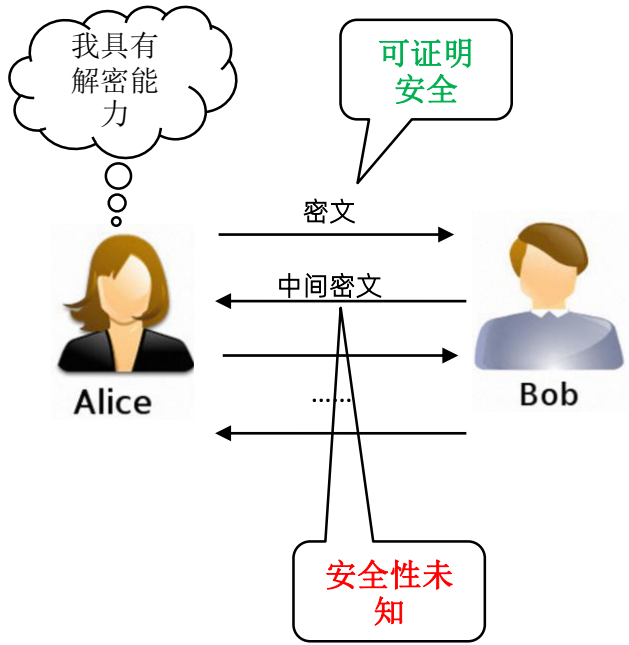
1. $\mathcal{A}$ initializes V^* with input graph G, auxiliary input z and a uniform random tape r.

如何证明方案的安全性?

- Step 1：证明每个底层模块的安全性
  - 只有每个模块都安全，才可以讨论整体方案的安全
- Step 2：判断模块的运行方式
  - 模块之间是串行运行：方案满足可证明安全
    - 因为证明是sequential composable的
  - 模块之间不是串行运行：则还需要相关模块满足UC（universal composable）特性
    - 一般的PPML任务可认为是串行的，不必考虑UC问题

一些错误的打开方式

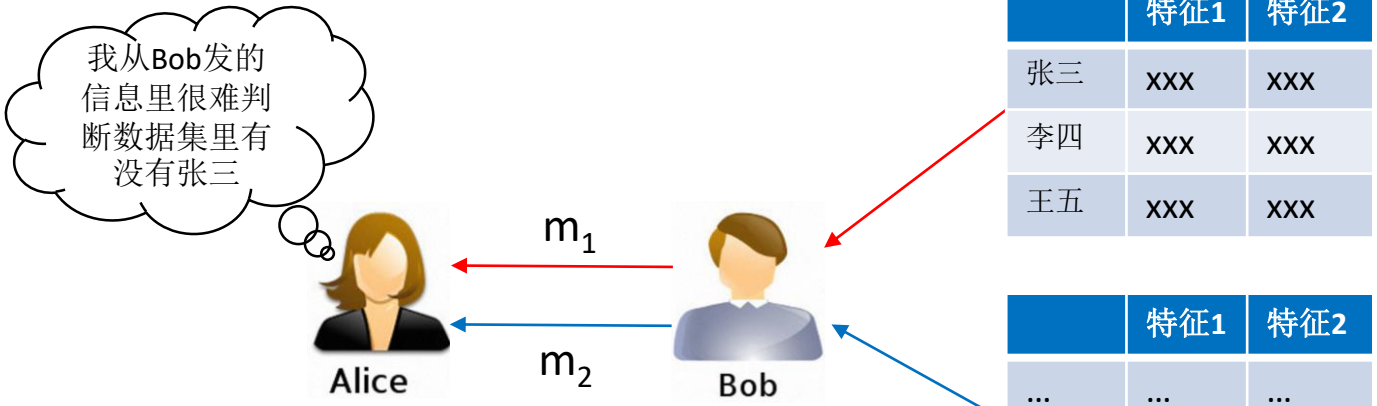
- ~~算法在xxx步骤使用了Paillier同态加密，Paillier可证明安全，所以算法是安全的~~
  - 需要算法中所有的模块都是可证明安全
    - 例：某个模块直接把中间结果发回去解密
- ~~只要不能反推原始数据，就是安全的~~
  - 有的泄露一开始认为不可反推，后来发现可反推
    - 例：Deep leakage from gradients
  - 有的泄露是原始数据的一个函数约束
    - 虽然不可直接反推原始数据，但可以间接反推
    - 例：泄露了张三的年龄+工资 = 25000



目录 CONTENT

- Q：这些证明太难学了，还有别的方法可以证明隐私计算方案的安全性吗？
- A：
  - 好消息：有别的方法
  - 坏消息：也很难

- 差分隐私 Differential Privacy
  - Alice从Bob的信息中难以知晓任意指定行的信息
  - 推论：Bob数据集里所有的行都得到了保护



