# 第十章、零知识证明及其在区块链中的应用

# 目　录

# 目　录

# 10.1 引言

**零知识证明(Zero—Knowledge Proof)**，是由S.Goldwasser、S.Micali及C.Rackoff在20世纪80年代初提出的。它指的是证明者能够在不向验证者提供任何有用的信息的情况下，使验证者相信某个论断是正确的。

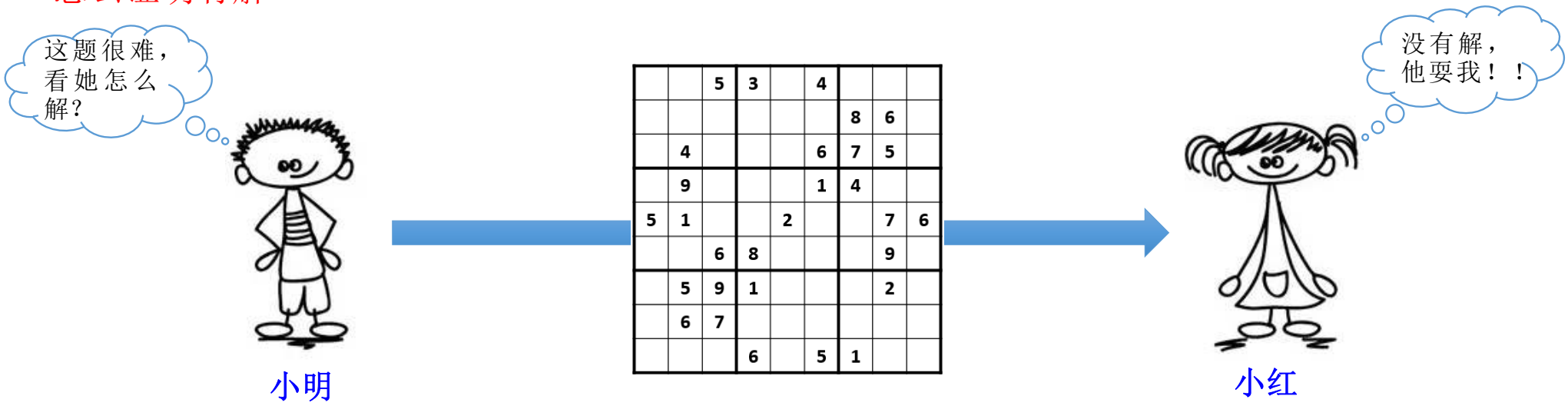
零知识证明实质上是一种涉及两方或更多方的协议，即两方或更多方完成一项任务所需采取的一系列步骤。零知识证明必须包括两个方面，一方为证明者P，另一方为验证者V。证明者试图向验证者证明某个论断是正确的，或者证明者拥有某个知识，却不向验证者透露任何有用的消息。

零知识证明目前在密码学中得到了广泛的应用，尤其是在认证协议、数字签名方面。

# 10.1 引言

## 例子：数独解的零知识证明

小明和小红很喜欢玩数独游戏，平日里他们相互给对方出题做。有一天，小明出了一道很难的数独题，可小红花了很长时间都没有解出来。于是，她就怀疑小明出的这道题没有解，认为小明再耍她。但是，小明信誓旦旦的说这个题有解，并且他知道这个解。小红就问“你怎么证明有解？”




| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 5 | 3 | | 4 | | | |
| | | | | | | 8 | 6 | |
| | 4 | | | | 6 | 7 | 5 | |
| | 9 | | | | 1 | 4 | | |
| 5 | 1 | | | 2 | | | 7 | 6 |
| | | 6 | 8 | | | | 9 | |
| | 5 | 9 | 1 | | | | 2 | |
| | 6 | 7 | | | | | | |
| | | | 6 | | 5 | 1 | | |

*源自《一个数独引发的惨案：零知识证明（Zero-Knowledge Proof）》

# 10.1 引言

## 例子：数独解的零知识证明

那么小明要如何证明呢？

- 方法1：用数学模型来证明。缺点：复杂，难以理解；
- 方法2：用正确答案来证明。缺点：小红知道答案了。

有没有一种方法，既能够让小红相信小明知道答案，又不告诉小红答案呢？

有，用零知识证明的方法来证明该数独题目有解。

下面，将展示小明的零知识证明方法：

# 10.1 引言

## 例子：数独解的零知识证明

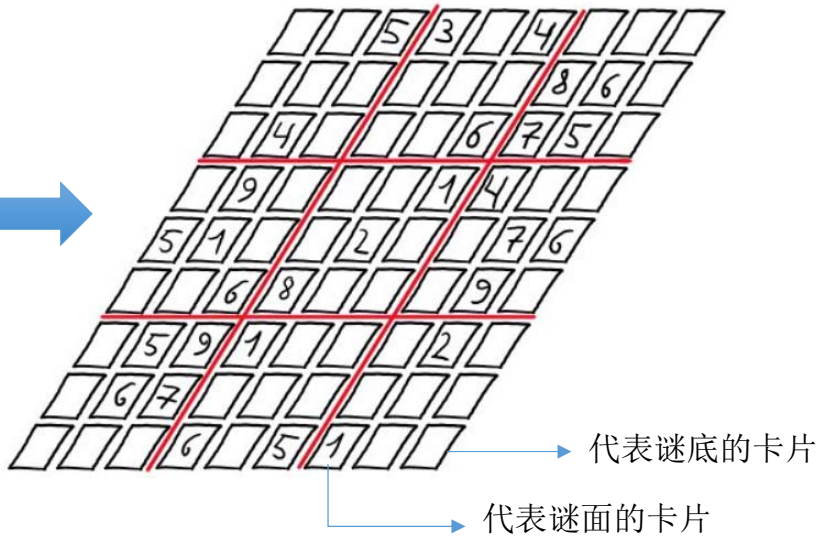
### 步骤一：承诺

小明拿出81张空白的卡片，并按照解的排列将1~9中的数字写到卡片上。之后，将代表谜底的卡片，数字面朝下放在桌上；代表谜面的卡片，则数字面朝上放在桌上。

在此过程中，小红不能够偷看。（亦可以由小明预先准备好）

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 5 | 3 | | 4 | | | |
| | | | | | | 8 | 6 | |
| | 4 | | | | 6 | 7 | 5 | |
| | 9 | | | | 1 | 4 | | |
| 5 | 1 | | | 2 | | | 7 | 6 |
| | | 6 | 8 | | | | 9 | |
| | 5 | 9 | 1 | | | | 2 | |
| | 6 | 7 | | | | | | |
| | | | 6 | | 5 | 1 | | |

承诺



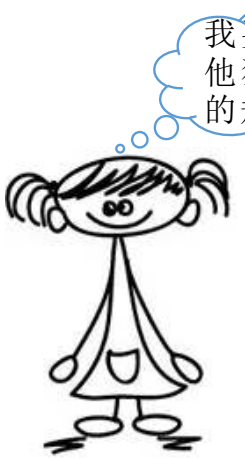
# 10.1 引言

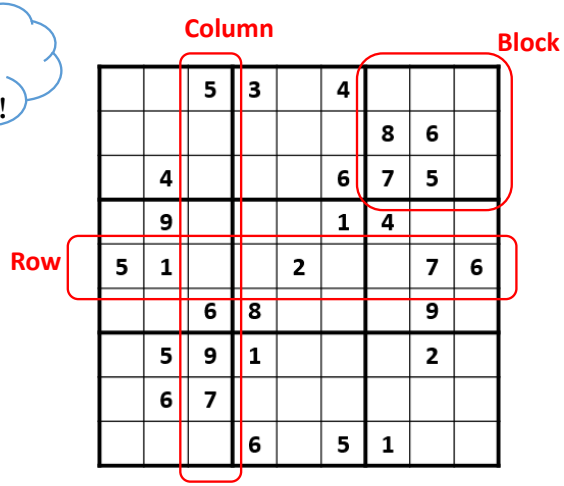
## 例子：数独解的零知识证明

### 步骤二：随机挑战

在小明放好卡片后，小红想打开卡片揭晓谜底。但是，小明不允许，因为他不想让小红知道答案（通过承诺获取答案是不被允许的）。那么，小红就会质疑小明的答案，她会按照数独的规则来检验答案。如果答案是正确的，那么必然会满足规则：每行（row）/每列（column）/每个九宫格（block）是由1~9的数字组成，且不可重复。小红就按照以上的任何一种规则来验证。小红将其随机选取的验证规则发送给小明。

挑战：Row 或 Column 或Block



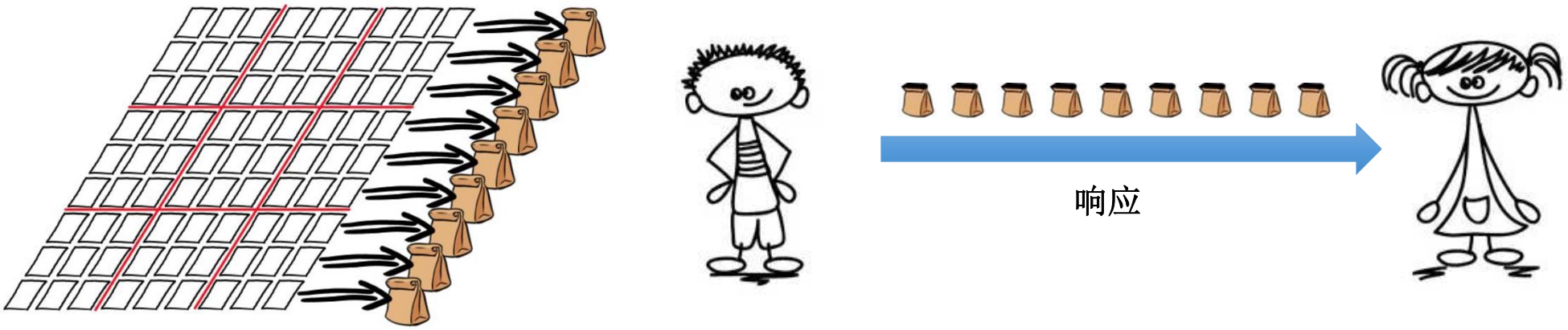
# 10.1 引言

## 例子：数独解的零知识证明

### 步骤三：响应

小明收到小红的验证挑战（不妨设验证规则为按行验证）后，他将每一行的卡片收集放入一个麻布袋中并打乱。所有卡片都被收完放在了9个麻布袋里。

小明接着把这9个麻布袋交给小红。（放卡片的过程是被小红监视的）



# 10.1 引言

## 例子：数独解的零知识证明

### 步骤四：验证

小红打开9个袋子，分别验证每个袋子是不是包含1~9的卡片。

- 如果全是，那么验证通过，即小明可能知道解。（是可能知道，而不是确定知道）
- 如果不是，那么验证失败，即小明不知道解。



➢ 小明答疑：

袋子中卡片的打包是在我承诺的组合上完成的，我事先不知道你会选择哪个规则来验证。

如果我按照行规则准备了卡片，当你选列规则的时候，我就会失败的。

也就是说，我不知道答案的话，我是可能失败的。

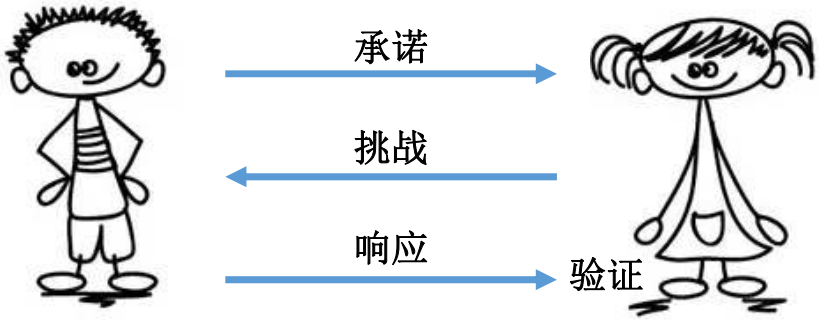
# 10.1 引言

## 例子：数独解的零知识证明

### 重复证明（Repeat Times）+零知识（Zero Knowledge）：

小明与小红每执行一次上述的证明过程，小红都有1/3的概率抓获没有答案的小明。换而言之，小明没有答案并且通过小红验证的可能性只有1/3（即小明猜测到了小红要选择的挑战规则）。

