

ZK SHANGHAI  
零知识证明工作坊

WORKSHOP!

# 数学基础构件

现代零知识密码学

Hosted by [SutuLabs](#) & [Kepler42B-ZK Planet](#)

课程资源: [zkshanghai.xyz](https://zkshanghai.xyz)

# 个人介绍



## 梁爽

区块链 架构师

上海交大 计算机博士生  
(休学创业中)

微信: icerdesign  
微博: @wizicer  
Github: @wizicer  
Twitter: @icerdesign  
LinkedIn: www.linkedin.com/in/icerdesign

- 1999年**
  - 正式开始学习写程序
- 2009年**
  - 在新媒传信（飞信）做高性能服务器程序架构及开发
- 2012年**
  - 在Honeywell工业控制部门做PLC、RTU上位机组态软件架构及开发
- 2017年**
  - 接触区块链，并开始创业开发区块链数据库
- 2020年**
  - 入学上海交大攻读博士学位，研究零知识证明数据库
- 2022年**
  - 获Chia全球开发大赛第一名，并开始Pawket钱包的开发
- 2023年**
  - 获得零知识链Mina的项目资助

# 起源

交互式零知识证明的诞生

# 零知识证明之前的历史

- Mental Poker over the Telephone [[SRA81](#)]

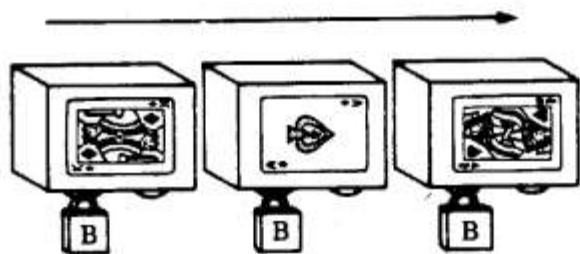


Once there were two “mental chess” experts who had become tired of their pastime. “Let’s play ‘Mental Poker,’ for variety” suggested one. “Sure” said the other. “Just let me deal!”

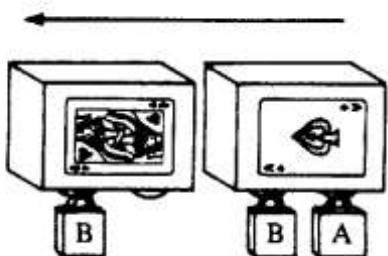


- 两个可能不诚实的玩家能否在没有使用任何纸牌的情况下，公平地玩扑克，例如通过电话？该文提供以下答案：
  - 1. 不行。（提供严谨的数学证明。）
  - 2. 可以。（给出正确而**完备**的协议。）

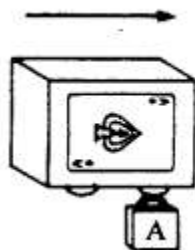
# 零知识证明之前的历史



- Bob对牌进行加密，并以随机顺序将其发送给Alice。



- Alice为Bob选择一张牌，并加密另一张牌用于自己，并将它们都发送给Bob。



- Bob解密两张牌，并将Alice的加密牌返回给她。

# 零知识证明之前的历史

- Mental Poker over the Telephone [[SRA81](#)]



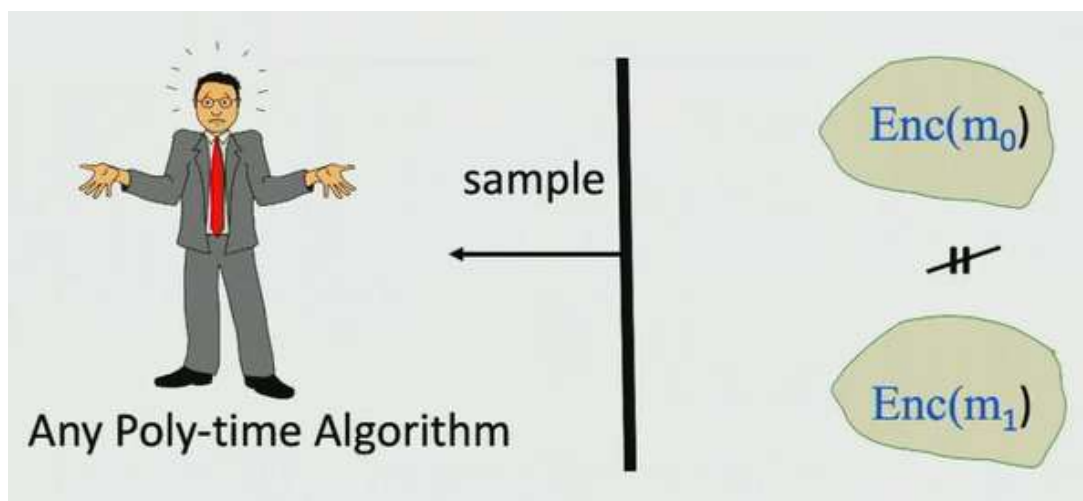
Once there were two “mental chess” experts who had become tired of their pastime. “Let’s play ‘Mental Poker,’ for variety” suggested one. “Sure” said the other. “Just let me deal!”



- Mental Poker是否可证明的隐藏了所有部分信息?
  - 如何定义部分信息?
  - 如何定义隐藏?
  - 如何理解可证明?
  - 以上对于Mental Poker来说是否足够?