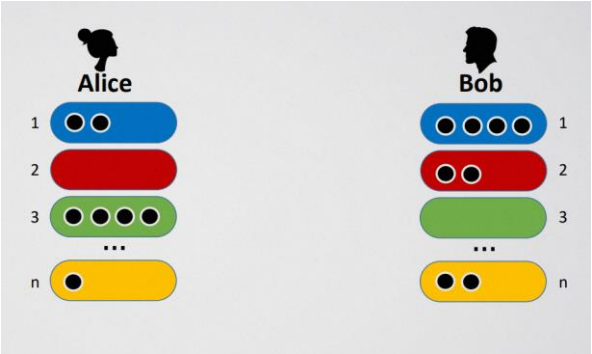
m_1和m_2的统计分布非常接近：

$Pr[m_1=x] \le e^{\varepsilon} * Pr[m_2=x]$

- 例：DP-SGD
  - Clip, aggregate, then add noise
  - Noise值与每条记录算得梯度的最大值（args.max_grad_norm）有关
  - TensorFlow Privacy集成了相关算法
  - 可以容易的用于横向分割的联邦学习

```
for batch in Dataloader(train_dataset, batch_size=32):
    for param in model.parameters():
        param.accumulated_grads = []

    # Run the microbatches
    for sample in batch:
        x, y = sample
        y_hat = model(x)
        loss = criterion(y_hat, y)
        loss.backward()

        # Clip each parameter's per-sample gradient
        for param in model.parameters():
            per_sample_grad = p.grad.detach().clone()
            clip_grad_norm_(per_sample_grad, max_norm=args.max_grad_norm)  # in-place
            param.accumulated_grads.append(per_sample_grad)

    # Aggregate back
    for param in model.parameters():
        param.grad = torch.stack(param.accumulated_grads, dim=0)

    # Now we are ready to update and add noise!
    for param in model.parameters():
        param = param - args.lr * param.grad
        param += torch.normal(mean=0, std=args.noise_multiplier * args.max_grad_norm)
```

- DP不仅可以用在FL，也可以用在MPC
- 例：[1] 使用DP保护PSI中的桶内元素数目，以降低padding，提高PSI性能



[1]: Cheaper Private Set Intersection via Differentially Private Leakage, PETS19

DP挑战1：DP会大幅影响数据分析的准确率

No DP	10 epochs	10 epochs	30 epochs	60 epochs	90 epochs
	tuned LR	learning rate = 0.4			
Resnet-18	58.7%	57.6%	67.5%	70.3%	70.8%
Resnet-50	62.1%	60.5%	72.0%	74.5%	75.3%

DP	privacy loss bound ε							
	4.6	13.2	71	$\approx 10^7$	10^9	10^{11}	10^{13}	10^{15}
Resnet-18	3.7%	6.9%	11.3%	45.7%	55.4%	56.0%	56.3%	56.4%
Resnet-50	2.4%	5.0%	7.7%	44.3%	58.8%	57.8%	58.2%	58.6%

Table 4: Comparison of the best Resnet-18 and Resnet-50 top-1 accuracies obtained at 10 epochs and batch size 1024, for various values of the privacy loss bound ε

Google: [Toward Training at ImageNet Scale with Differential Privacy]

DP挑战2：目前DP+隐私计算的研究集中在横向分割

- 纵向分割方面的研究不多
  - 而纵向是国内隐私计算的主流应用场景
  - 有待从业者投入研究

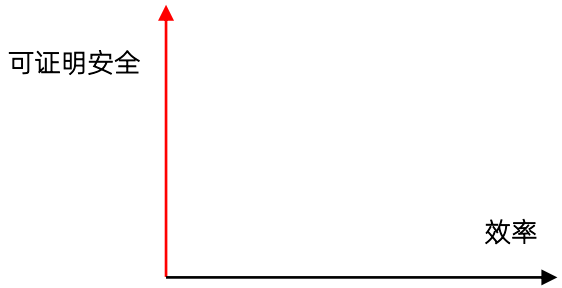
	特征1	特征2
张三	xxx	xxx
李四	xxx	xxx
王五	xxx	xxx

	特征3	特征4
张三	xxx	xxx
李四	xxx	xxx
王五	xxx	xxx

目录 CONTENT

总结：隐私计算领域公认的两种可证明安全方法

- 基于游戏/模拟的证明方式
  - 目标是刻画信息的泄露边界
- 基于差分隐私的证明方式
  - 目标是防止信息重识别到单条记录
- 呼吁隐私计算业界做好安全证明，让方案的安全性看得见，摸得着



THANKS

可用不可见 BLINDFOLDED COMPUTING | DataFun.