数学之美2022季

零知识证明-初探零知识证明

深圳市元宇元宙智能科技有限公司

01

初探零知识证明

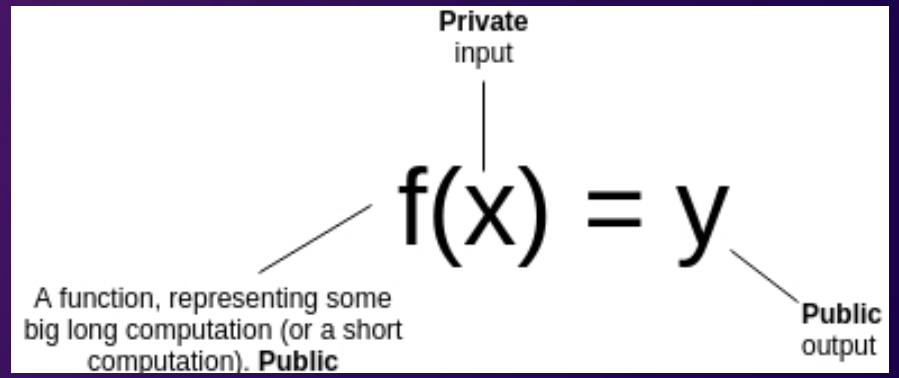
从头构造零知识证明

零知识证明并不神秘

零知识证明最早由MIT的Shafi Goldwasser和Silvio Micali在1985年一篇名为《互动式证明系统的知识复杂性》的论文中提出。作者在论文中提到，证明者（prover）有可能在不透露具体数据的情况下让验证者（verifier）相信数据的真实性。零知识证明可以是交互式的，即证明者面对每个验证者都要证明一次数据的真实性；也可以是非交互式的，即证明者创建一份证明，任何使用这份证明的人都可以进行验证。

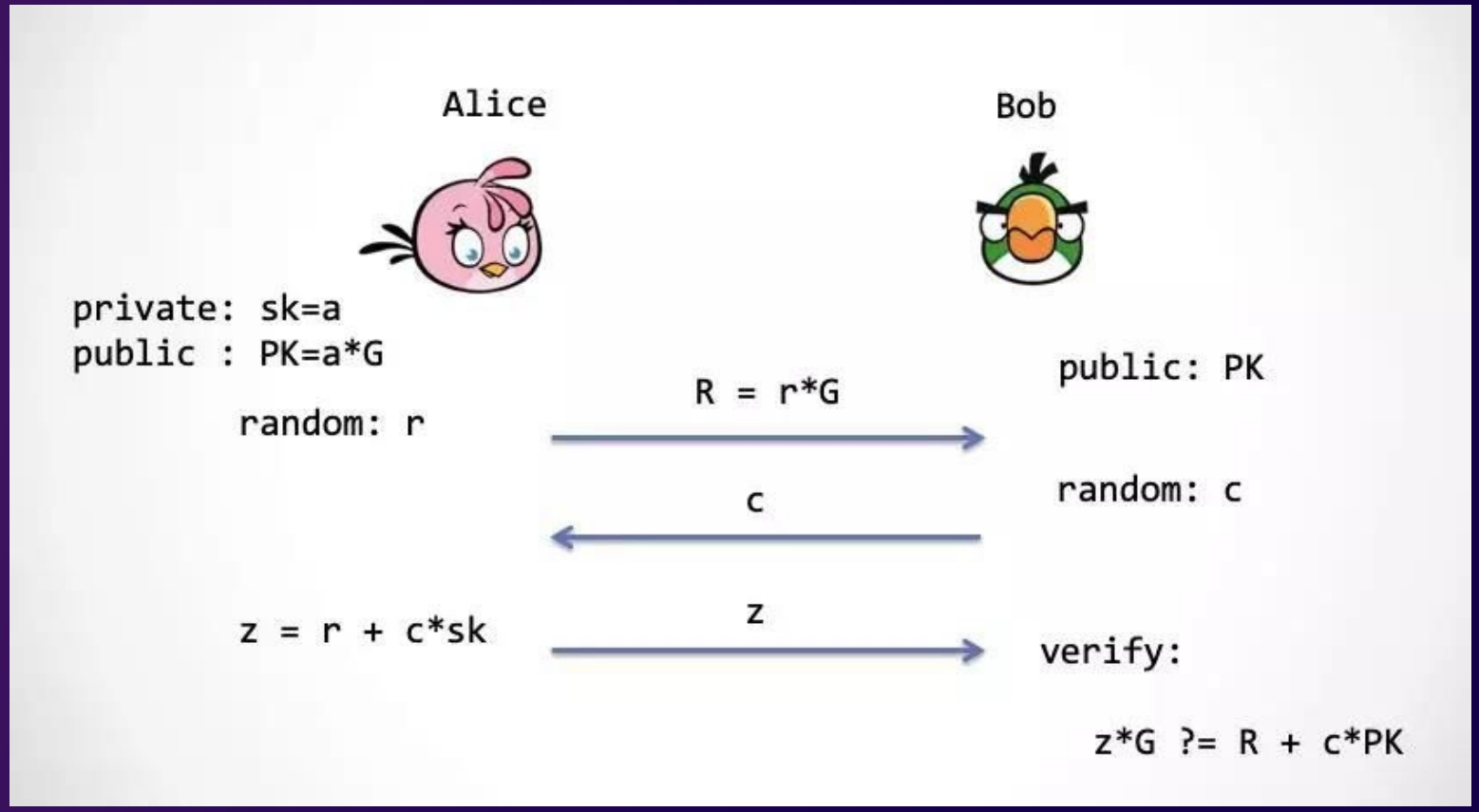
零知识证明有三个基本特征，即：

- **完整性**：如果statement为true，则诚实的验证者可以相信诚实的证明者确实拥有正确的信息。
- **可靠性**：如果statement为false，则任何不诚实的证明者都无法说服诚实的验证者相信他拥有正确的信息。
- **零知识性**：如果statement为true，则验证者除了从证明者那里得知statement为true以外，什么都不知道。