# 零知识证明之前的历史

## 计算不可区分加密



## 如何加密一个比特位

- 找到一个**判定性问题**
  - 平均难度：不能在PPT中区分
    - 即对于随机实例，Yes/No实例的产生概率为 $1/2 + \text{negl}$
  - 易于生成：随机Yes/No实例
- 设定
  - 随机Yes实例=加密0
  - 随机No实例=加密1

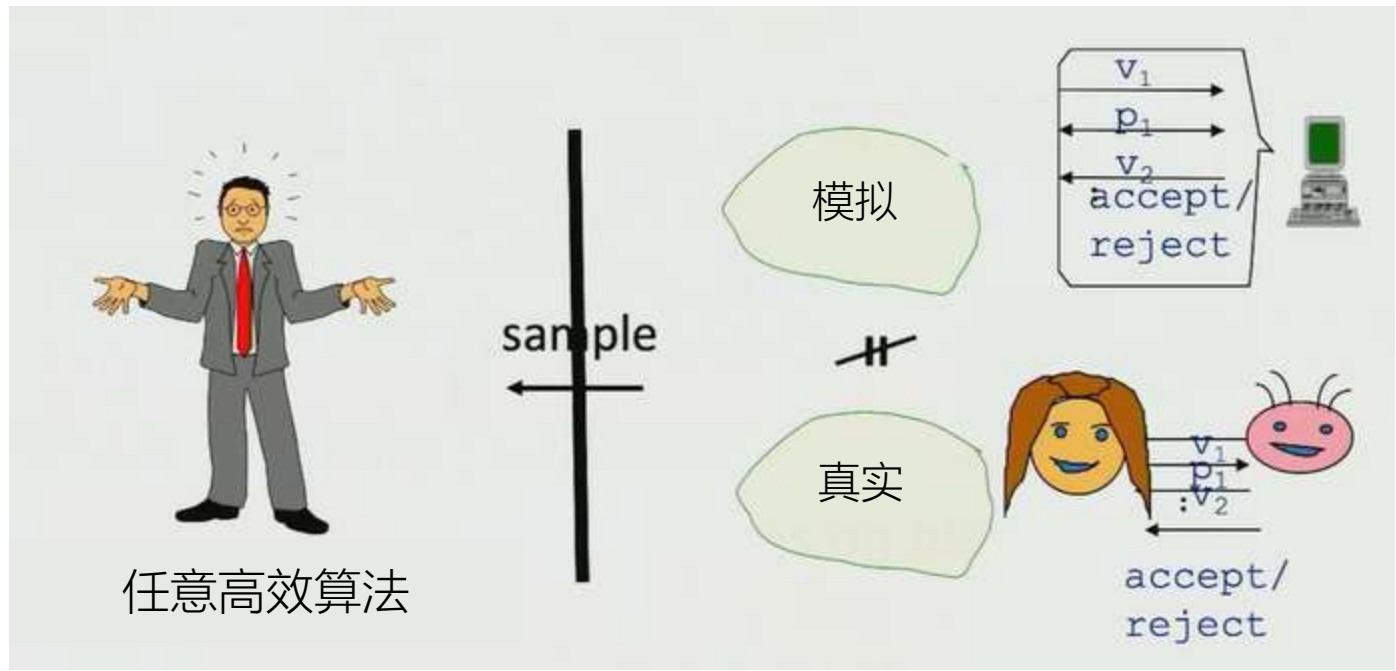
# Shafi和第一篇零知识证明论文





# 完美零知识性

如果“模拟视图”和“真实交互”在计算上无法区分，则为完美零知识。



# 课堂练习：利用非对称加密的零知识证明

- 1. V提出挑战：选择不可预测的随机明文M1，并利用公钥pk将明文M1加密为密文E
  - 2. V提出挑战：要求P对密文E解密
  - 3. P响应挑战：利用私钥sk对密文E解密为明文M2
  - 4. V验证挑战：接收明文M2，并比对M1=M2
- 
- 该方案是否满足零知识证明，是否满足完美零知识证明？