但是，小红还是不服气。她觉得可能性太大了，于是她让小明重复证明n次，直到她觉得小明猜测成功概率非常非常，即$1/3^n$非常小。




问题：小红知道部分答案了吗？

- 小红答疑：

这么多次试验下来，我还是不知道真正的题解。

我只知道每次小明放置卡片的排列里很大几率每行每列每个九宫格确实都是没有重复的数字，这就说明很大几率这题是有解的，而且小明很大几率确实知道这题的解。

# 10.1 引言

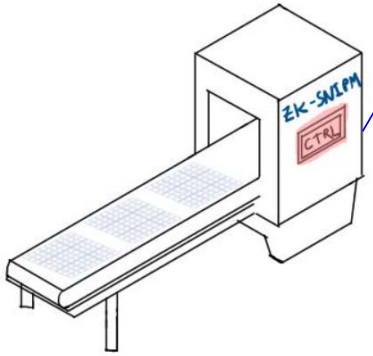
## 例子：数独解的零知识证明

### 非交互式扩展（Non-Interactive ）：

有了这个证明方法后，小明和小红就养成了通过零知识证明去证明给对方看自己知道某题解的习惯。

可是，新的问题来了。小明如何向所有人证明他的题解呢？让小红来挑战？那么，有人就会怀疑，小红配合小明来作弊。

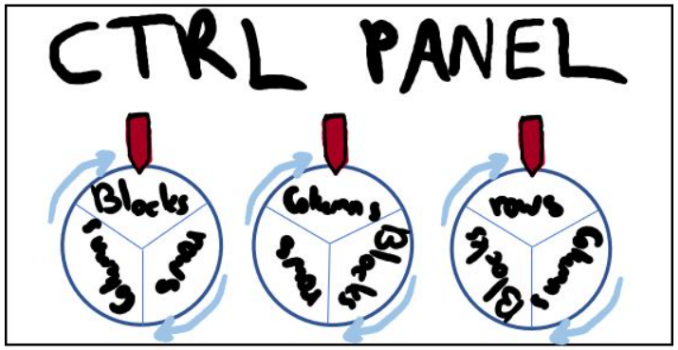
于是，小明又发明了一个“零知识数独非交互式证明机”。




零知识数独非交互式证明机

自动化证明，无需人为交互：

小明只要把卡片放在传送带上，机器会自动选择按行，或列，或九宫格来收取卡片，放到袋子里打乱顺序，然后把袋子通过传送带再送出来。然后小明就可以当着镜头的面拆开袋子展示里面的卡片。




挑战随机产生+非人为控制：

这台机器有一个控制面板，打开里面是一串旋钮，这些旋钮用来指示每次试验的选择（行，列， 九宫格）。

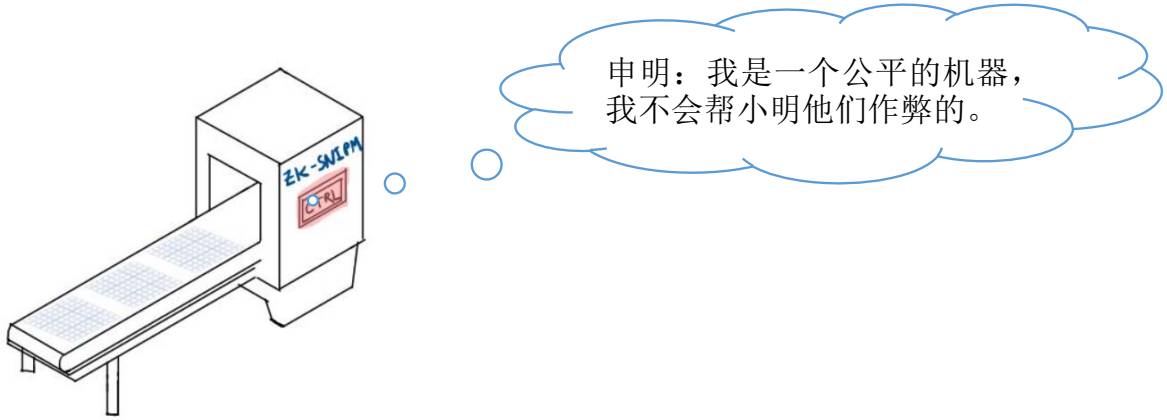
# 10.1 引言

## 例子：数独解的零知识证明

### 可信初始设置（Trusted Setup）：

当然，还有很多喜欢“抬杠”的人。他们认为，这个机器是小明和小红做的，他们可以在里面设置后门，或者他们知道控制面板随机序列的方式。这样，小明也能够作弊。

没办法，小明只有开直播，让大家看到他如何安装这个机器的，并且公布机器的随机序列产生方式如何，等等。。。只要观众怀疑的，小明都需要将其公开解释。这一过程，就可以称之为“可信任的初始设置仪式（trusted setup ceremony）”。



# 10.1 引言

## 例子：数独解的零知识证明

至此，一个比较完整的零知识证明方法构造完成。在此过程中，出现了零知识证明的一些概念：

- 证明：什么是交互证明系统？什么是非交互式证明系统？
- 零知识：什么是知识？什么信息包含知识？
- 证明的安全属性包含什么？
- 保障系统可信的条件有哪些？
- ……

# 目　录

# 10.2 零知识证明的概念

## 1. 交互证明系统

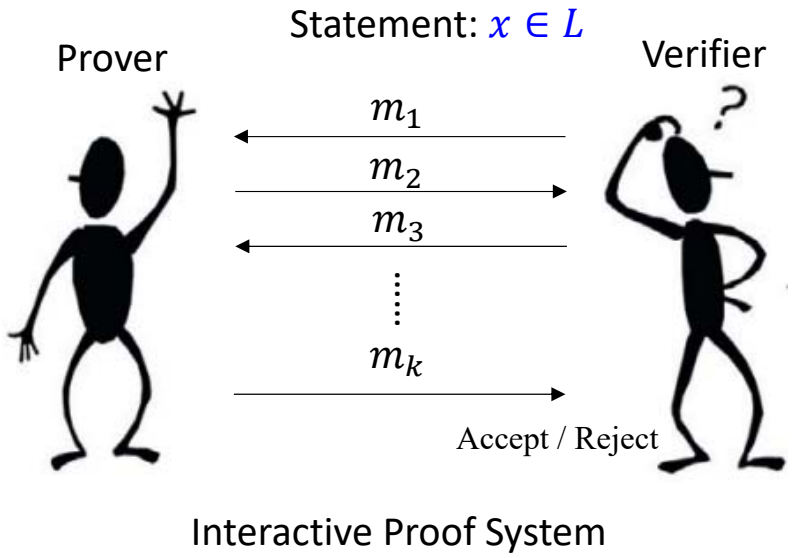
在计算机复杂性理论中，**交互证明系统（Interactive proof system）**是一种抽象机器，其将计算建模为两个参与方（证明者和验证者）之间的消息交换。通过交换信息，参与方证明某个声明成立（例如：$x \in L$，$L$为NP语言）。其中，

- 证明者（Prover）

  拥有无穷的计算能力，且不可信；

- 验证者（Verifier）

  拥有受限的计算能力（概率多项式时间），且诚实。




Interactive Proof System

*注：如果证明者能力也是受限、多项式时间的，该系统称为**交互论证系统（Interactive argument system）**。*

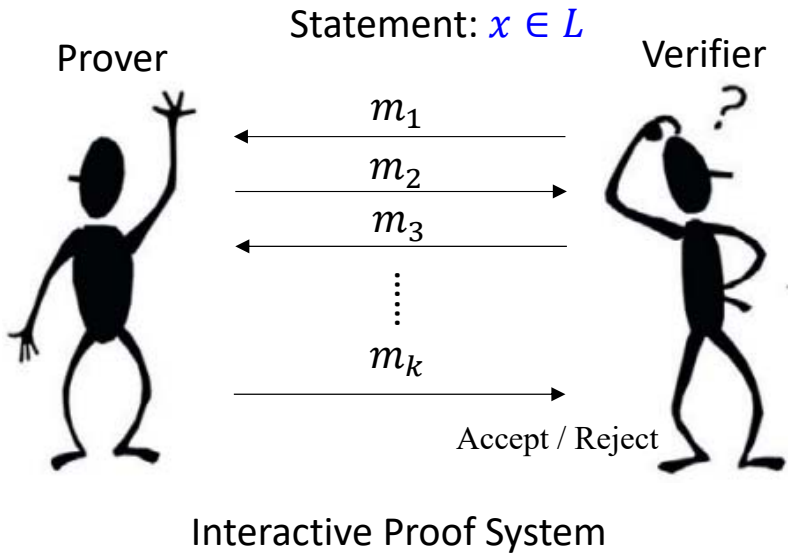
# 10.2 零知识证明的概念

## 1. 交互证明系统

### ➢ 交互证明系统的性质

- 完备性：对于正确的声明，验证者“总是”接受。也就是说，一个诚实的验证者是能够被一个诚实的证明者用一个真声明来说服。
- 可靠性：对于错误的声明，验证者“总是”拒绝。也就是说，任何欺骗的证明者都不能够让诚实的验证者相信一个错误的声明。

此外，还有强调一个特性，即交互式：证明者与验证者之间采用交互的形式来完成证明过程。




Interactive Proof System

# 10.2 零知识证明的概念

## 1. 交互证明系统

### ➢ 交互证明系统的定义

***Def.*** 对于语言$L \in \{0,1\}^*$，以及一对交互图灵机$< P,V >$，其中$P$拥有无限的计算能力，称为证明者，$V$为概率多项式时间的验证者。

称$< P,V >$为语言$L$的交互证明系统，如果满足以下条件：

- 完备性（Completeness）：对于任何公共输入$x \in L$，

$$\Pr[(P,V)(x) = 1 | x \in L] \leq 1 - negl(|x|).$$

- 可靠性（Soundness）：对于任意公共输入$x \notin L$和任意无限计算能力的证明者$P^*$,

$$\Pr[(P^*,V)(x) = 0 | x \notin L] \geq 1 - negl(|x|).$$

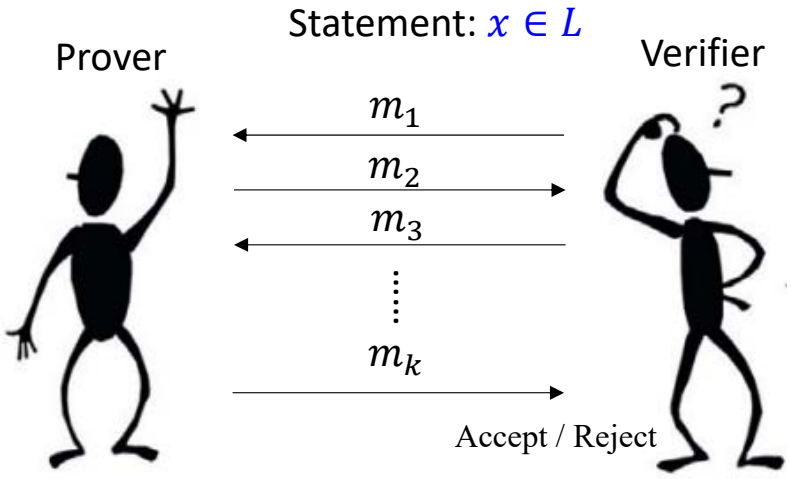
其中，$negl(|x|)$为一个可忽略函数。

# 10.2 零知识证明的概念

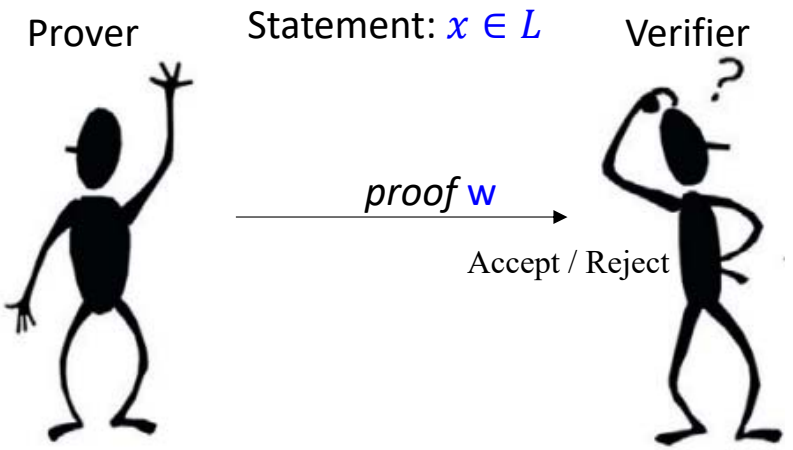
## 1. 交互证明系统

### ➢ 与传统证明系统的区别

由定义可知，交互证明系统与传统数学证明最主要的区别在于交互性、参与者的随机性以及完备性和可靠性错误。数学证明系统是严谨的、要么使用不证自明的陈述，要么使用事先验证的证明。