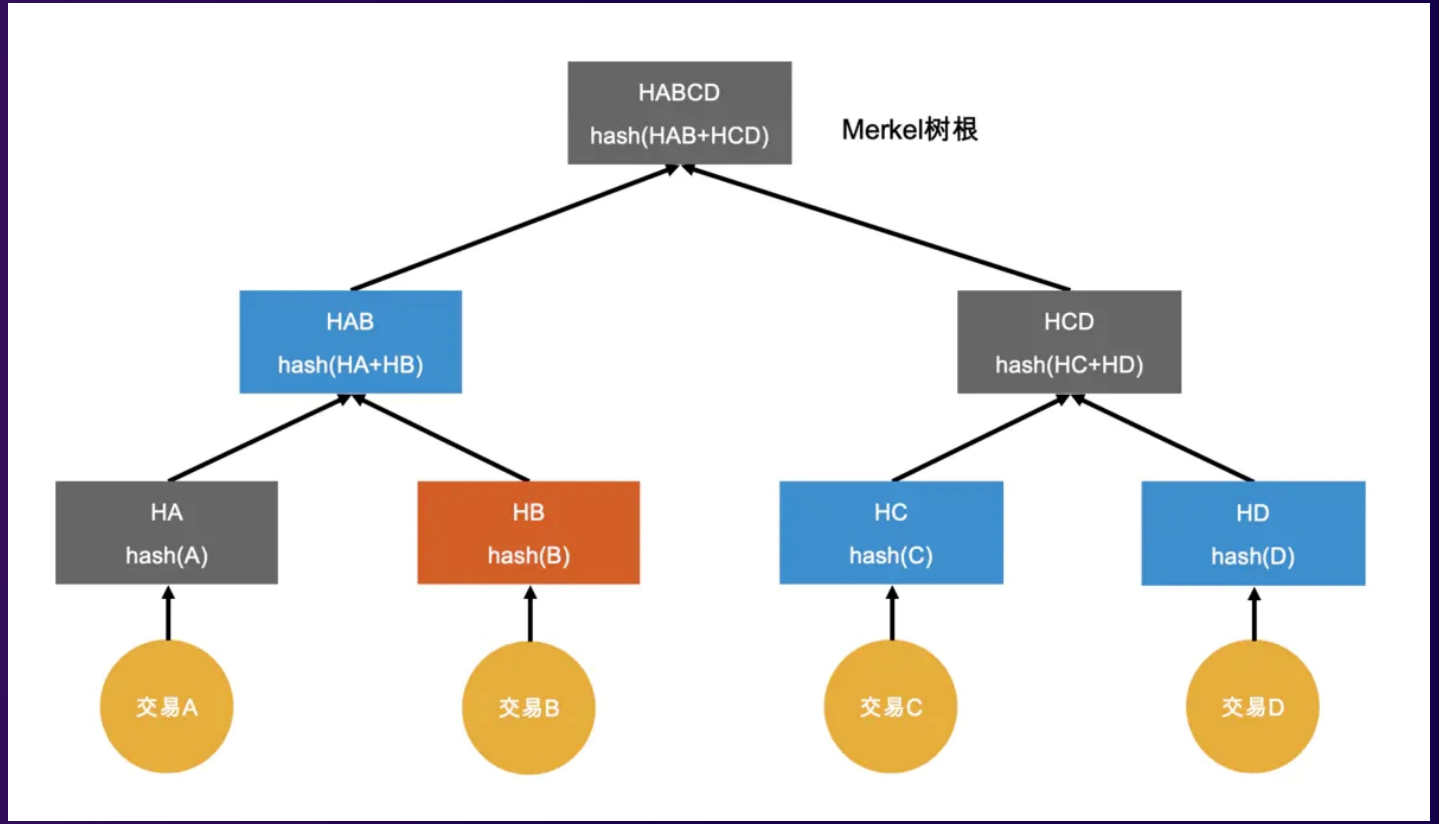
零知识证明-schnorr签名

Schnorr签名已经是一个零知识案例



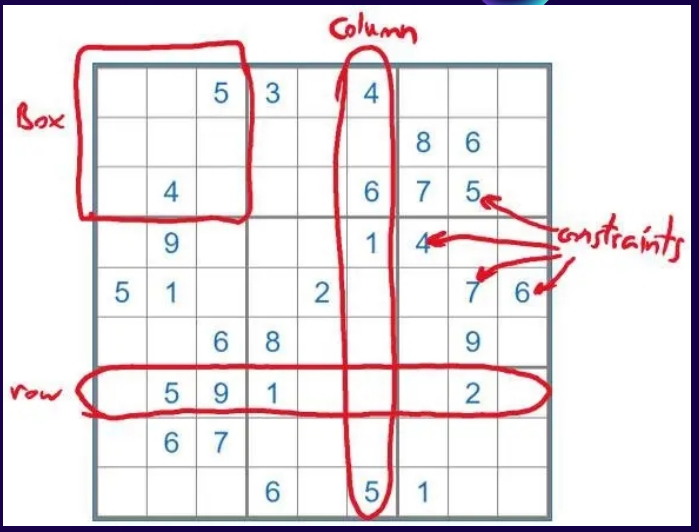
零知识证明-区块链merkle树

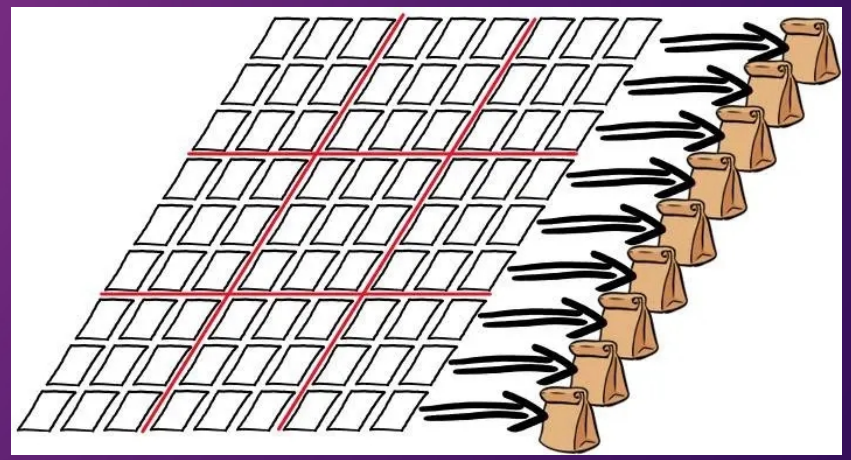
只需要构造如图所示的Merkel树，公布HA，HCD和Root(HABCD)。交易B的拥有者通过验证生成的Root是否与公布的一致，即可证明交易B是否被包含在该区块中，整个过程无需知道其他交易的真实内容。