# 通用交互式零知识证明

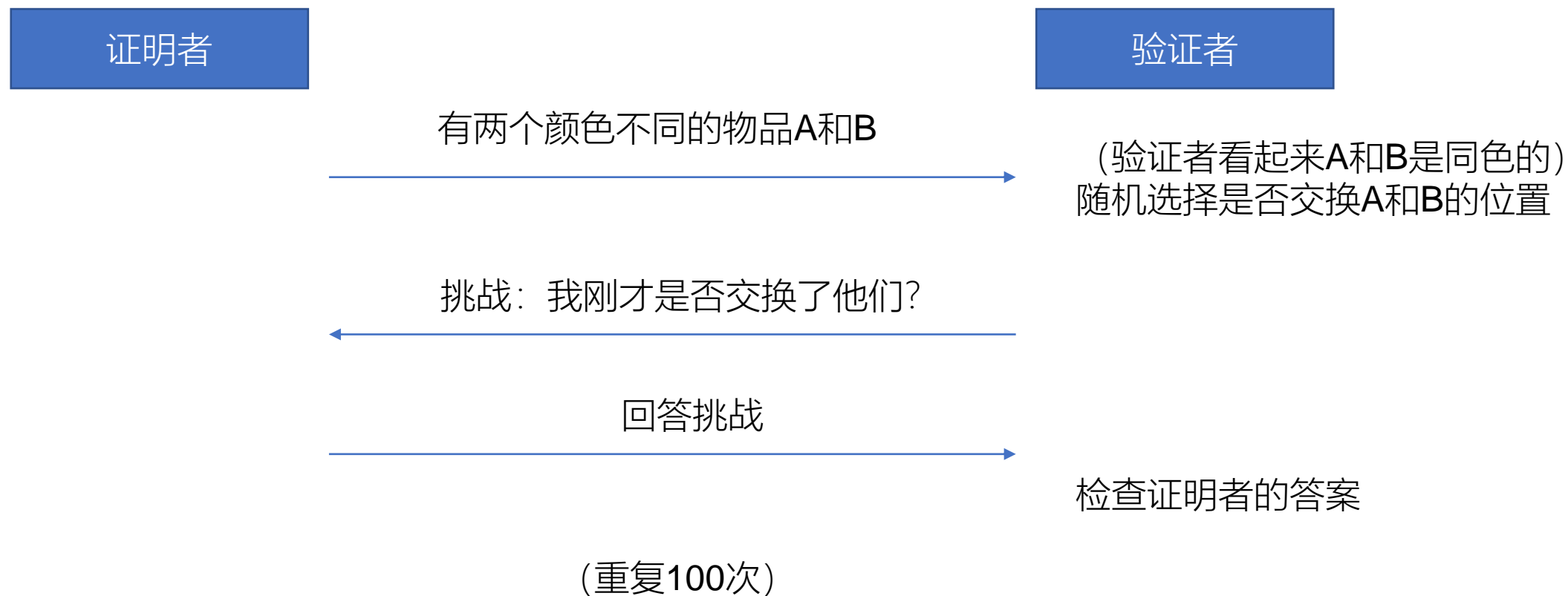
基于哈密顿回路的一种实现方式

# 零知识证明的基本概念

- 属于：证明系统
- 形式：交互性证明
- 角色：证明者 (P) , 验证者 (V)
- 理想属性
  - 完备性
    - 如果所有人都诚实且遵守规则，则一定可以正确运行。
  - 可靠性 (知识可靠性)
    - 如果证明者不遵守规则，则一定失败。
  - 零知识性
    - 验证者除了正确性外，无法获得任何其他信息

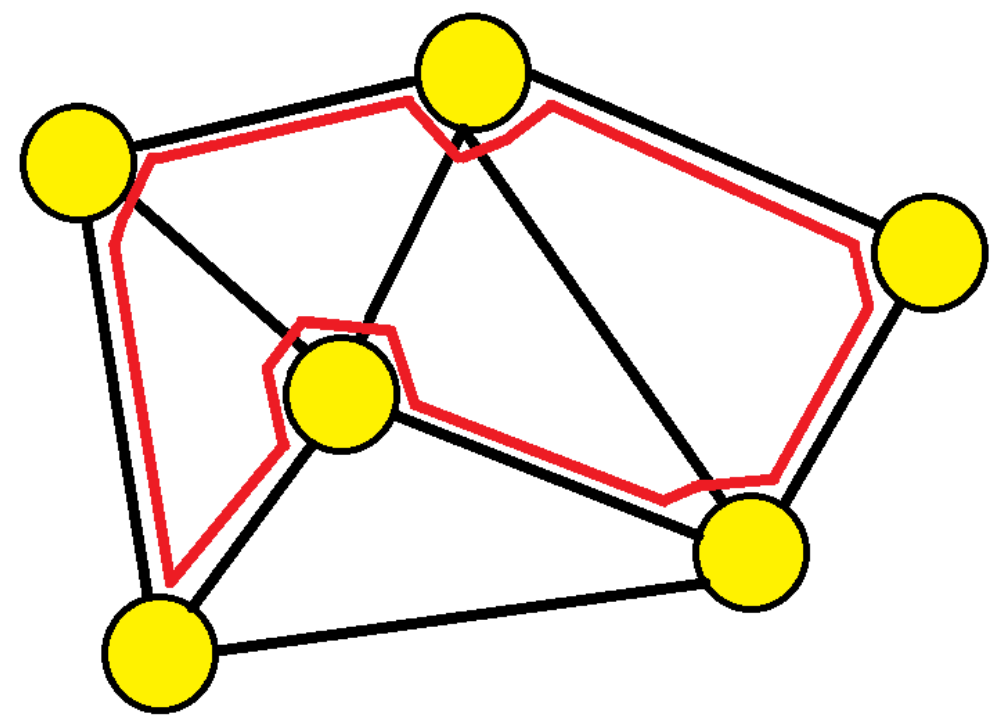
# 交互式证明

- 我（证明者）想要说服你（验证者），我能够区分出在你看起来是相同的两种颜色



# 汉密尔顿回路

- 汉密尔顿环：在图中每个顶点恰好经过一次的环，最终返回起点。
- 公共知识：图G
- 目的：证明者想向验证者证明他知道图G的一个汉密尔顿回路，而不泄露任何额外信息。



# 汉密尔顿回路

目的：证明者想向验证者证明他知道图G的一个汉密尔顿回路，而不泄露任何额外信息。

证明者

验证者

- 根据随机排列，为每个顶点分配一个1到n之间的标签，并记住这个排列。
- 对于每一对顶点 $ij$ ，将 $B_{ij}$ 放进加密盒子，其代表 $ij$ 是否是G的一条边。

所有的加密盒子 $B_{ij}$

随机选择 $b \in \{0,1\}$

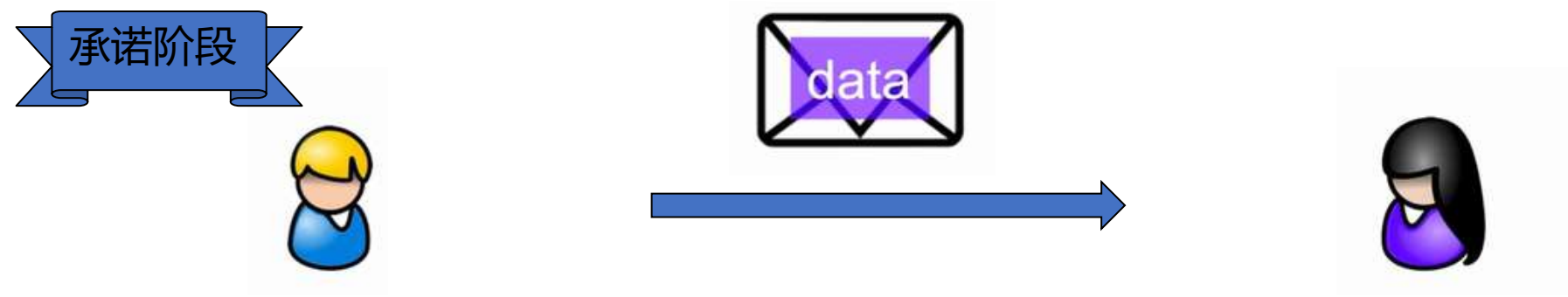
挑战：  $\begin{cases} \text{如果 } b = 0: \text{ 给我看图} \\ \text{如果 } b = 1: \text{ 给我看汉密尔顿回路} \end{cases}$