Interactive Proof System




Traditional Proof System

# 10.2 零知识证明的概念

## 1. 交互证明系统

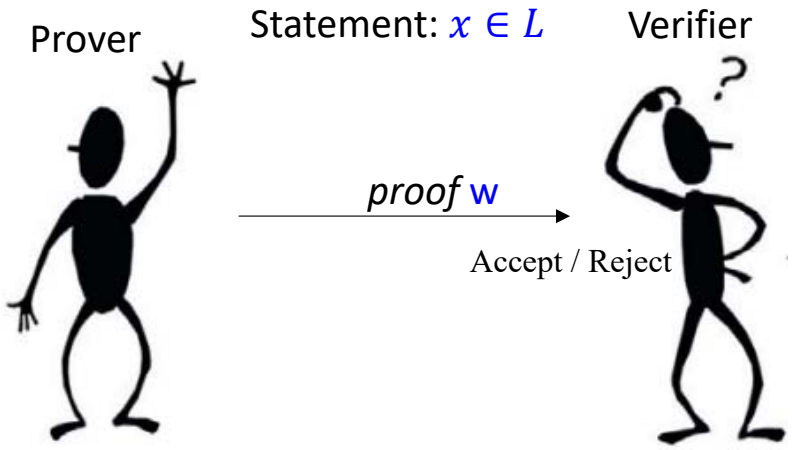
### ➢ 与传统证明系统的区别

相比而言，传统数学证明存在两个特点：

1）证明是短而简洁的。因为，验证者没有很多精力去读很长的证明。

2）验证者可能获取一些知识。譬如，证明者所采用的定理、证明的方式等等，一旦验证者获取了证明，那么它也可以向其他人证明。

交互证明系统可以改变上述两个特点，能够满足密码学的需求。




Traditional Proof System

# 10.2 零知识证明的概念

## 1. 交互证明系统

### ➢ IP语言类

拥有交互证明系统的语言类，称之为IP语言类。显然，可以看出：

- 如果任何语言$L$有一个传统证明系统，那么它必然有一个交互证明系统。

  因为，传统证明系统可以看成是一轮的交互证明系统，即$m_1 = w$。

- NP语言类属于IP语言类。

  NP语言类是属于多项式时间内可验证的，都有一个传统证明系统。

# 10.2 零知识证明的概念

## 1. 交互证明系统

### ➢ IP语言类

传统证明系统

NP问题的举例说明：

- 图同构

$$GI := \{(G_0, G_1) : \exists \phi \, s.t. \, \phi(G_0) = G_1\};$$

给出证据$\phi$

- 二次剩余

$$QR := \{(x, n) : \exists w \, s.t. \, x = w^2 mod n\};$$

给出证据$w$

- 哈密尔顿环

$$HC := \{G : G \text{ has a cycle visiting each vertex exactly once}\};$$

给出证据cycle

# 目　录

# 10.2 零知识证明的概念

## 2. 零知识证明

零知识证明系统是交互证明系统的一个实例。

零知识证明系统所要完成的任务是“证明某一个事实并且不泄露知识”。

简而言之，零知识证明是一个两方参与的交互协议，其中证明者向验证者证明某一个断言的正确性，并且满足以下三个条件：

（1）正确性，即若断言为真，则验证者总是接受证明；

（2）可靠性，即若断言为假，则验证者总是拒绝证明；

（3）零知识，即验证者无法从该证明过程中获取额外的信息。

零知识证明系统自身所具有的保密性和认证性正是密码学所追求的基本安全属性。

# 10.2 零知识证明的概念

## 2. 零知识证明

对于语言$L \in \{0,1\}^*$，以及一对交互图灵机$< P,V >$，其中$P$拥有无限的计算能力，称为证明者，$V$为概率多项式时间的验证者。

称$< P,V >$为语言$L$的零知识交互证明系统，如果满足以下条件：

- 完备性（Completeness）：对于任何公共输入$x \in L$，

$$\Pr[(P,V)(x) = 1 | x \in L] \leq 1 - negl(|x|).$$

- 可靠性（Soundness）：对于任意公共输入$x \notin L$和任意无限计算能力的证明者$P^*$，

$$\Pr[(P^*,V)(x) = 0 | x \notin L] \geq 1 - negl(|x|).$$

- 零知识（Zero-knowledge）：对任意概率多项式时间验证者$V^*$，都存在一个概率多项式时间的模拟器$S$，使得任意的$x \in L$，

$$< P,V^* > (x) \approx_c S(x).$$

其中，$negl(|x|)$为一个可忽略函数，$\approx_c$ 表示计算不可区分。

# 10.2 零知识证明的概念

## 2. 零知识证明

➢ 可靠性的两种形式

可靠性能够防止验证者相信不诚实的证明者（$x \notin L$）。

根据不诚实证明者的敌手能力，可分为：

- 计算可靠性（Computational zero-knowledge）

如果$x \notin L$，对任意的无限计算能力的$P^*$，验证者$V$接受$< P^*, V >(x)$的概率不大于1/3。对应的是交互论证系统。

- 统计可靠性（Statistical zero-knowledge）

如果$x \notin L$，对任意的多项式时间计算能力的$P^*$，验证者$V$接受$< P^*, V >(x)$的概率不大于1/3。对应的是交互证明系统。

。

# 10.2 零知识证明的概念

## 2. 零知识证明

➢ 零知识性的三种形式

零知识性能够防止证明者向验证者泄露不必要的信息。

根据模拟器的输出分布$S(x)$与真实协议的输出分布$< P, V^* > (x)$之间的关系，可分为：

- 计算零知识性（Computational zero-knowledge）

  两个分布是计算不可区分的，即没有有效的算法能够区分两个分布。

- 统计零知识性（Statistical zero-knowledge）

  两个分布是统计不可区分的，即它们的统计距离是可忽略的。

- 完美零知识性（Perfect zero-knowledge）

  两个分布是同分布的。

# 10.2 零知识证明的概念

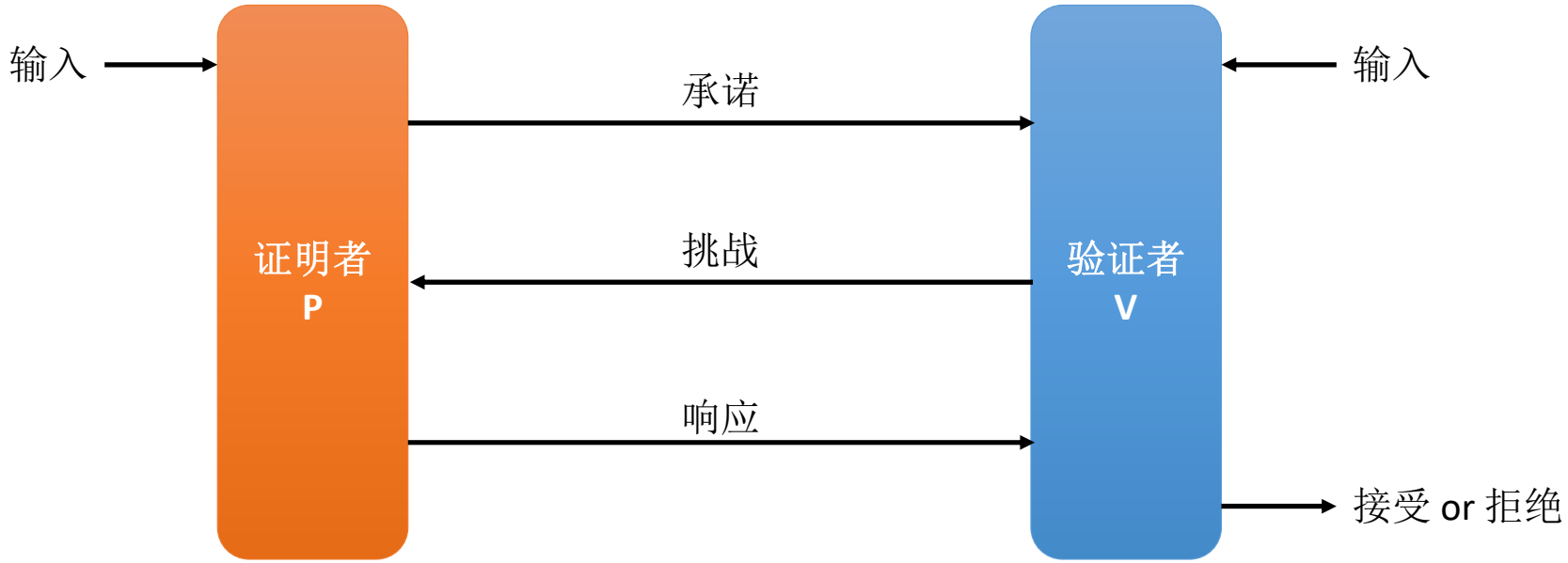
## 2. 零知识证明

- 辅助输入零知识（Auxiliary input zero-knowledge）
  - 。

# 目　录

# 10.3 经典交互式零知识证明

交互式零知识证明的一般模型




- 证明者和验证者共享一个公共输入，证明者可能拥有某个秘密输入。
- 如果验证者认可证明者的响应，则输入接收（Accept）；否则，输出拒绝（Reject）。

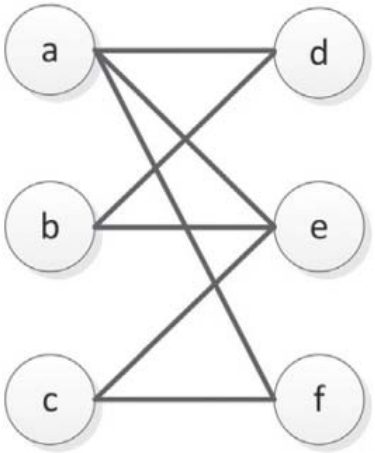
# 目　录

# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（1）——图同构问题

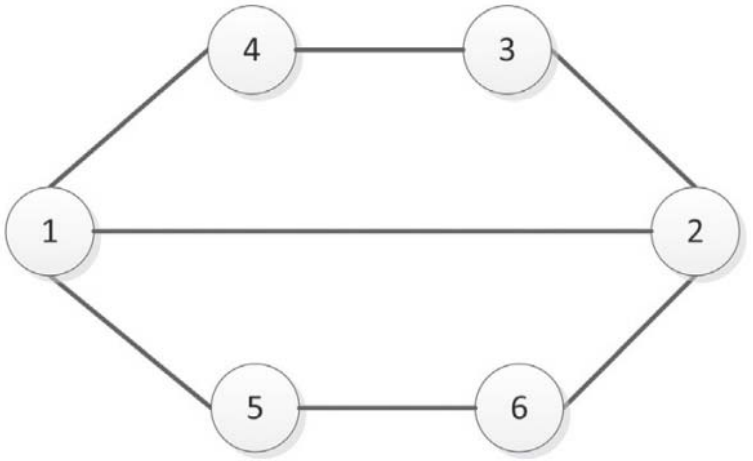
### ➢ 图同构（Graph Isomorphism）的定义

***Def.*** 两个图$G_0(V, E_0)$和$G_1\ (V, E_1)$是同构的，当且仅当存在有一个置换$\phi \in S_{|V|}$使得对于任意的$(u, v) \in E_0$仅有$\big(\phi(u), \phi(v)\big) \in E_1$。




$G_0(V, E_0)$

$$\phi: \begin{cases} \phi(a) = 1 \\ \phi(b) = 6 \\ \phi(c) = 3 \\ \phi(d) = 5 \\ \phi(e) = 2 \\ \phi(f) = 4 \end{cases}$$

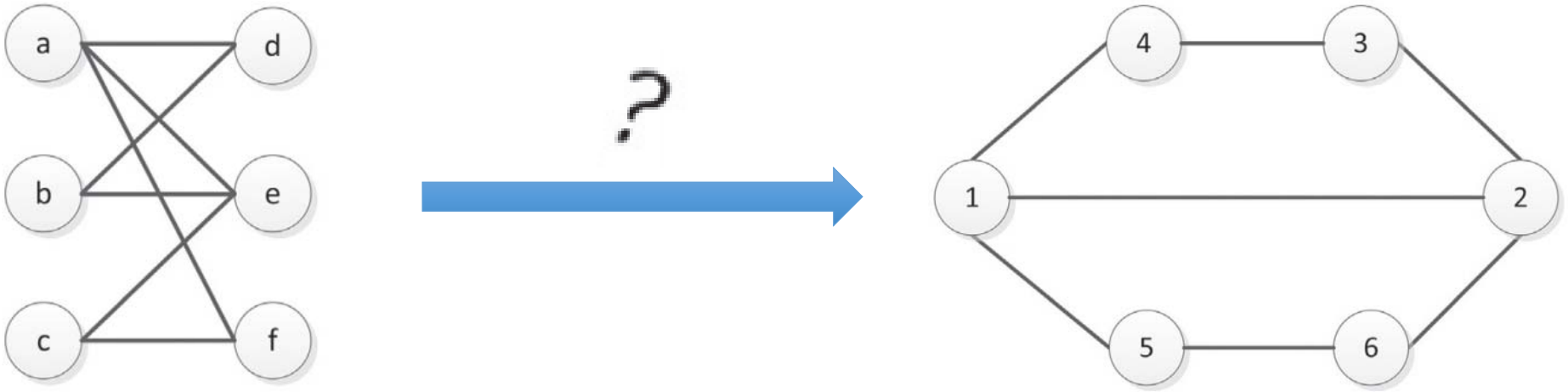



$G_1\ (V, E_1)$

# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（1）——图同构问题

**GI问题**：给定两个图$G_0$和$G_1$，判断两个图之间是否存在一个同构映射？




$G_0(V, E_0)$　　$G_1\ (V, E_1)$

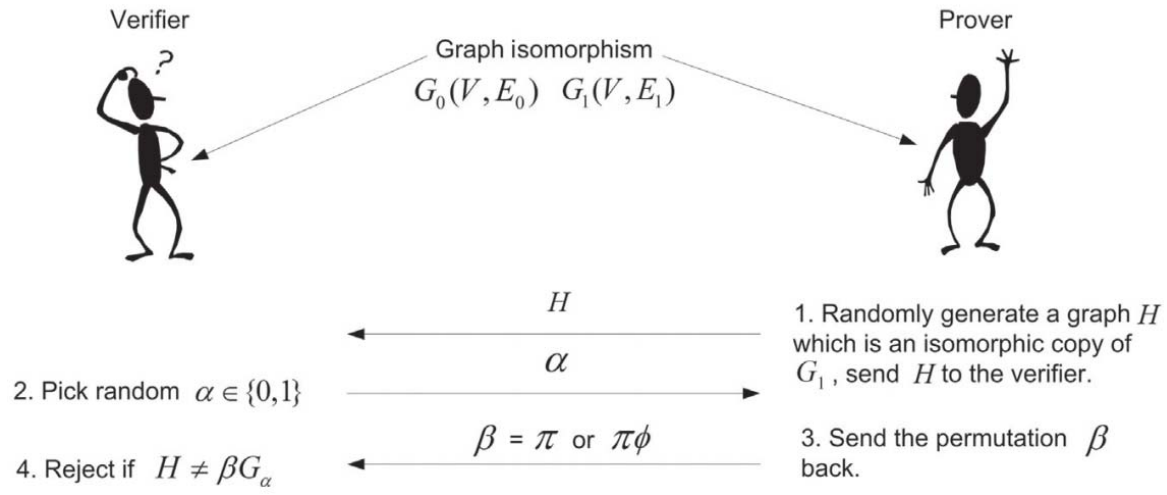
# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（1）——图同构问题