零知识证明-数独案例

证明prove：
Alice 拿出 81 （9x9）张空白的卡片，并在每张纸上写上 1-9 中的一个数字，接着她将代表谜面的卡片数字面朝上、代表谜底的卡片数字面朝下都放在桌上，并组成了 9x9 的矩阵。

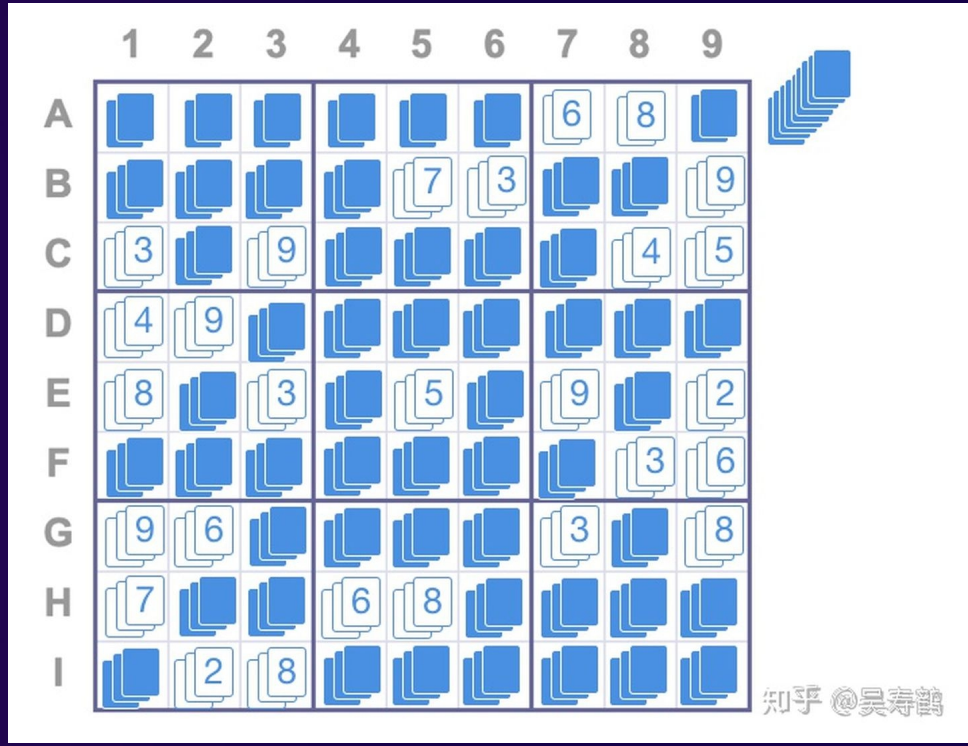




验证verify：
由 Bob 随机选择行、列或是粗线宫中的一种进行验证，假如选择行，则将这 81 张卡片按 9 行分别放到 9 个麻布袋中，摇匀并确保卡片次序打乱。

零知识证明-数独案例改进版非交互

证明prove：
Alice 拿出 81 （9x9）张空白的卡片，并在每张纸上写上 1-9 中的一个数字，接着她将代表谜面的卡片数字面朝上、代表谜底的卡片数字面朝下都放在桌上，并组成了 9x9 的矩阵。 每个单元格放三张，组成27个袋子



Bob向机器获取证明，机器返回给Bob27个袋子：
•机器将数独中每一行9张卡片取出，并分别混淆后放入一个袋子中，一共有9行，所以9个袋子
•机器将数独中每一列9张卡片取出，并分别混淆后放入一个袋子中，一共有9列，所以9个袋子
•机器将数独中每个粗线宫（3*3）内卡片取出，并分别混淆后放入一个袋子中，一共有9个，所以9个袋子

万物皆可零知识证明

零知识证明技术可以「模拟」出一个第三方，来保证某一个论断是可信的
换句话说，当我们收到一个加了密的数据，然后还有一个零知识证明。这个零知识证明是说「关于数据的 X 断言成立」，那么这等价于有一个天使在我们耳边悄声说，「关于数据的X 断言成立」！

对于这个X 断言，可以非常灵活，它可以是一个NP复杂度的算法。大白话讲只要我们能写一段程序（一个多项式时间的算法）来判断一个数据是否满足X断言，那么这个断言就可以用零知识证明的方式来表达。通俗点讲，只要数据判定是客观的，那么就零知识证明就适用。