挑战：  $\begin{cases} \text{如果 } b = 0: \text{ 解密所有盒子及排列规则} \\ \text{如果 } b = 1: \text{ 解密包含汉密尔顿回路的盒子} \end{cases}$

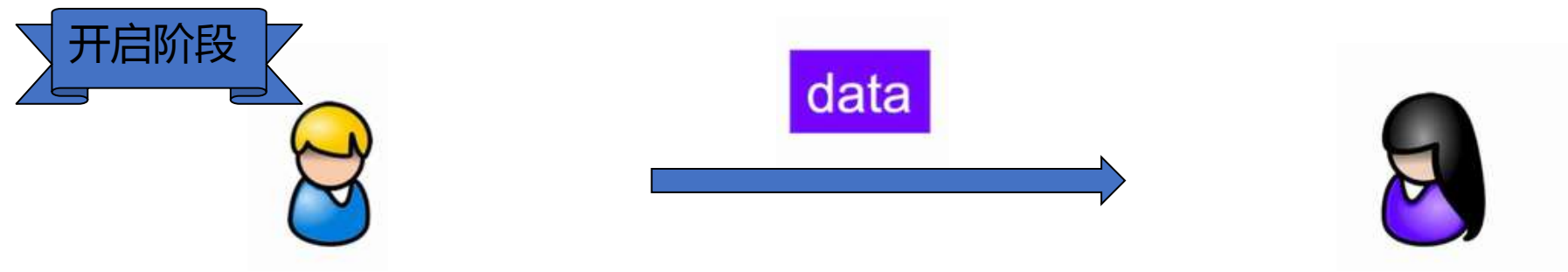
检查：

$\begin{cases} \text{如果 } b = 0: \text{ 是同一幅图} \\ \text{如果 } b = 1: \text{ 是汉密尔顿回路} \end{cases}$

# 承诺



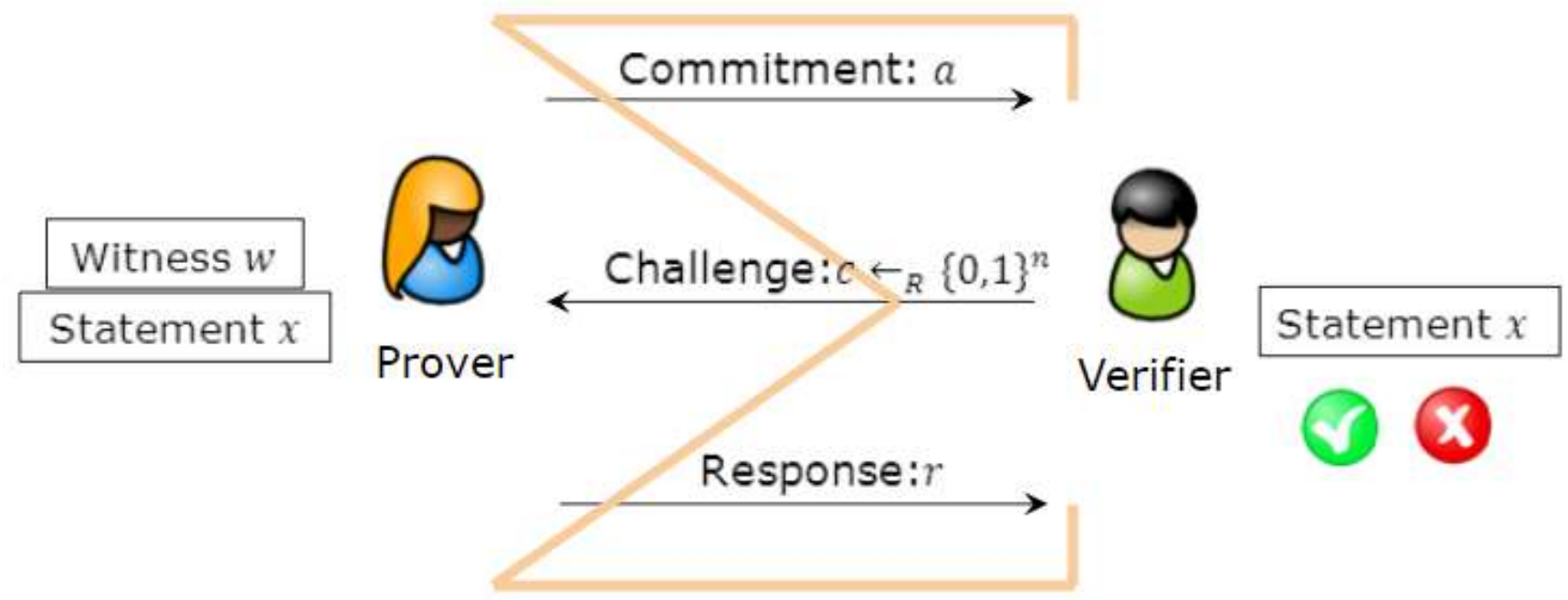
- 隐藏性 hiding:  不会泄漏任何关于  任何信息



- 绑定性 binding:  只能由  “开启”