公共输入：两个同构的图$G_0(V,E_0)$和$G_1(V,E_1)$。

1）P: 证明者随机产生一个置换$\pi$，并计算图$H=\pi G_1$。然后证明者将$H$发送给验证者。

2）V: 验证者随机生成一个比特值$\alpha \in_R \{0,1\}$，并将$\alpha$发送给证明者。



# 10.3 经典交互式零知识证明

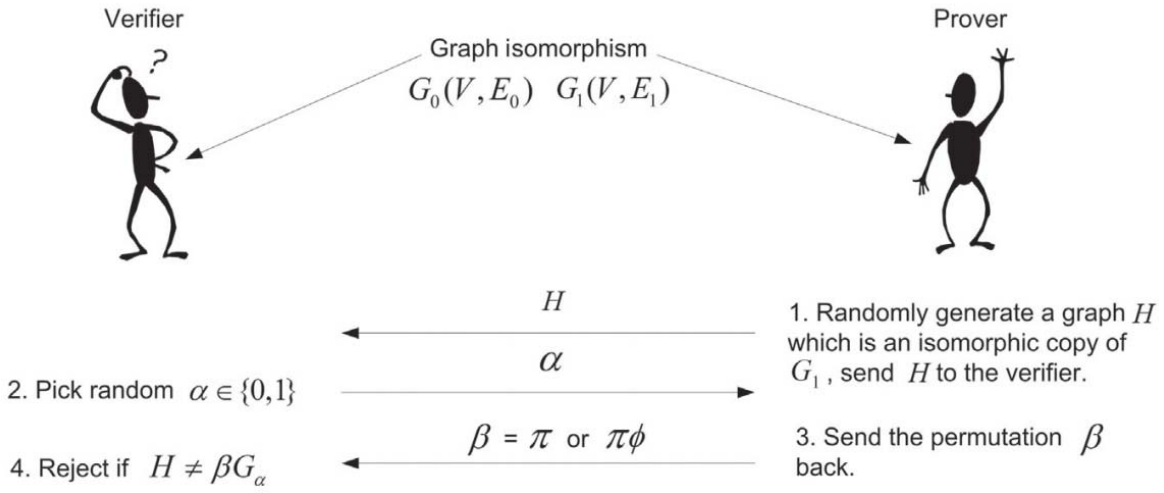
## 1. NP问题的零知识证明（1）——图同构问题

3）P: 证明者根据$\alpha$做出不同的响应：

- 如果$\alpha = 1$，证明者发送$\beta = \pi$给验证者；
- 如果$\alpha = 0$，证明者发送$\beta = \pi \cdot \phi$给验证者。

4）V: 验证者判断置换$\beta$是否是$H$和$G_\alpha$的同构。

如果不是，验证者拒绝接受证明；否则，验证者进入下一轮证明。




验证者执行上述证明过程$t$轮后，且均未拒绝，则验证者接受证明者的证明，即相信两个图是同构的。

# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（1）——图同构问题

### ➢ GI零知识证明性质分析

◆ 完备性：

如果$(G_0, G_1) \in GI$，即存在映射$\phi$使得$G_1 = \phi G_0$，那么当$\alpha = 0$时，$\beta G_\alpha = \pi\phi G_0 = H$；当$\alpha = 1$时，$\beta G_\alpha = \pi G_1 = H$。显然，诚实验证者接收的概率为1.

◆ 可靠性：

如果$(G_0, G_1) \notin GI$，那么$H$只可能是$G_0$或者$G_1$中某个的同构图。诚实验证者每轮拒绝的概率为1/2.

◆ 零知识性：

可知的是，可以泄露的信息只有$\pi$或$\pi \cdot \phi$。由于$\pi$是随机产生的，模拟器可以模拟证明者和验证者之间的交互，并且交互信息与真实交互是计算不可区分的。

# 10.2 经典交互式零知识证明

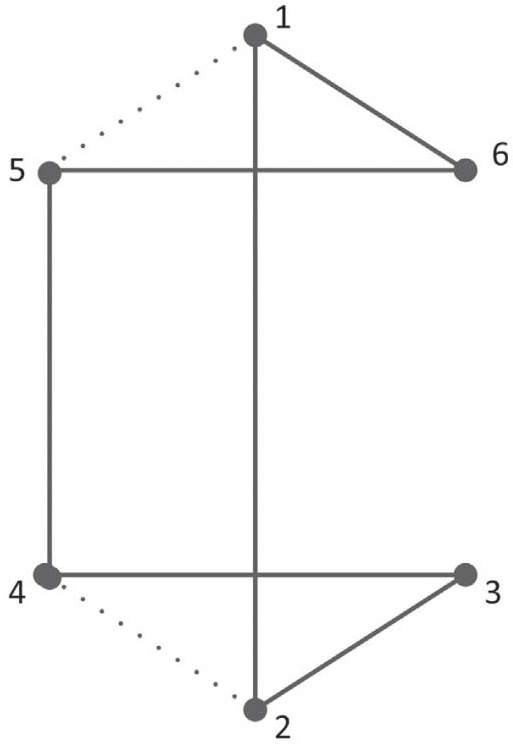
## 1. NP问题的零知识证明（2）——哈密顿回路问题

### ➢ 哈密顿回路的定义

***Def.*** 哈密顿回路（Hamiltonian Cycle）是指一个图$G$中一个回路，途中经过图中所有节点且只经过一次。记$n$个节点分别为$N_1, N_2, \cdots, N_n$。

### ➢ 哈密顿难题

***HC Problem.*** 给定一个图G，找出该图的一个哈密顿回路。




哈密顿回路示例

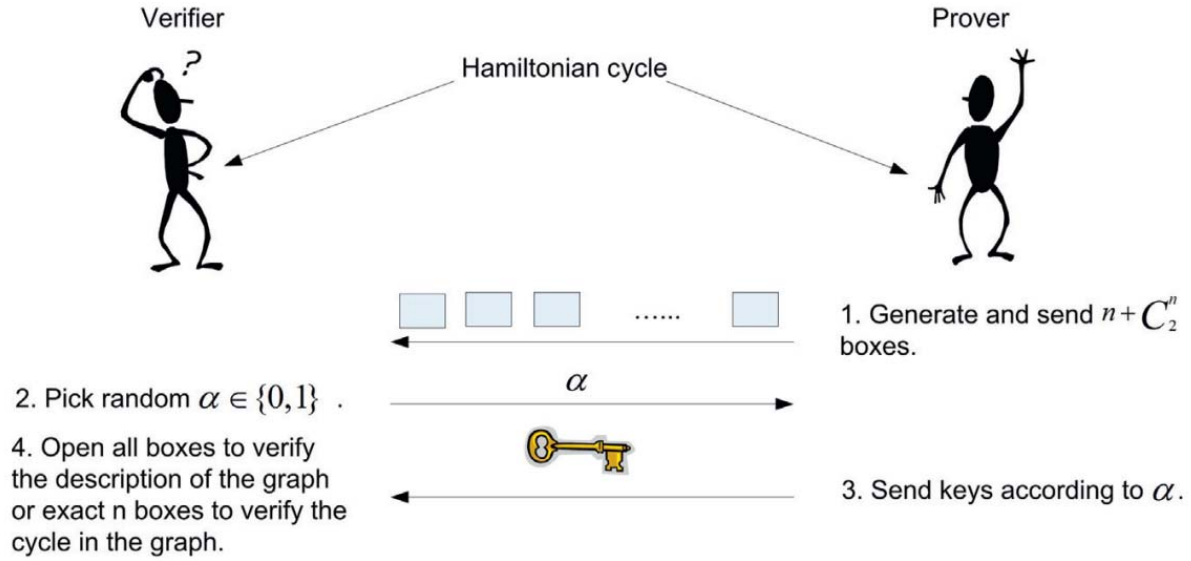
# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（2）——哈密顿回路问题

公共输入：一个无向图$G$，顶点数量为$n$。

1）P: 证明者随机加密$n$个顶点，即产生一个随机置换将$N_1, N_2, \cdots, N_n$映射成为$B_1, B_2, \cdots, B_n$。之后，设置$B_{ij}$的值，如果$(B_i, B_j)$是一条边，则$B_{ij} = 1$；否则$B_{ij} = 0$。证明者将$B_{i(1 \le i \le n)}$和$B_{ij(1 \le i < j \le n)}$放入$(n + C_n^2)$个黑盒中，并发送给验证者。

2）V: 验证者随机生成一个比特值$\alpha \in_R \{0,1\}$，并将$\alpha$发送给证明者。

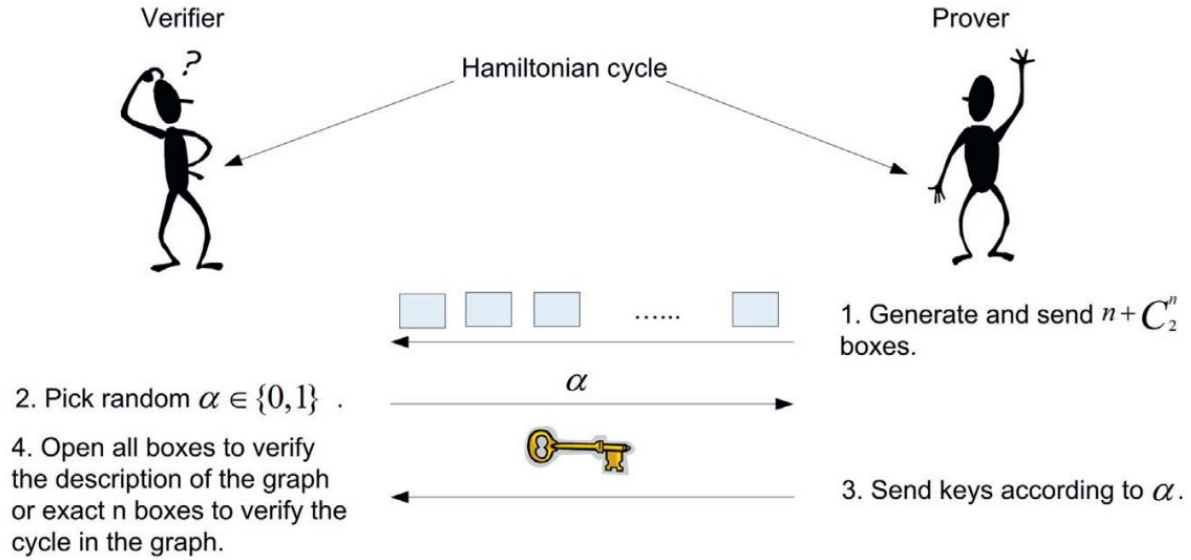


# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（2）——哈密顿回路问题

3）P: 证明者根据$\alpha$做出不同的响应：

- 如果$\alpha = 0$，证明者将打开所有黑盒的钥匙以及置换发送给验证者；
- 如果$\alpha = 1$，证明者将一个打开HC回路$(B_{ij}, B_{jk}, B_{kl}, \cdots, B_{ti})$ 的钥匙发送给验证者。