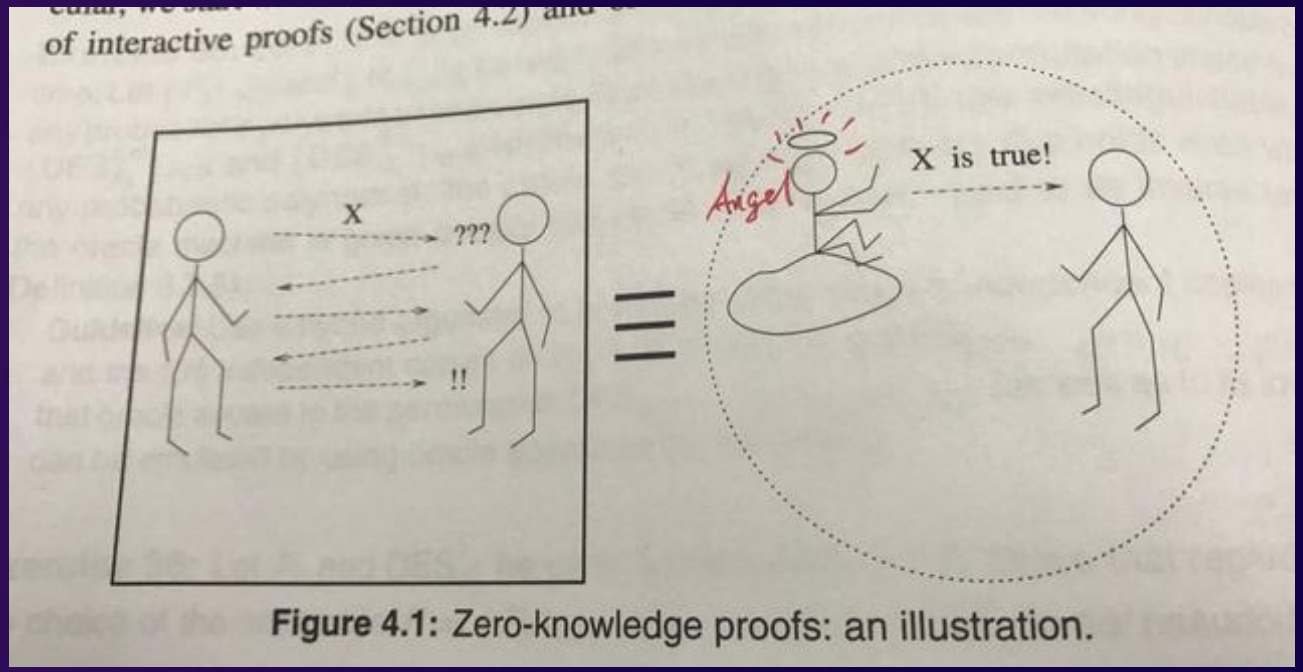
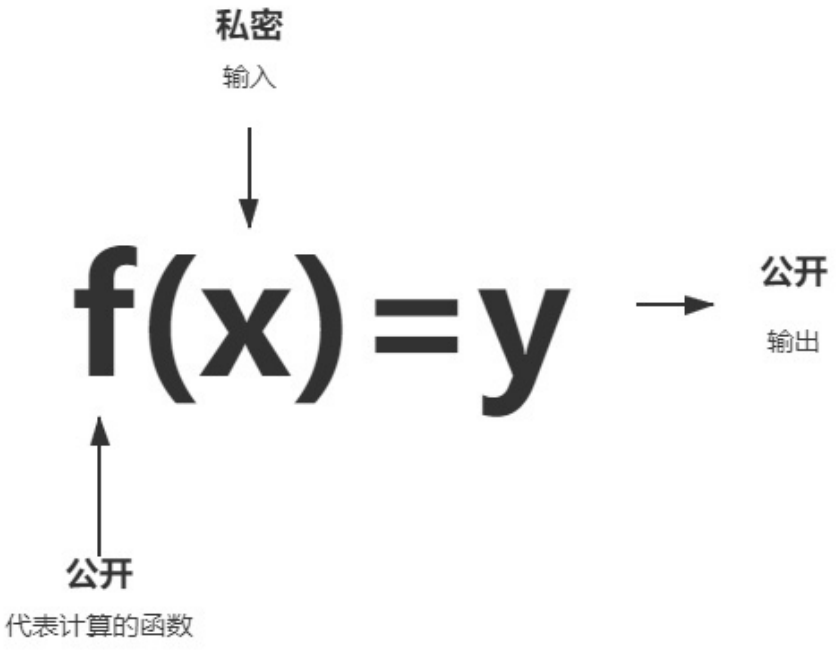


Figure 4.1: Zero-knowledge proofs: an illustration.

零知识证明基本流程

•函数f代表一个计算任务，计算的规则是公开的，但是计算的过程比较耗时；
•计算者使用私密的输入x，执行计算过程f得到输出y；
•目标是：
  •完备性（completeness）：计算者通过私密输入x，可以得到证据　　，使得其他人可以相信y是正确的输出；
  •可靠性（soundness）：假设计算者不知道x，那么通过伪造证据　　，从而使得其他人相信是一件概率极低的事件，换句话说，作弊被抓到的概率是很大的。
  •零知识：证据　　不能泄露有关输入x的任何信息；
  •简洁性（可扩展性）：验证证据的复杂度远小于计算的复杂度（O(log)），生成证据的复杂度近似于计算的复杂度（O(n)）