# $\Sigma$ -Protocol (Sigma Protocol)



# 汉密尔顿回路的零知识证明协议属性

- 完备性
  - 如果所有人都按照协议要求执行，则协议成功。
- 可靠性
  - 如果没有汉密尔顿回路，无论证明者做什么，验证者都会以 $1/2$ 的概率拒绝。
- 知识可靠性
  - 这是可靠性的更强要求，它认为即使图中存在汉密尔顿回路，但证明者不知道它的存在，协议仍将失败。
- 零知识性
  - 如果验证者接受，则他无法从交互中获得除了证明是正确的以外的任何信息。

# 通用零知识证明

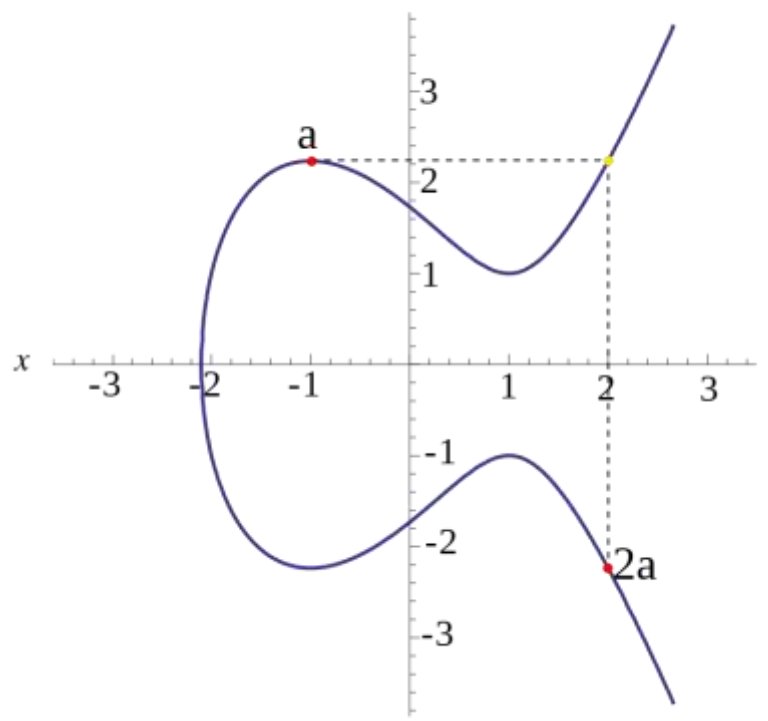
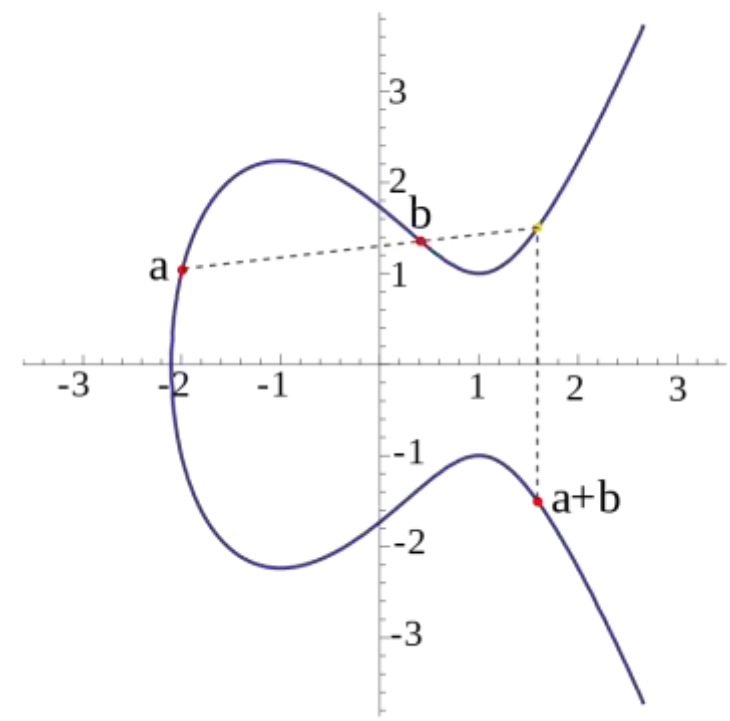
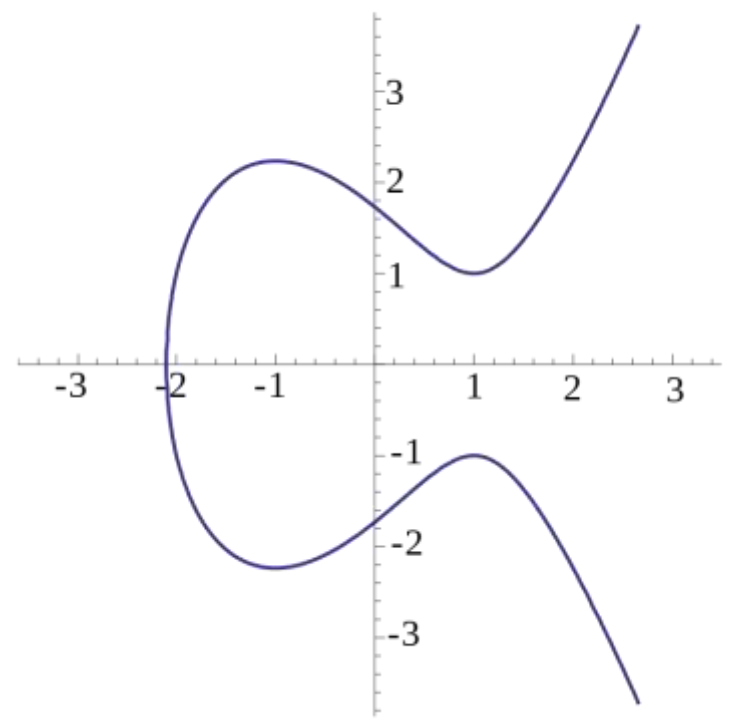
- 为什么关注汉密尔顿回路问题
  - 因为他是NP完全问题
  - 更多关于NP问题: <https://zkshanghai.xyz/math/computation.html>
- 每个NP问题都能在多项式时间内被规约为NP完全问题
- 通过Fiat-Shamir启发式将交互式证明转换为非交互式证明
- 利用汉密尔顿回路问题做零知识证明既不精简, 也不实用
  - 后续课程中的zk-SNARK会解决这个问题