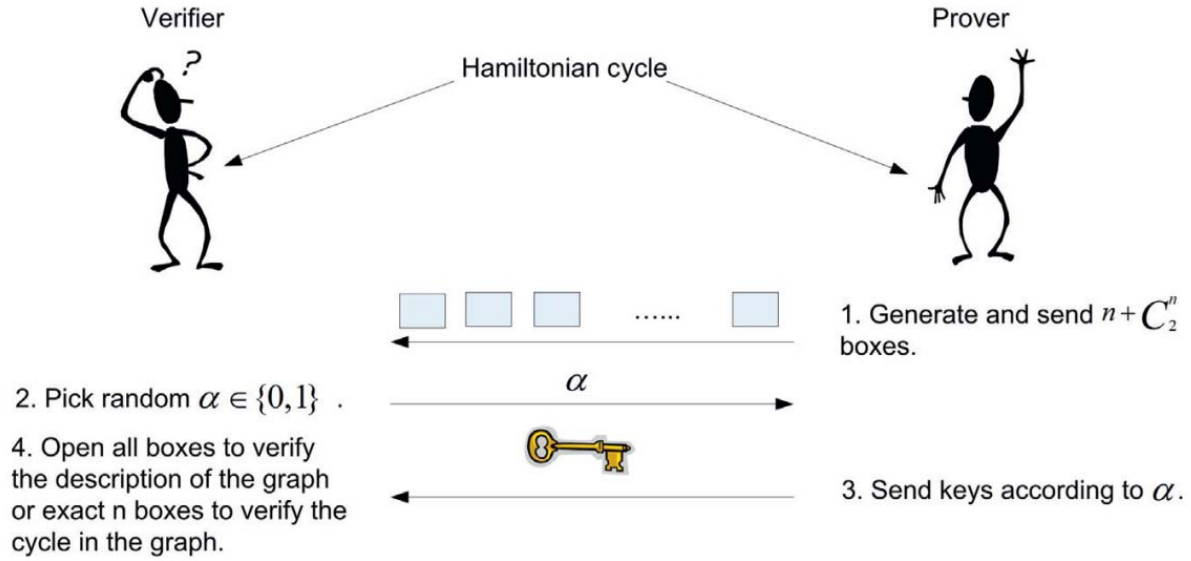
# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（2）——哈密顿回路问题

4）V: 验证者根据$\alpha$做出不同的验证：

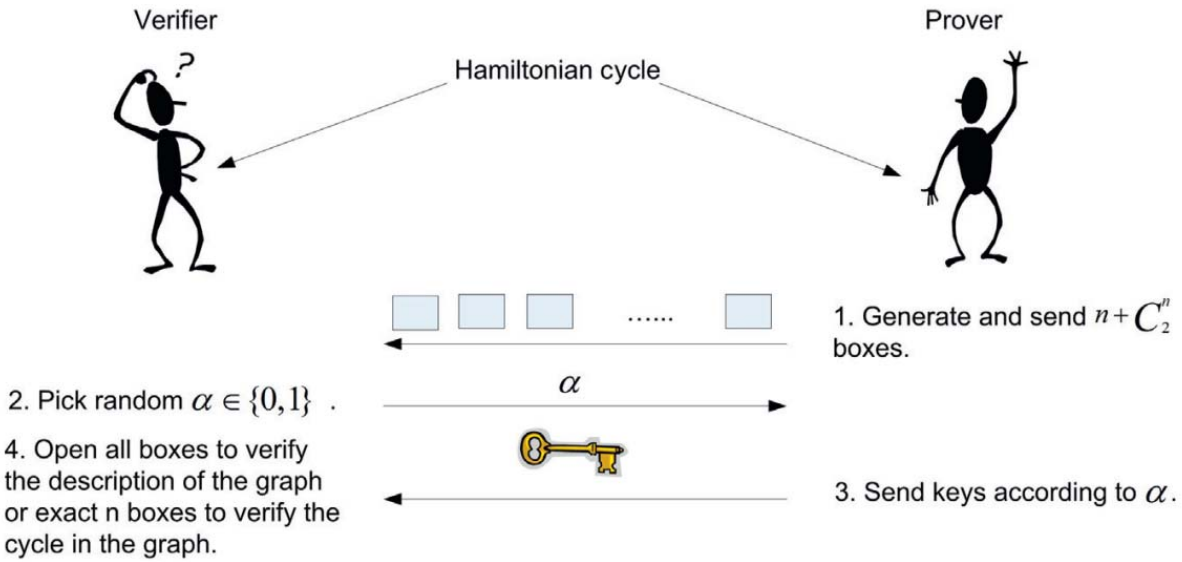
- 如果$\alpha = 0$，验证者打开所有黑盒，并验证其是否为图G的一个同构；如果不是，则拒绝；
- 如果$\alpha = 1$，验证者打开指定的黑盒，验证是否所有打开的$B_{ij}$都是1。如果不全是，则拒绝。

如果验证者未拒绝，则继续下一轮证明。



# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（2）——哈密顿回路问题




问题1：步骤1中的黑盒可以产生多少种？

答：$n!$ 个。如果考虑概率加密方式实现黑盒，那么黑盒的情况会更多。

问题2：当$\alpha = 1$时，是否会泄露真实的HC？

答：非常低，因为所有的$B_i$未泄露，验证者无法知道给出的HC环路在图$G$中的真实环路。

# 10.3 经典交互式零知识证明

## 1. NP问题的零知识证明（2）——哈密顿回路问题

### ➢ HC零知识证明性质分析

◆ 完备性：

如果证明者知道图$G$的哈密顿回路，他对于不同的$\alpha$都可以做出正确的响应。显然，诚实验证者接收的概率为1。

◆ 可靠性：

如果证明者不知道图$G$的哈密顿回路，那么他只能够以1/2的概率欺骗验证者。这个概率可以理解为证明者预先猜测成功验证者$\alpha$值的概率。即诚实验证者每轮拒绝的概率为1/2.

◆ 零知识性：

可知的是，当$\alpha = 0$时，V获取的仅仅是一个$G$的同构；当$\alpha = 1$时，V确实得到了$B_i$对应图的哈密顿回路，但是$B_i$与$N_i$的映射并未泄露。所以，V仍然无法获取$G$的哈密顿回路。

# 目　录

# 10.3 经典交互式零知识证明

## 2. 身份鉴别协议

在一个安全的身份认证协议中，要保证用户身份识别的安全性，身份鉴别协议至少要满足以下条件：

1）证明者P能够向验证者V证明他的确是P（P向V证明自己有P的私钥）。

2）在证明者P向验证者V证明他的身份后，验证者V没有获得任何有用的信息（V不能模仿P向第三方证明他是P ）。

*思考：用数字签名对V提供的随机数签名是否安全的证明身份？*

*答案：否，例如中间人攻击，或者让证明者对一段文件进行签名（而证明者完全不知道他对某个文件进行了签署）。*

# 10.3 经典交互式零知识证明

## 2. 身份鉴别协议

该协议的目的是证明者P向验证者V证明他的身份（私钥），且事后V不能冒充P。

后续介绍三种经典的身份鉴别协议：

1）Feige-Fiat-Shamir身份鉴别协议

2）Guillo-Quisquater身份鉴别协议

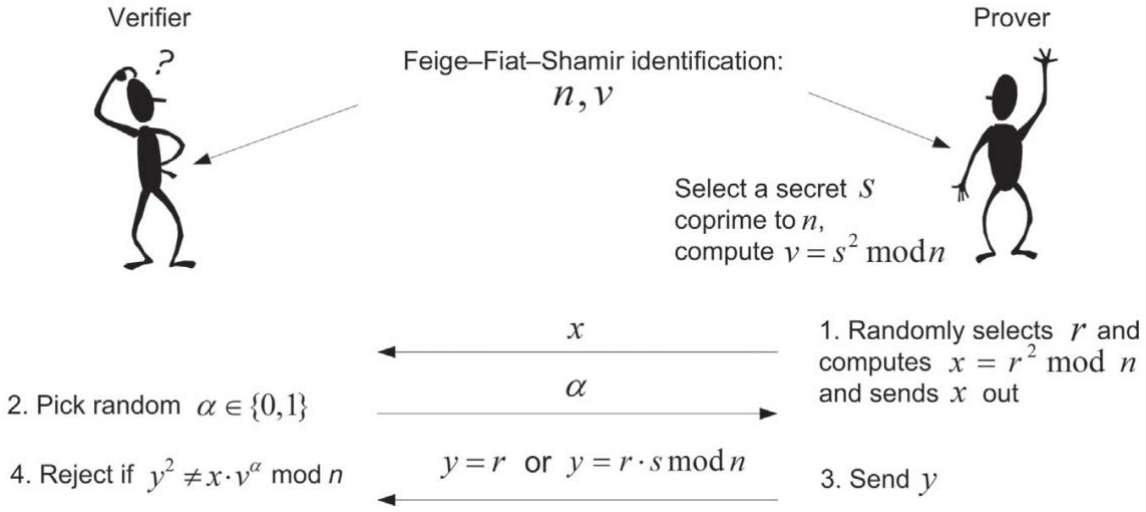
3）Schnorr身份鉴别协议

# 10.3 经典交互式零知识证明

## 2. 身份鉴别协议——Feige-Fiat-Shamir鉴别协议

1986年，Feige、Fiat和Shamir基于零知识的思想设计了一个零知识身份鉴别协议，这就是著名的 Feige-Fiat-Shamir 零知识身份鉴别协议。

该协议的目的是证明者P向验证者V证明他的身份（私钥），且事后V不能冒充P。



# 10.3 经典交互式零知识证明

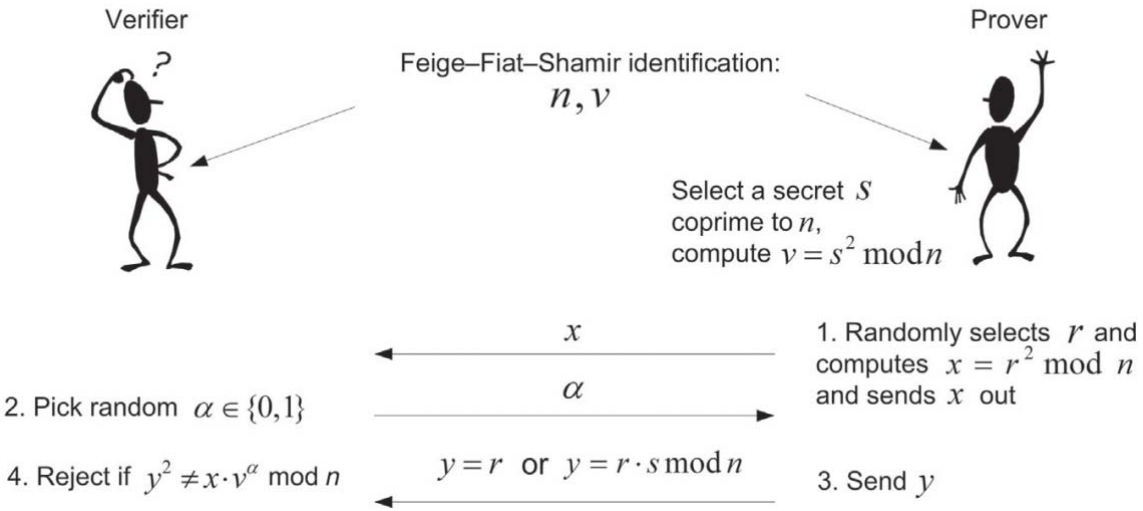
## 2.身份鉴别协议—— Feige-Fiat-Shamir鉴别协议

➢ 简化版本

系统初始化（一次性）：

① 信任中心TA选择并公布一个RSA型模数 $n = p \cdot q$，并对素数$p$和$q$保密。

② 每一个参与者P选择一个与$n$互素的秘密值$s$，$1 \leq s \leq n-1$，并计算$v = s^2 (\mathrm{mod}\, n)$，并向TA注册$v$为其公钥。