02

初探零知识证明

从头构造零知识证明

Alice和Bob

Alice作为证明者，试图向Bob证明自己拥有x，但并不泄露关于秘密x的任何知识。

Alice的**声明**是：拥有一个长度为10^6 的整数列表x，并且其中的每一个数都**处于1~10之间**

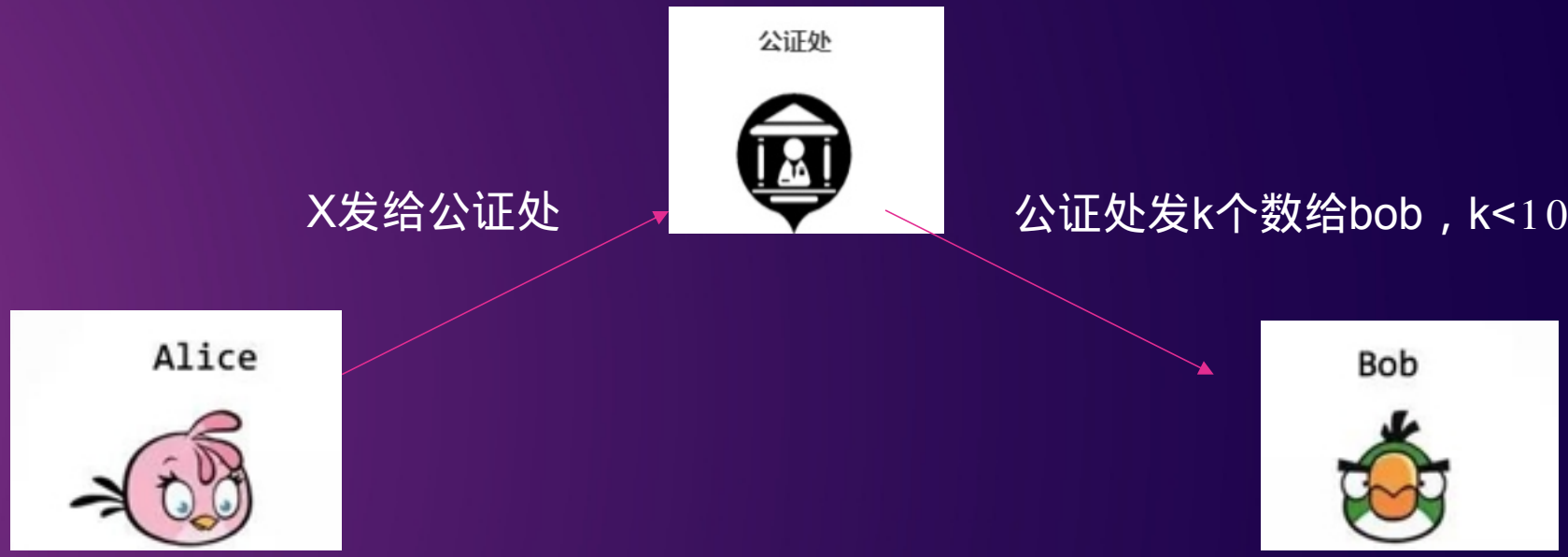
Bob作为验证者，对Alice抱有天然的怀疑；

Trial#1 直觉证明方案

X直接发给bob

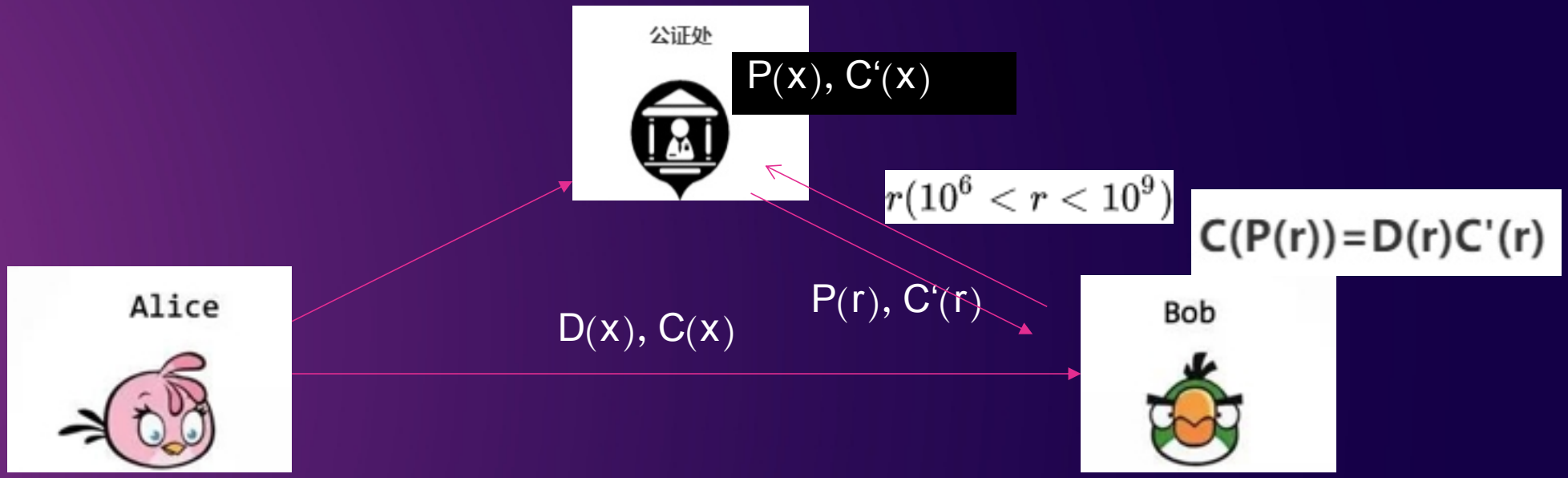
显然完备性和可靠性是满足的，零知识是不满足的。
简洁性也是不满足的，因为Alice验证的方法是逐个检查，
其复杂度等价于计算过程。

Trial#2 可信第三方



显然上述协议满足完备性，那么问题是k值的选择：如果我们令k=10^6，方案就退化成了方案1，而无论k取其他任何值，零知识总是无法满足的，当然k越小，泄露的隐私数量也就越小。那么考虑在泄露较少隐私的前提下（k取较小值），该协议可以满足可靠性吗？答案是不能。

Trial#3 构造多项式



Alice把10^6个整数转换成二维平面上的点坐标

$$(1, x_1), (2, x_2), \ldots, (10^6, x_{10^6})$$

然后用多项式插值法找到这对应的多项式P(x)

$$1 \leq P(X) \leq 10, X \in [1, 10^6]$$

Alice然后再构造C(x),约束多项式（ constraint polynomial ）

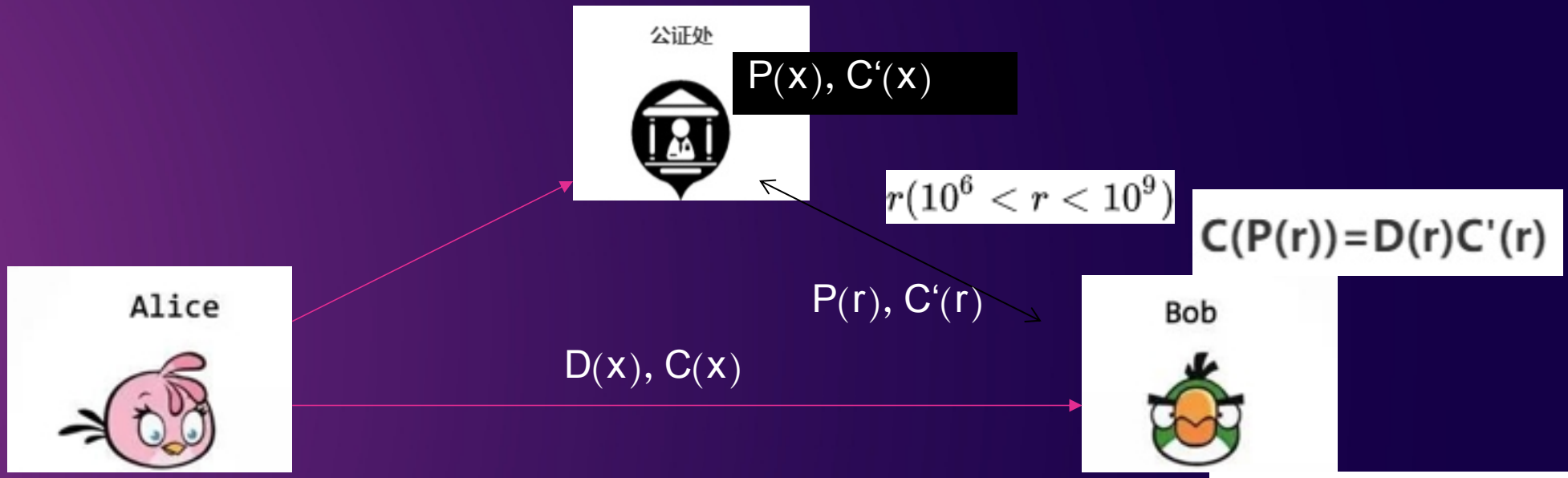
$$C(X) = (X-1) \cdot (X-2) \ldots (X-10)$$