# 椭圆曲线

椭圆曲线及离散对数问题

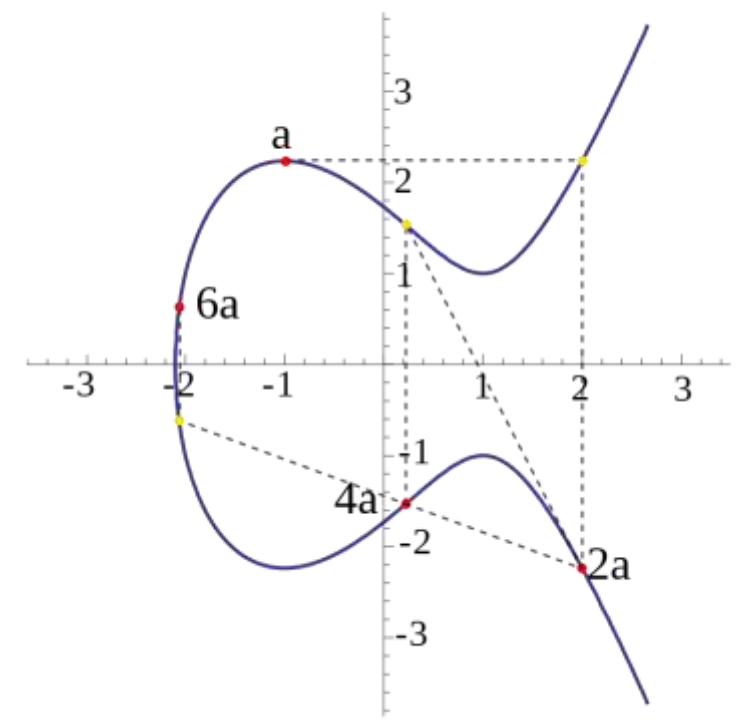
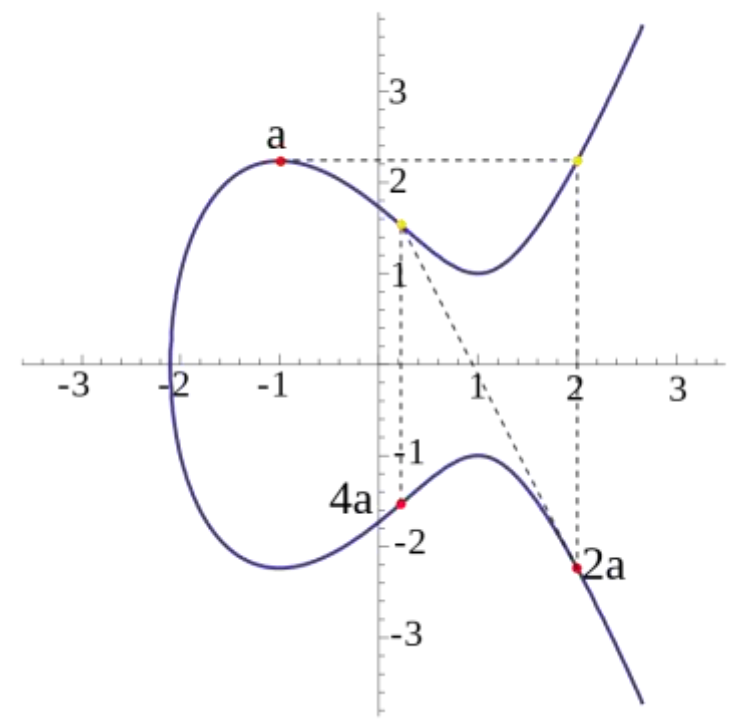
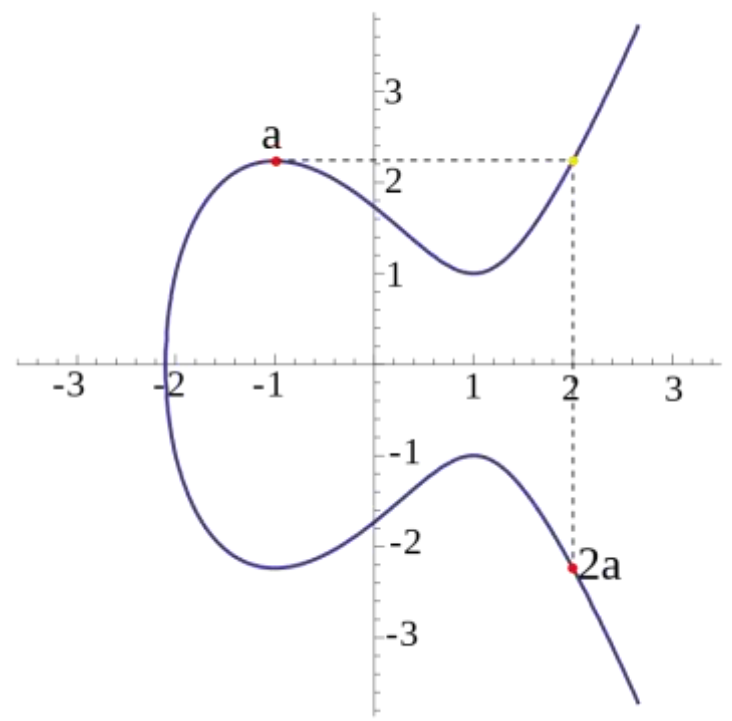
# 定义曲线上点的加法