# 10.3 经典交互式零知识证明

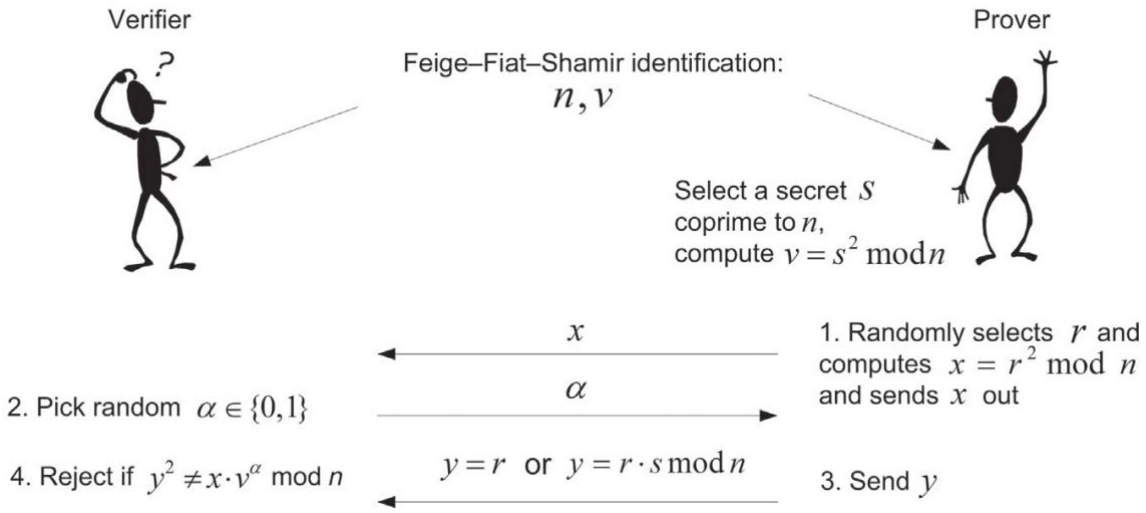
## 2.身份鉴别协议—— Feige-Fiat-Shamir鉴别协议

➢简化版本

**鉴别协议流程：**

1）P选择一个随机数$r$，$1 \leq r \leq n-1$，计算并发送$x = r^2 (\mathrm{mod}\, n)$给V。

2）V随机选择一个比特值$\alpha \in_R \{0,1\}$，并将$\alpha$发送给证明者。



# 10.3 经典交互式零知识证明

## 2.身份鉴别协议—— Feige-Fiat-Shamir鉴别协议

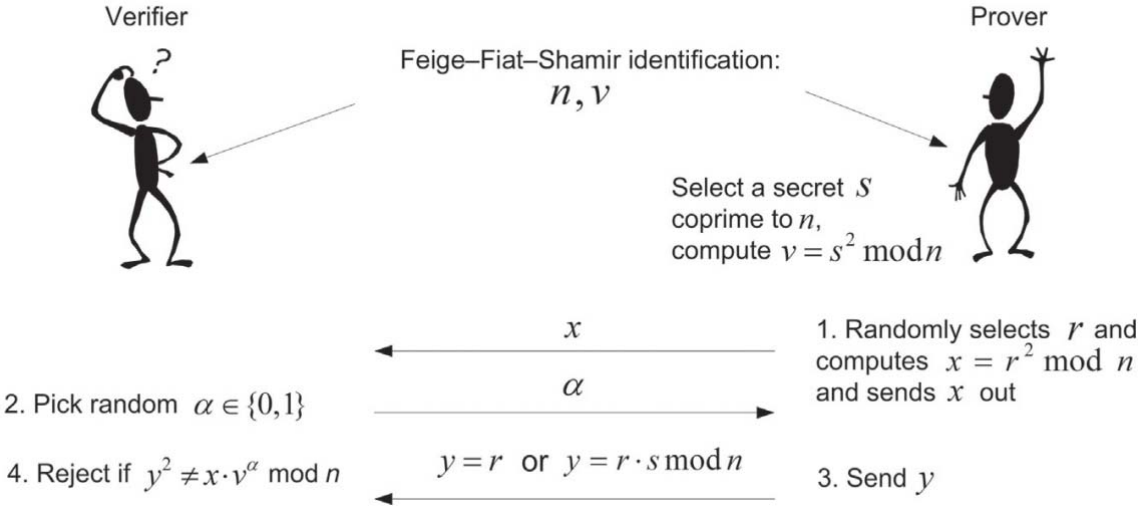
➢简化版本

鉴别协议流程：

3）P根据$\alpha$做出不同的响应：

若$\alpha = 0$，令$y = r$；

若$\alpha = 1$，令$y = r \cdot s \pmod n$。




4）V验证等式$y^2 = x \cdot v^\alpha \pmod n$。如果等式不成立或者$y = 0$，V不接受证明。否则，进行下一轮证明。

验证者V执行上述证明过程$t$轮后，且均未拒绝，则验证者接受证明者P的证明，即相信他的身份。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议—— Feige-Fiat-Shamir鉴别协议

### ➢ 简化版本的性质分析

◆ **完备性：**

如果证明者知道秘密值$s$，他对于不同的$\alpha$都可以做出正确的响应。显然，诚实验证者接收的概率为1。

◆ **可靠性：**

如果证明者不知道秘密值$s$，那么他只能够以1/2的概率欺骗验证者。执行$t$轮后，欺骗概率下降到$2^{-t}$。

◆ **零知识性：**

协议交互过程中泄露的信息有：$x = s^2 (mod\ n)$、$y = r$ 或$y = r \cdot s (mod\ n)$，即$(x, y)$。

模拟器的模拟方式：随机选择$y$，并令$x = y\text{^}2$（当$\alpha = 0$）或$x = y^2/v$（当$\alpha = 1$）。

模拟器产生的$(x, y)$与真实交互的$(x, y)$是计算不可区分的。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议—— Feige-Fiat-Shamir鉴别协议

➢ 完整版本

**系统初始化（一次性）：**

① 信任中心TA选择并公布一个RSA型模数$n = p \cdot q$，并对素数$p$和$q$保密。

② 每一个参与者P选择k个与$n$互素的秘密值$s_1, s_2, \cdots, s_k$，$1 \leq s_i \leq n-1$，并计算$v_i = s_i^2 (\mathrm{mod}\, n)$，并向TA注册$(v_1, v_2, \cdots, v_k)$为其公钥。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Feige-Fiat-Shamir鉴别协议

➢ 完整版本

**鉴别协议流程：**

1）P选择一个随机数$r$，$1 \leq r \leq n-1$，计算并发送$x = r^2 (\mathrm{mod}\, n)$给V。

2）V随机选择$k$个比特值$\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_k) \in_R \{0,1\}^k$，并将$\vec{\alpha}$发送给P。

3）P计算$y = r \cdot \prod_{i=1}^{k} s_i^{\alpha_i} (\mathrm{mod}\, n)$，并将$y$发送给V。

4）V验证等式$y^2 = x \cdot \prod_{i=1}^{k} v_i^{\alpha_i} (\mathrm{mod}\, n)$。如果等式不成立或者$y = 0$，V不接受证明。否则，进行下一轮证明。

验证者V执行上述证明过程 $t$轮 后，且均未拒绝，则验证者接受证明者P的证明，即相信他的身份。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Feige-Fiat-Shamir鉴别协议

### ➢ 完整版本的性质分析

◆ **完备性：**

显然，诚实验证者接收的概率为1。

◆ **可靠性：**

如果证明者不知道秘密值$s$，那么他只能够以$1/2^k$的概率欺骗验证者。执行$t$轮后，欺骗概率下降到$2^{-k\cdot t}$。

◆ **零知识性：**

同样地，与简化版本一致，交互数据元组$(x,y)$可以被模拟器模拟，达到计算不可区分性。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Feige-Fiat-Shamir鉴别协议

### ➢安全假设

无论简化版本还是完整版本，协议的安全性依赖于未知分解的大合数的模平方根求解难题。这个问题等价于大合数的分解困难问题。

### ➢参数选择

以完整版本为例，$k \cdot t$（简化版本中$k = 1$）需要足够大，才能够保证非诚实证明者欺骗成功概率几乎可忽略。同时，$n$的分解困难性也是安全性考虑之一。

### ➢安全平衡

每增加一轮协议，计算量和通信量均上升，但安全性越高。因此，需要在保证足够安全的前提下，减少协议重复轮数$t$，提升效率。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Guillo-Quisquater身份鉴别协议

Guillo和Quisquater给出了一个身份鉴别方案，这个协议需要可信第三方参与、三轮交互、利用RSA公钥密码体制。

**系统初始化（一次性）：**

1）信任中心TA选择并公布一个RSA型模数$n = p \cdot q$，并对素数$p$和$q$保密。令$\phi = (p-1)(q-1)$，选取$e$使得$\gcd(e, \phi) = 1$，计算$d = e^{-1}(mod\ \phi)$，公开$(e, n)$。

2）用户P选取唯一性身份$I_P$，通过哈希函数变换得到哈希值$J_P = H(I_P)$,使得 $1 \le J_P \le n$，$\gcd(\phi, J_P) = 1$。TA给A分配密钥$S_P = (J_P)^{-d}(mod\ n)$。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Guillo-Quisquater身份鉴别协议

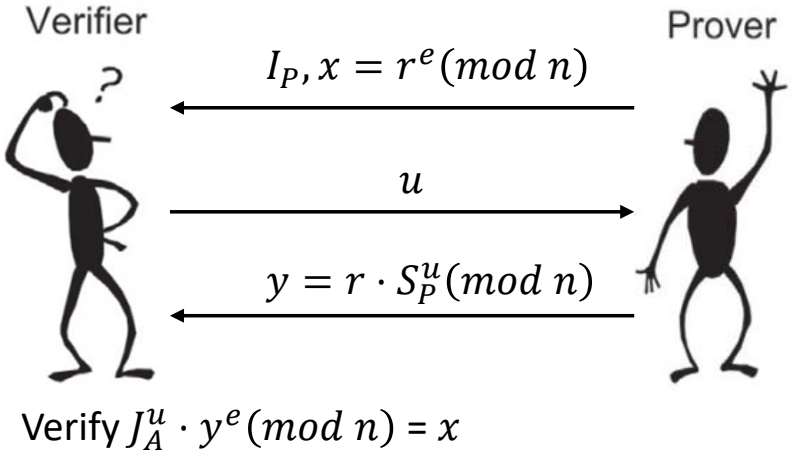
鉴别协议流程：

1）P产生随机数$r$，$1 \le r \le n-1$，计算$x = r^e (mod\ n)$，并将$(I_P, x)$发送给V；

2）V选取随机数$u$，$1 \le u \le e$，将$u$发送给P。

3）P计算$y = r \cdot S_P^u (mod\ n)$，并将y发送给V。

4）V计算$J_P = H(I_P)$，并验证$J_A^u \cdot y^e (mod\ n)$不为0且等于$x$。如果成立，则此轮接收证明；否则，输出拒绝。



# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Guillo-Quisquater身份鉴别协议

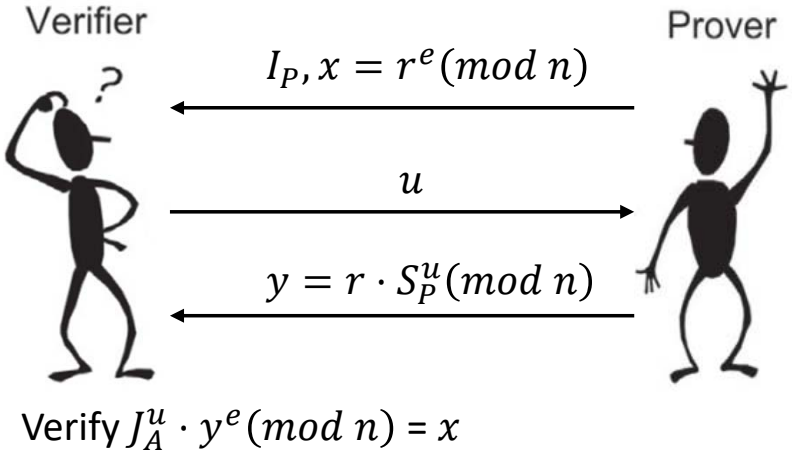
### ➢协议性质分析

◆ **完备性：**显然，诚实验证者接收的概率为1。

◆ **可靠性：**如果证明者不知道秘密值$S_P$，那么他只能够以$1/e$的概率欺骗验证者。执行$t$轮后，欺骗概率下降到$e^{-t}$。

◆ **零知识性：**可被模拟器模拟，达到计算不可区分性。

如果$e$比较大时，每轮的错误概率就会很低，那么需要重复执行的轮数$t$就可以很小，甚至为1（即$e$为$k$比特素数）。



# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Schnorr身份鉴别协议

Schnorr提出了一种基于离散对数困难问题的身份鉴别协议，可以做预计算来降低实时通信量，所传输的数据量也减少许多，特别适合于计算能力有限的情况。

- 安全假设

离散对数问题是计算困难的。

- 参数选取

可信的第三方TA首先选取系统参数：选取素数$p$和$q$，满足$q|p-1$；选取$g$为$q$阶生成元，满足$g^q = 1 \ mod \ p$；选取一个小整数$t$作为安全参数，满足$2^t < q$。

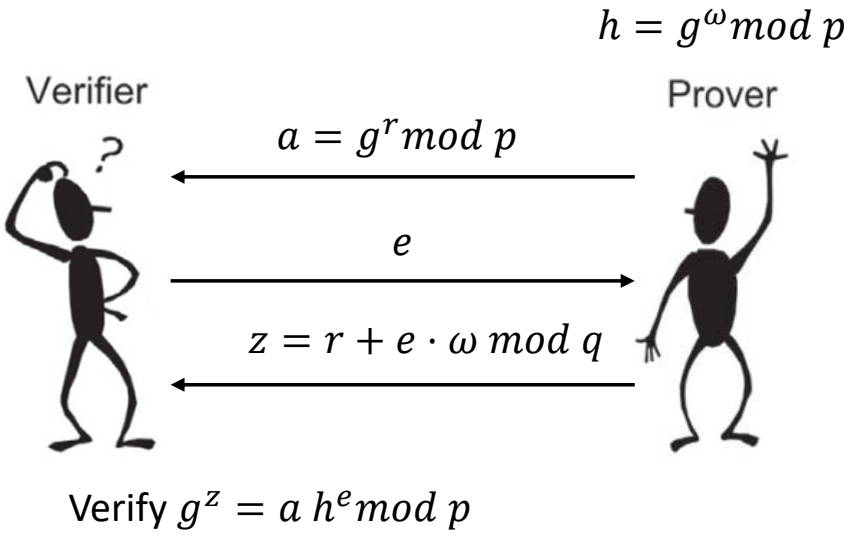
TA向各用户发布系统参数$(p, q, g, t)$。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Schnorr身份鉴别协议