$$C(P(X)) = 0, X \in [1, 10^6]$$

$$C(P(X)) = D(X) \cdot C'(X)$$

$$D(X) = (X-1) \cdot (X-2) \ldots (X-10^6)$$

Trial#3 构造多项式-完备性和简洁性

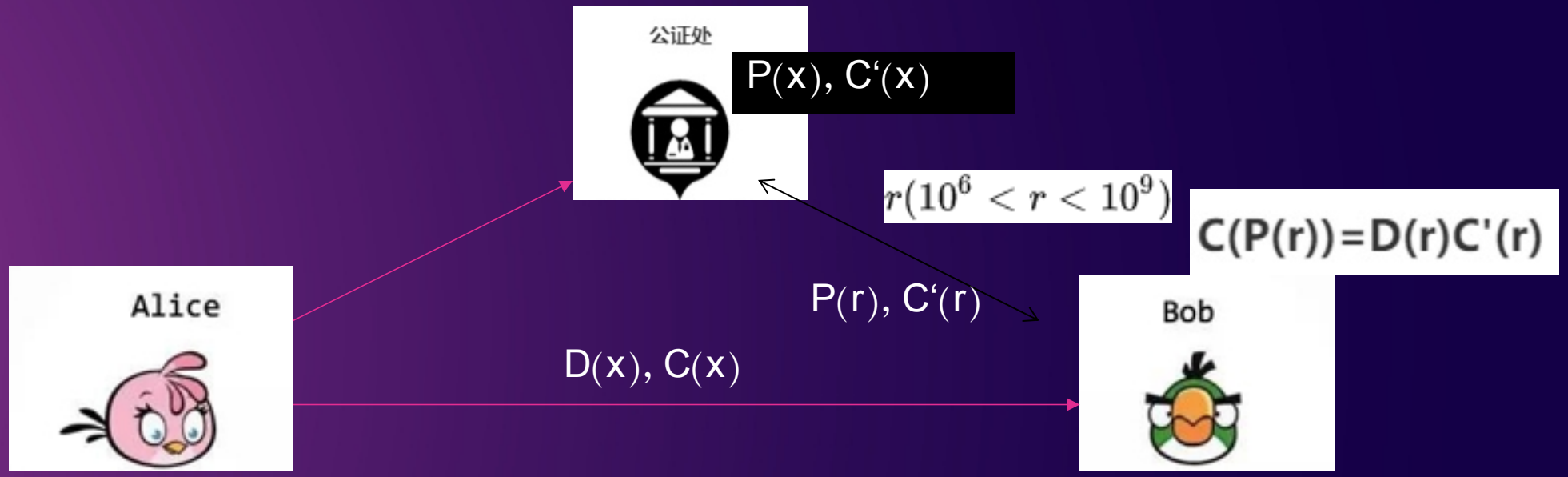


完备性：显然上述协议满足完备性，如果alice确实知道10^6个满足要求的整数就可以根据上述协议一步步操作，得到满足要求的多项式

简洁性也是满足的，因为Bob只需要检查一个点，因此验证的计算复杂度是很小的。

- $degree(P(X)) = 10^6 - 1$
- $degree(C(X)) = 10$
- $degree(C(P(X))) = 10^7 - 10$
- $degree(D(X)) = 10^6$
- $degree(C'(X)) = 10^7 - 10^6 - 10$