曲线:  $y^2 = x^3 + ax + b$



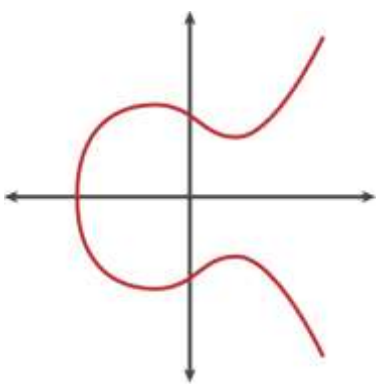
# 加出任意位置

曲线:  $y^2 = x^3 + ax + b$

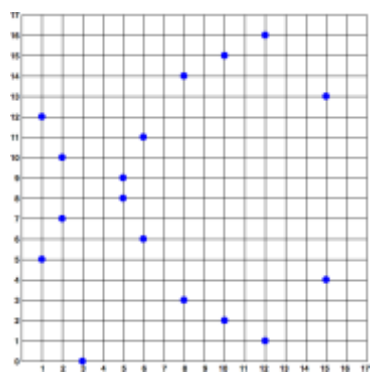


# 椭圆曲线

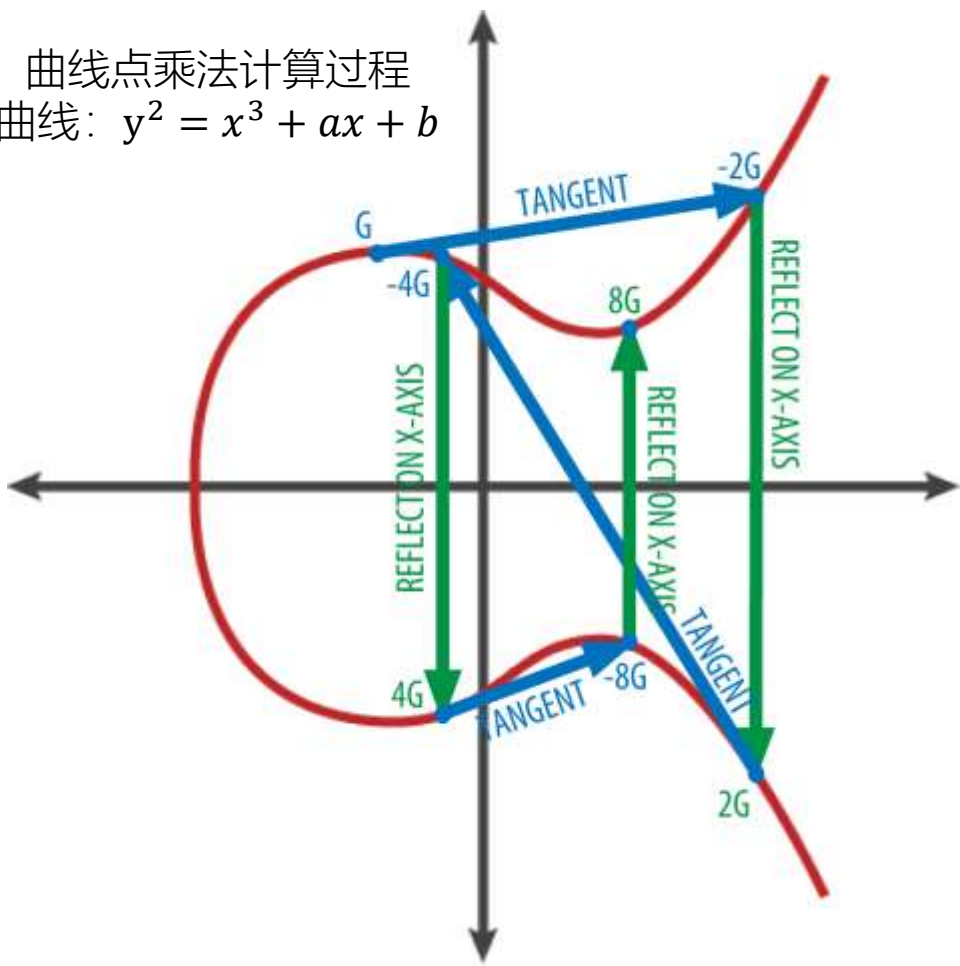
曲线点乘法计算过程  
曲线:  $y^2 = x^3 + ax + b$



实域上的曲线



有限域上的曲线



- 私钥=群上元素 (标量)
- 公共定义:
  - 曲线参数
  - 生成元 (基础点)
  - 阶 (个数)
- 公钥=私钥\*生成元

离散对数问题:  
从点坐标恢复标量是困难的  
即从公钥恢复出私钥是困难的

注意: 不是所有离散对数问题  
都是困难的

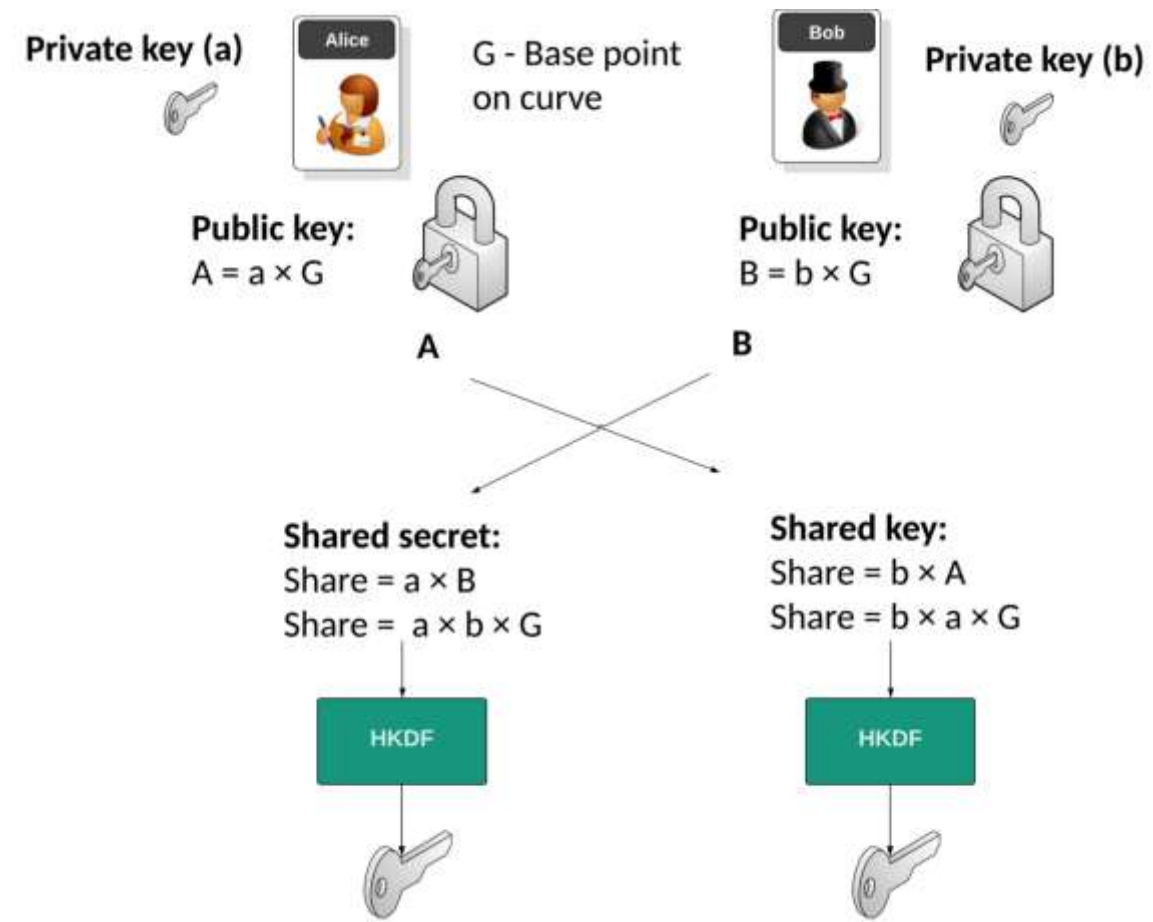
# 各类曲线

Curve	Safe?	Parameters:			ECDLP security:				ECC security:			
		field	equation	base	rho	transfer	disc	rigid	ladder	twist	complete	ind
Anomalous	False	True✓	True✓	True✓	True✓	False	False	True✓	False	False	False	False
M-221	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓
E-222	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓
NIST P-224	False	True✓	True✓	True✓	True✓	True✓	True✓	False	False	False	False	False
Curve1174	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓
Curve25519	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓	True✓