**用户注册流程：**

用户P随机选取一个$\omega$（$1 \le s \le q-1$）作为秘密值，并计算$h = g^{\omega} mod\ p$。之后，用户P向TA注册并发布公钥$h$。




**鉴别协议流程：**

1）P随机选择$r$，$1 \le r \le q-1$，计算并发送$a = g^{r} mod\ p$给V；

2）V随机选择$e$，$1 \le e \le 2^{t}$，发送$e$给P；

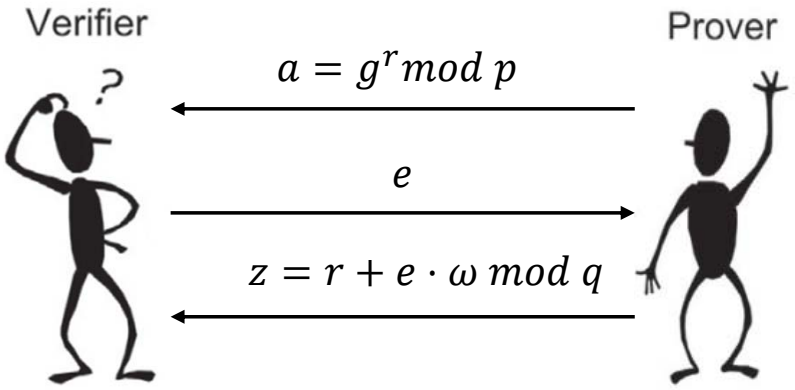
3）P计算并发送$z = r + e \cdot \omega\ mod\ q$给V；

4）V验证$g^{z} = a\ h^{e} mod\ p$是否成立。若成立，则V接受；否则，V拒绝。

# 10.3 经典交互式零知识证明

## 2.身份鉴别协议——Schnorr身份鉴别协议

$h = g^{\omega} mod\ p$




Verify $g^z = a\ h^e mod\ p$

### ➢协议性质分析

- **完备性**：显然，诚实验证者接收的概率为1。
- **可靠性**：如果证明者不知道秘密值$\omega$，那么他只能够以$2^{-t}$的概率欺骗验证者。
- **零知识性**：可被模拟器模拟，达到计算不可区分性。

如果$t$比较大时，每轮的错误概率就会非常低，几乎可以忽略不计。

后面，将以Sigma协议簇形式对Schnorr协议进行安全分析。

# 目　录

# 10.3 经典交互式零知识证明

### 3. Sigma协议簇

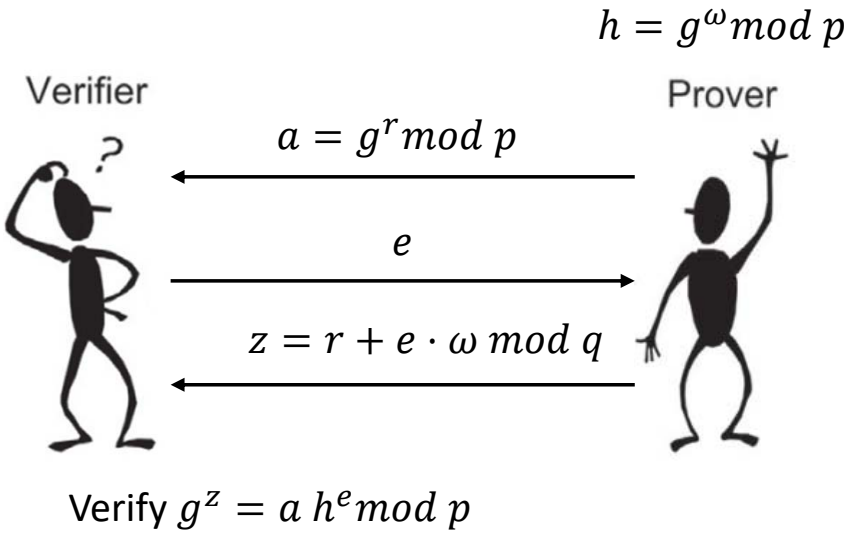
以Schnorr协议为例，进行如下分析：

1）假设P对同一个承诺$x$进行了两次响应，分别是$(a, e_1, z_1)$和$(a, e_2, z_2)$满足$e_1 \neq e_2$ 和$z_1 \neq z_2$。

那么有，$g^{z_1} = x\, h^{e_1}$和$g^{z_2} = x\, h^{e_2}$。相除可得，

$h = g^{(z_1 - z_2)(e_1 - e_2)^{-1}} \ mod\ p$。

这意味着，$\omega = (z_1 - z_2)(e_1 - e_2)^{-1}$，也就是说任何人可以通过公开交互信息获取证明者的私钥$\omega$。




理解：如果具备这一性质，那么一个不诚实的证明者至多只能伪造一个可以被接受的响应。

因为，如果他能够伪造多于1个的响应，他必然知道秘密值$\omega$ 。

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇

以Schnorr协议为例，进行如下分析：

2）不诚实的证明者P*如何成功欺骗验证者V呢？

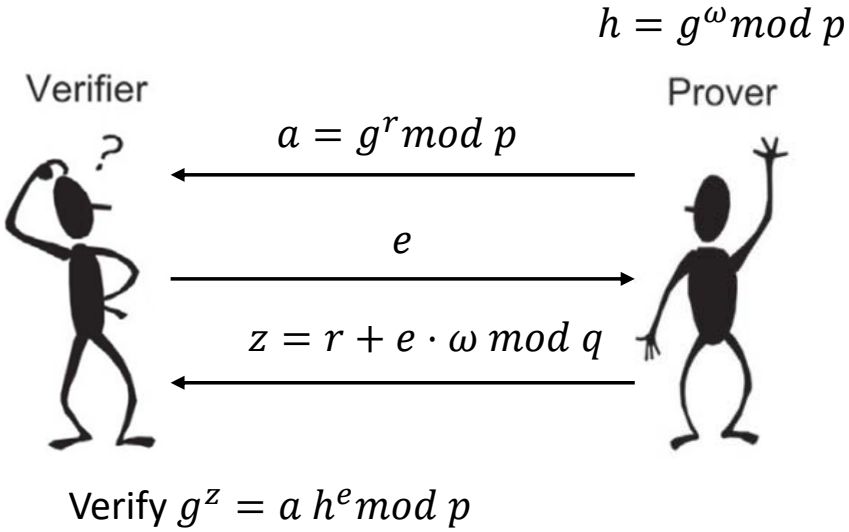
方法一：通过$h$来获取$\omega$，相当于破解离散对数问题。

方法二：猜测验证者的挑战值$e$，而$e$的产生是均匀随机的，其猜测碰对的概率为$2^{-t}$。

理解：安全的协议参数需要满足以下两点：

A）$q$必须足够大，能够抵抗离散对数攻击（涵盖了对$s$的穷举猜测）；

B）$t$必须足够大，使得协议的错误概率非常低（一般无需重复运行）。



# 10.3 经典交互式零知识证明

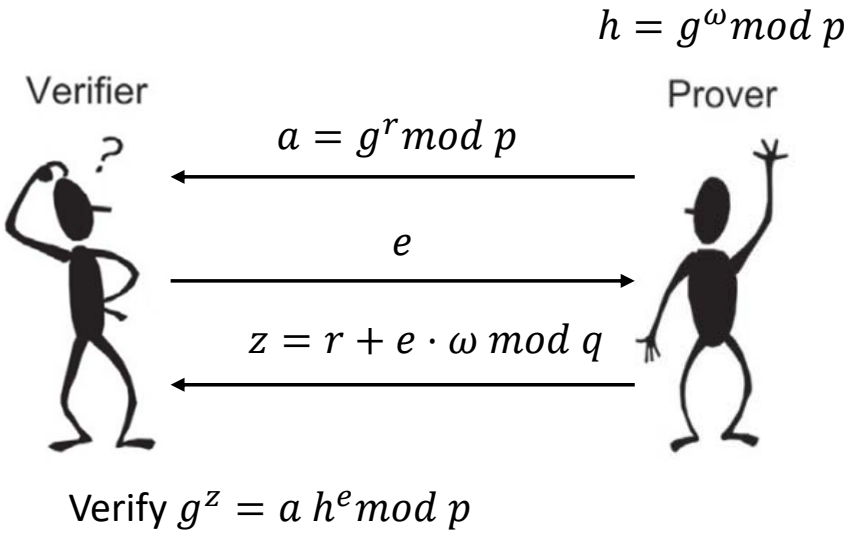
## 3. Sigma协议簇

以Schnorr协议为例，进行如下分析：

3）模拟器如何能够模拟一个交互元组$(x, e, y)$？

模拟器随机选择$z \in Z_p^*$和$e \in Z_q$，计算$a = g^z h^{-e} \bmod p$。

显然，模拟器产生的元组与真实交互产生的元组是计算不可区分的。




理解：模拟器与零知识性之间的关系？

注意，模拟器不具备秘密值$\omega$，但拥有一个能力决定或者预知验证者的挑战值e。如果，模拟元组与真实元组是计算不可区分的，那么任何PPT敌手（包括验证者）都无法区分元组是由谁产生的。这也意味着，敌手不可能获取知识$\omega$。因为，敌手有可能从模拟元组中获取$\omega$，这是矛盾的。

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇——形式化定义

### ➢ 二元关系

***Def.*** 记$\boldsymbol{R}$是一个二元关系（**binary relation**）即$\boldsymbol{R}$是$\{0,1\}^* \times \{0,1\}^*$的子集，存在某些约束条件只有当$(x,\omega) \in \boldsymbol{R}$时成立。其中，$\omega$的长度最多是$x$的一个多项式。

一般而言，$x$是某些问题或陈诉的实例（Instance），而$\omega$是这些问题的解或者证据（Witness）.

举例说明：

$\{(x,w)|\ x=(p,q,g,h), ord(g)=ord(h)=q, h=g^w\}$,　　$\{(x,w)|\ x=(n,q,y), y,w \in Z_n^*, q\ prime,\ and\ y=w^q \bmod n\}$.

$\{(x,w)|\ x=(p,q,g,\bar{g},h,\bar{h})\ and\ h=g^w, \bar{h}=\bar{g}^w\}$.　　$\{(x,(w_1,w_2))|\ x=(p,q,g_1,g_2,h)\ and\ h=g_1^{w_1}g_2^{w_2}\}$.

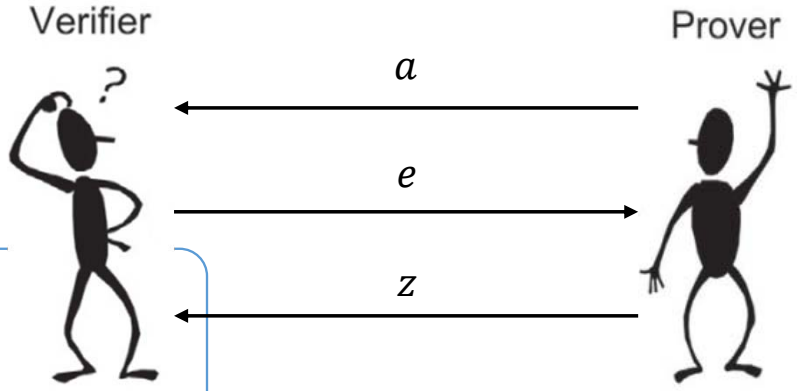
如果relation是困难的,一般称为困难关系（Hard Relations）。

例如，NP问题、RSA问题、离散对数问题等等。

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇——形式化定义

### ➢ 二元关系




错误概率：$2^{-t}$

***Def.*** **【Sigma协议】**协议$\boldsymbol{P}$称为关系$\boldsymbol{R}$的三轮形式的Sigma协议，满足以下性质：

1）完备性：令$x$为公共输入，$\omega$为证明者P的私有输入。若$(x,\omega)\in R$，则验证者V总是接受P的证明。

2）特殊可靠性（Special Soundness）：对于任意x，任意两个被接收的元组$(a,e_1,z_1)$和$(a,e_2,z_2)$且$e_1\neq e_2$，则可以有效的计算出$\omega$。

3）特殊诚实验证者零知识性（Special honest-verifier zero-knowledge）：存在一个多项式时间的模拟器M，输入x和随机的e，能够有效输出一个元组$(a,e,z)$，且该元组的分布于P、V之间的交互元组分布概率相同。