Trial#3 构造多项式-可靠性概率性满足



假设Alice想欺骗bob,发送一个假的P(x)，C(x)和D(x)保持不变，对任意多项式C’(x)，如下结论继续成立：

$$D(X)C'(X) = 0, X \in [1, 10^6]$$

而如下结论不成立：

$$C(P(X)) = 0, X \in [1, 10^6]$$

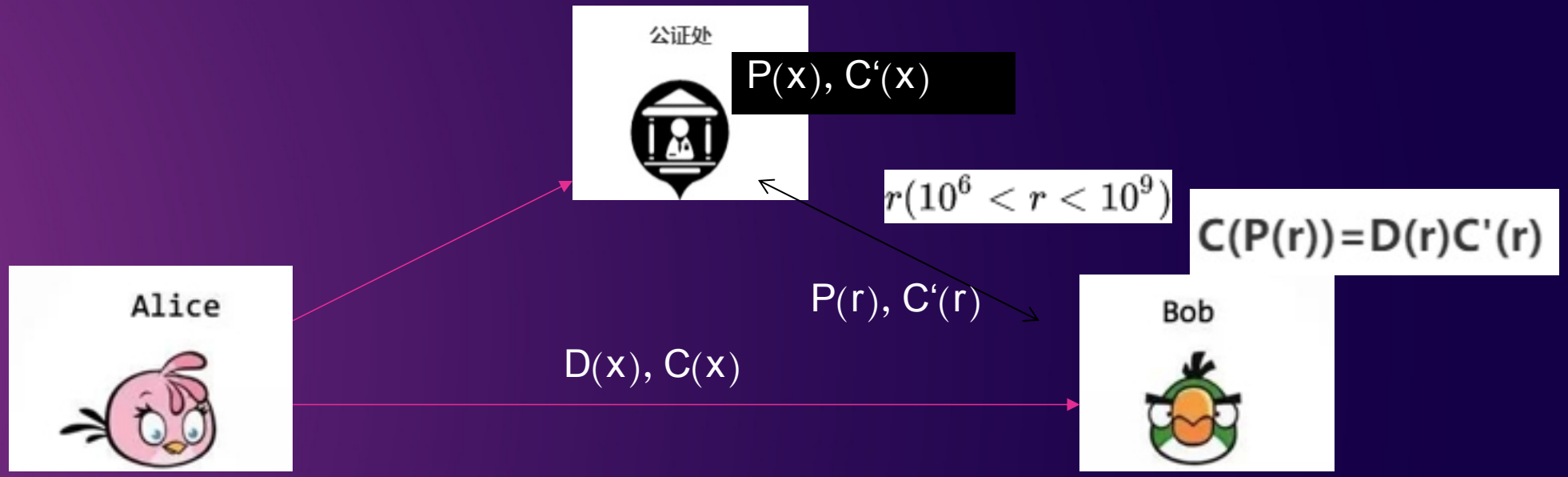
Alice无论发送什么样的多项式C’(x)，C(P(x))和D(x)C’(x) 都是两个不同的多项式

$$C(P(X)) = D(X)C'(X)$$

最多只有10^7-10个根

协议能验证通过的概率不超过 $\frac{10^7 - 10}{10^9 - 10^6} = 0.01001$

Trial#3 构造多项式-零知识不满足

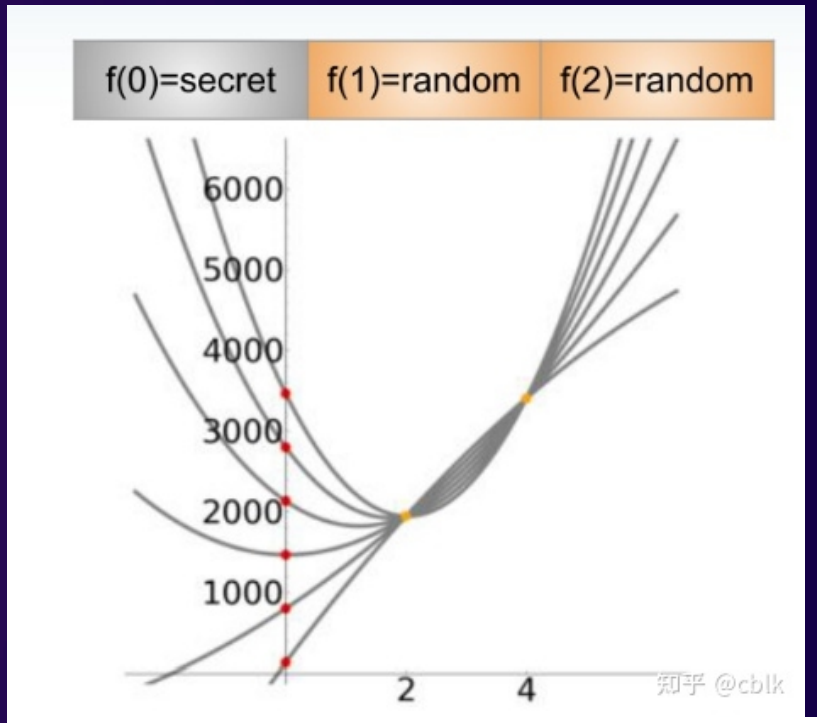


Bob可以自己生成任意的10^6个1-10整数，然后生成P‘(x)，如果知道了任意r的取值P(r)，判断P’(r)和P(r)是否相同，来得到P(x),的部分信息，虽然概率很小，但也不满足零知识要求

Trial#4 shamir-secret-sharing零知识

我们就可以试图构造一种在多项式中隐藏秘密值的方法。

Alice在10^6个点后面追加几个（比如说11个）随机点；那么这个多项式就不再由Bob持有的秘密数组唯一确定，这就意味着，即使Bob知道了多项式上一个点(r, P(r))，也无法从中推断出关于原始秘密的任何知识。



当且仅当拥有 $f(X)$ 上超过d+1个点的具体值时，才能知道 $f(0)$ 处的秘密值。

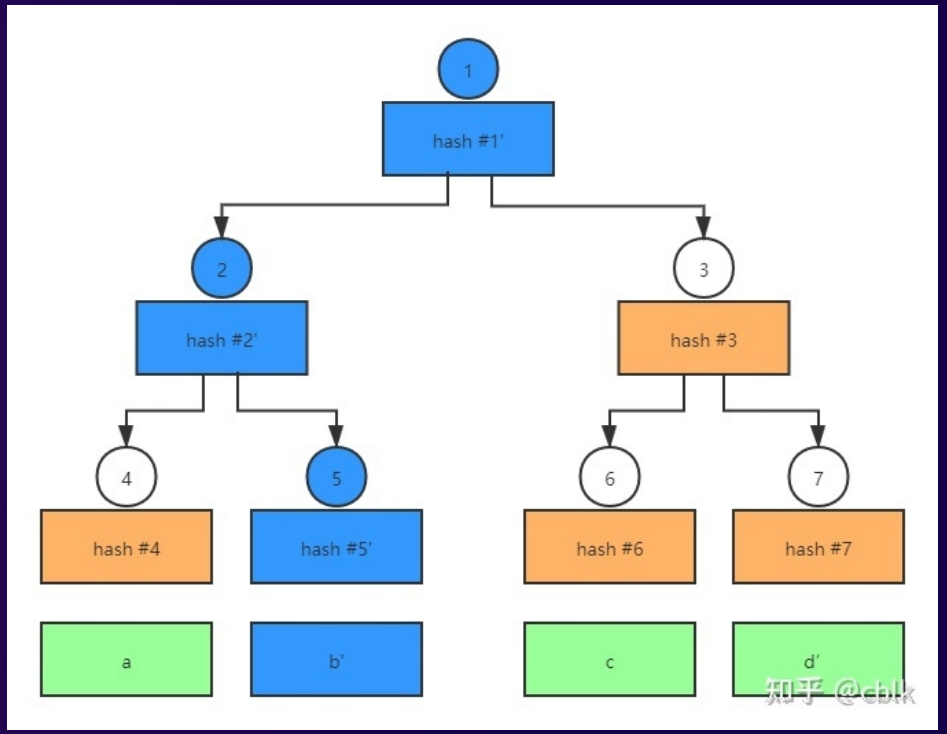
这样我们就找到了一种方法，把一个秘密值 S 分成了 $S_1, S_2, \ldots S_n (n > d)$ ，使得：