BN(2,254)	False	True✓	True✓	True✓	True✓	False	False	True✓	False	False	False	False
brainpoolP256t1	False	True✓	True✓	True✓	True✓	True✓	True✓	True✓	False	False	False	False
ANSSI FRP256v1	False	True✓	True✓	True✓	True✓	True✓	True✓	False	False	False	False	False
NIST P-256	False	True✓	True✓	True✓	True✓	True✓	True✓	False	False	True✓	False	False
secp256k1	False	True✓	True✓	True✓	True✓	True✓	False	True✓	False	True✓	False	False



# ECDH

## Elliptic Curve Diffie-Hellman



# 困难问题

- 离散对数问题
  - The Discrete Logarithm Problem(DLP)
- 计算性DH问题
  - The Computational Diffie-Hellman Problem(CDH)
- 判定性DH问题
  - The Decisional Diffie-Hellman Problem (DDH)

难度：  $DLP > CDH > DDH$

# Schnorr Protocol

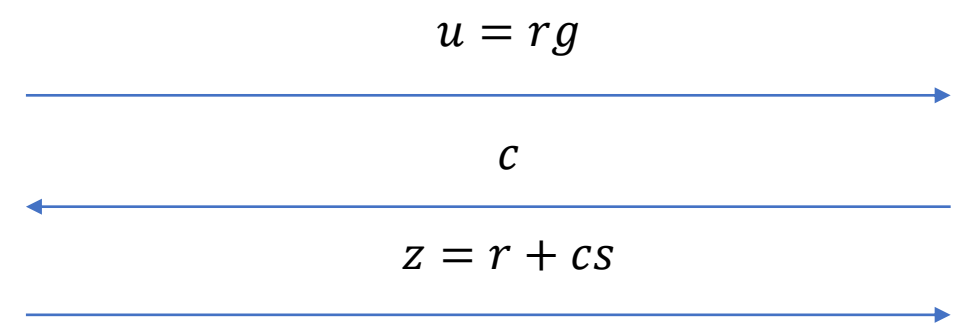
目的：证明者想向验证者证明他知道秘密  $s \in \mathbb{Z}_q$  使得  $x = sg \in \mathbb{G}$

证明者

- $s \in \mathbb{Z}_q, x = sg \in \mathbb{G}$
- $r \leftarrow_R \mathbb{Z}_q$

验证者

- $x \in \mathbb{G}$
- $c \leftarrow_R \mathbb{Z}_q$



检查：  $zg = u + cx$

论断：该协议是基于离散对数的零知识证明