- 知道 S_i 中的d+1份，就可以还原出 S；
- 知道的份数少于d+1，则无法还原出 S 。

Trial#5 commitment-scheme去掉公证处

在新的协议中，Alice还是按照同样的方法构造多项式，但是他不会将多项式发送给别人，而是对每个多项式在[1,10^9]范围内的取值做承诺，对每个多项式（ $P(x), C'(x)$ ）：

- Alice计算多项式在[1,10^9]内的所有取值，共计 10^9 个值；
- Alice根据这 10^9 个值生成一棵Merkle树；
- Alice将根哈希发送给Bob。



Bob现在请求多项式在随机值r处的取值，并不通过公证处，而是使用基于Merkle树的承诺方案：

- Bob向Alice请求多项式在r处的取值和对应的验证路径；
- Bob检查验证路径。

到此为止，我们的协议中已经可以去除公证处这个角色了，所有的交互都在Alice和Bob之间发生。

Trial#6 Fiat-shamir-heuristic最终非交互

Alice声称自己拥有一个长度为 ***10^6*** 的整数列表x，并且其中的每一个数都**处于1~10之间**，为了证明自己没有说谎，Bob生成证明的过程如下：

证明

•Alice把自己持有的 10^6 个（1~10之间的）整数转化为 10^6 个二维平面上的点坐标：

$$(1, x_1), (2, x_2), \ldots, (10^6, x_{10^6})$$

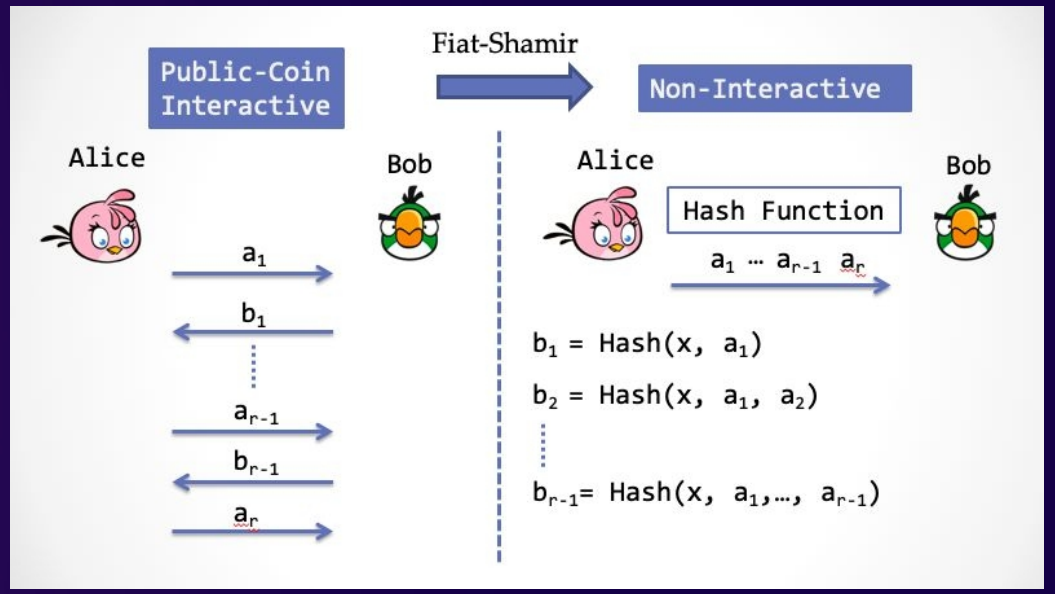
•Alice使用上一步生成的 ***10^6*** 个点和11个随机点，利用插值法构造多项式 P(x)；

•Alice构造约束多项式 ***C(x)*** 和多项式 ***D(x)***，这两个多项式都是公开的；

•Alice计算约束多项式 ***C'(x)***，满足关系 ；

•对于多项式 ***P(x)，C'(x)***，计算其在 ***【1-10^9】*** 之间的取值，并分别生成Merkle树

•利用上一步生成的Merkle树的根哈希作为随机种子，使用公开的抗碰撞哈希（CRH）函数计算随机值r，并根据随机值r得到两棵Merkle树上的验证路径。 Alice把这些作为证据发送给Bob。



验证

•Bob检查验证路径是否有效；

•Bob检查多项式取值是否满足约束。

为了提高协议的可靠性，我们可以设置多个检查点，也就是说Alice生成证明时选取多个随机值生成验证路径，Bob在检查的时候也一次性检查多个。这个改动不会影响协议的非交互性，但是可以有效地提高可靠性。

还不完美，但已经基本给出了ZKP的基本步骤

低阶测试（Low Degree Testing）问题

这是零知识证明协议中最复杂的问题，目前来看主要有两个方向的解决思路：
1、使用同态加密等方法，使得Alice只能够在事先提供的加密幂值（可信设置）上进行线性计算。由于加秘幂值的阶数是事先确认的，那么就可以保证Alice提供的多项式的度数在特定范围内。
2、使用承诺方案的思路，Bob可以让Alice提供更多的关于多项式的取值信息，通过设计好的协议，我们可以得到“Alice提供的取值点大部分（超过90%）在一个低阶多项式上”的结论。
第一种方法需要为每种特定的计算执行“可信设置”的操作，发展为**zk-SNARKs**一类的零知识证明协议。
第二种方法不需要执行可信设置，我们称之为“透明性（transparent）”，也就是**zk-STARK**方法。

谢 谢 观 看