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇

### ➢ 可扩展性

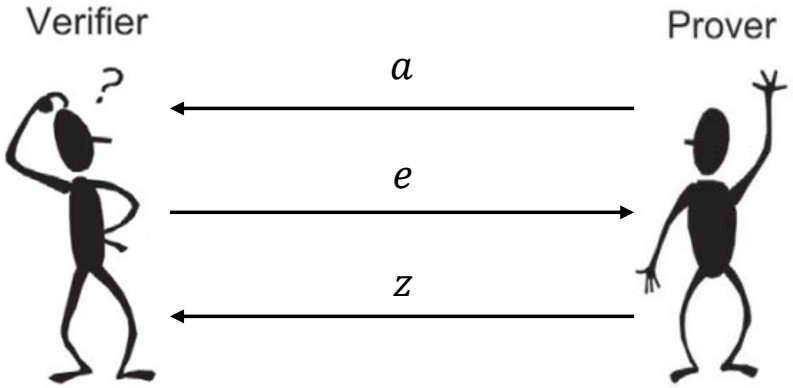
两种方式可以将挑战长度扩展或压缩：

**1）并行方式（parallel composition）**

例如：承诺$a_1$和$a_2$，挑战$e$的长度为$2t$；将$e$分成两段后，分别响应$z_1$和$z_2$。

**2）零填充（zero padding）**

例如：需要挑战长度为$t_1 < t$，生成$e$的时候，只产生$t_1$个随机比特，剩下$t - t_1$个比特用全零填充。



# 10.3 经典交互式零知识证明

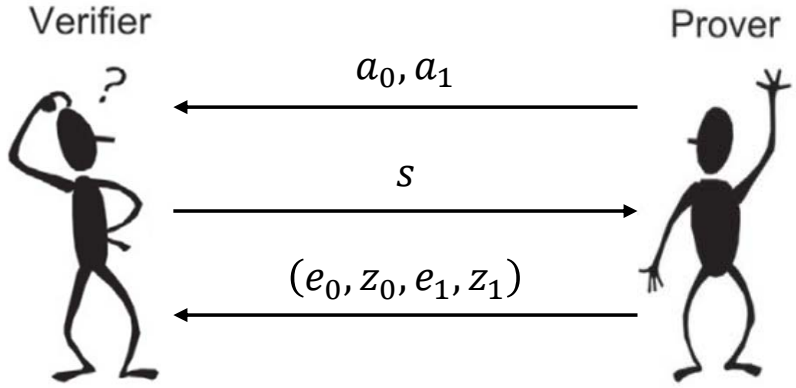
## 3. Sigma协议簇

### ➢ OR证明

证明描述：给定两个输入$x_0$和$x_1$，证明者P需要向验证者V证明他知道一个证据$w$使得$(x_0, w) \in R$或者$(x_1, w) \in R$，并且不泄露$w$与哪个输入匹配。

构造方式：

- 以sigma协议为基础构造；
- 利用模拟器的特性。




OR证明协议

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇

### ➢ OR证明

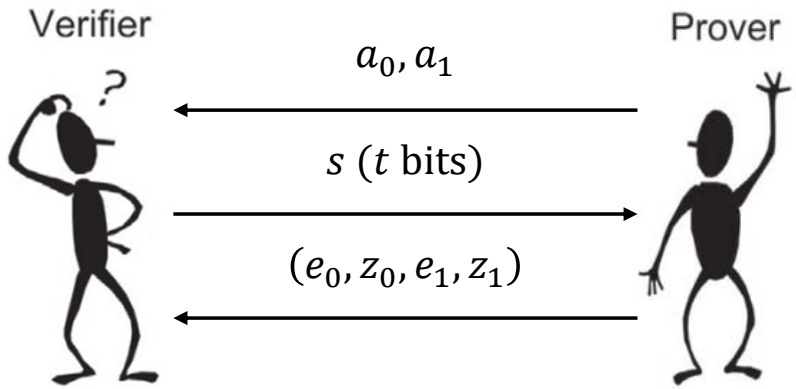
证明描述：给定两个输入$x_0$和$x_1$，证明者P需要向验证者V证明他知道一个证据$w$使得$(x_0, w) \in R$或者$(x_1, w) \in R$，并且不泄露$w$与哪个输入匹配。

协议交互过程如下：

1）P利用sigma协议的第一步生成$a_b$（假定$x_b$为$w$对应的实例）；之后，P运行模拟器M产生一组元组$(a_{1-b}, e_{1-b}, z_{1-b})$；

P将$(a_0, a_1)$发送给V；

2）V产生t比特的随机挑战s并发送给P；




OR证明协议

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇

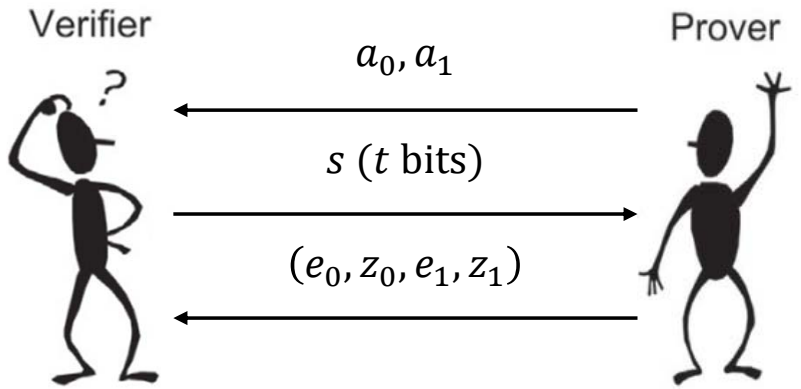
### ➢ OR证明

证明描述：给定两个输入$x_0$和$x_1$，证明者P需要向验证者V证明他知道一个证据$w$使得$(x_0, w) \in R$或者$(x_1, w) \in R$，并且不泄露$w$与哪个输入匹配。

协议交互过程如下：

3）P令$e_b = s \oplus e_{1-b}$，再运行sigma协议的第三步（以$e_b$为挑战值）得到$z_b$，之后将$(e_0, z_0, e_1, z_1)$发送给V；

4）V验证$s = e_0 \oplus e_1$，并运行sigma协议的第四步，验证元组$(a_0, e_0, z_0)$和$(a_1, e_1, z_1)$是否被接受。如果等式成立且元组均被接受，则V接受；否则拒绝。




OR证明协议

# 10.3 经典交互式零知识证明

## 3. Sigma协议簇

### ➢ OR证明的安全性

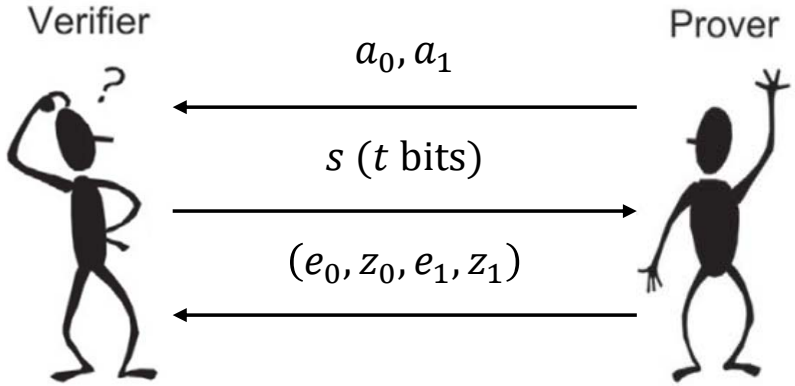
完备性 + 特殊可靠性 + 特殊诚实验证者零知识性 均满足；

OR证明协议也是一个sigma协议。

另外，该协议还具备一个性质：

证据不可区分性（Witness Indistinguishability, WI）

即任何敌手都无法区分证明者使用了哪个证据$w_0$或$w_1$。




OR证明协议

# 10.3 经典交互式零知识证明

### 3. Sigma协议簇

#### ➢ 证据隐藏性（Witness Hiding，WH）

此外，再考虑一个问题。对于一个给定的实例$x$，是否可以从$x$获取一个$w$且满足$(x,w) \in R$？

那么，需要将$R$所对应的约束条件进行区分：

1）Easy Relations

此类问题指存在多项式敌手可以从$x$中计算得到$w$。转换为陈诉，x可以是一个自证命题。

2）Hard Relations

此类问题指不存在多项式敌手能够从$x$中计算得到$w$。一般对应的是，数学困难问题（DLP等）。

Hard Relations可以提供WH。因为，对于多项式时间的验证者而言，他无法从$x$中获取$w$。

# 目　录

# 10.4 非交互式零知识证明

## 1. Fiat-Shamir转换

回顾下Sigma协议，可以发现以下问题：

1）sigma协议的零知识性需要验证者是诚实的；

那么，验证者不诚实的话，sigma协议是否是零知识的？目前，无法说明这个问题。

2）sigma协议的安全基石除了困难问题外，还有什么？

答案是随机挑战。在可靠性分析过程中，可以看出随机挑战值是决定错误概率的关键。

3）sigma协议的效率如何？限制有哪些？

显然，计算量和通信量受限于协议的构造基石（如困难问题、挑战轮数等等）。而且，交互形式的协议限制了它的使用场景。

# 10.4 非交互式零知识证明

## 1. Fiat-Shamir转换

交互式零知识证明 —— 是否可行？ ——> 非交互式零知识证明（NIZK）

1.只能够取信于一个验证者；

2. 只在某个时刻有效；

1. 可以取信于很多人（甚至所有人）；

2. 证明始终有效；

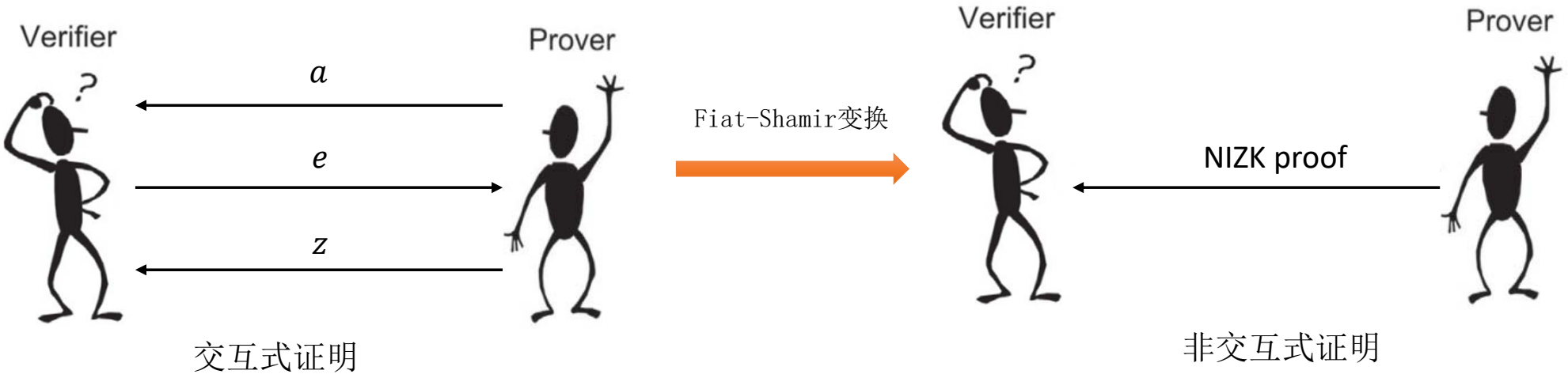
NIZK: non-interactive zero-knowledge

# 10.4 非交互式零知识证明

## 1. Fiat-Shamir转换

Fiat-Shamir变换是一种可以将Sigma协议变成非交互证明的技术。它能够让证明者Prover可以通过给验证者Verifier发送一个证明信息即可完成证明(无需交互，无需返回挑战)。

而且，它能把任何一个Simga协议变成一个数字签名，签名的含义就是“知道这个Sigma协议的秘密的人已经签署了这个消息”。Prover能够创造一个证明，然后分发给很多个验证者，验证者可以不必联系Prover即可验证证明有效性。同时零知识也变得容易了，因为验证者或者其他敌手不能做任何事情。



# 10.4 非交互式零知识证明

## 1. Fiat-Shamir转换

以Schnorr协议为例，如果挑战值e可预知，那么任何人都可以成功欺骗V。

在交互证明时，挑战值可以由验证者V随机产生；

但在非交互证明时，如何保障挑战随机性和不被证明者控制？

为例重建信任，Fiat-Shamir转换采用了随机预言机，即让“上帝”来选取挑战值。

问题又来了，真实的随机语言机是不存在。于是，密码学家作出了如下假设：

> 假设：一个密码学安全的 Hash 函数可以近似地模拟传说中的「随机预言机」。

注意：这个假设无法被证明，所以我们只能信任这个假设，或者说当做一个公理来用。
Hash 函数的广义抗碰撞性质决定了它的输出可以模拟随机数，同时在很多情况下（并非所有），对 Hash 函数实施攻击难度很高，于是许多的密码学家都在大胆使用。

# 目　录

# 10.4 非交互式零知识证明

## 2. NIZK的发展情况

# 目　录

# 10.5 零知识证明在区块链中的应用

## 1. 区块链隐私保护背景

◆身份匿名保护

**假名特性**

- 网络分析
- 地址聚类
- 交易图分析

◆交易数据隐私

**公开可验**

- 资金关系
- 资金流向
- 交易规律

◆合约代码隐私

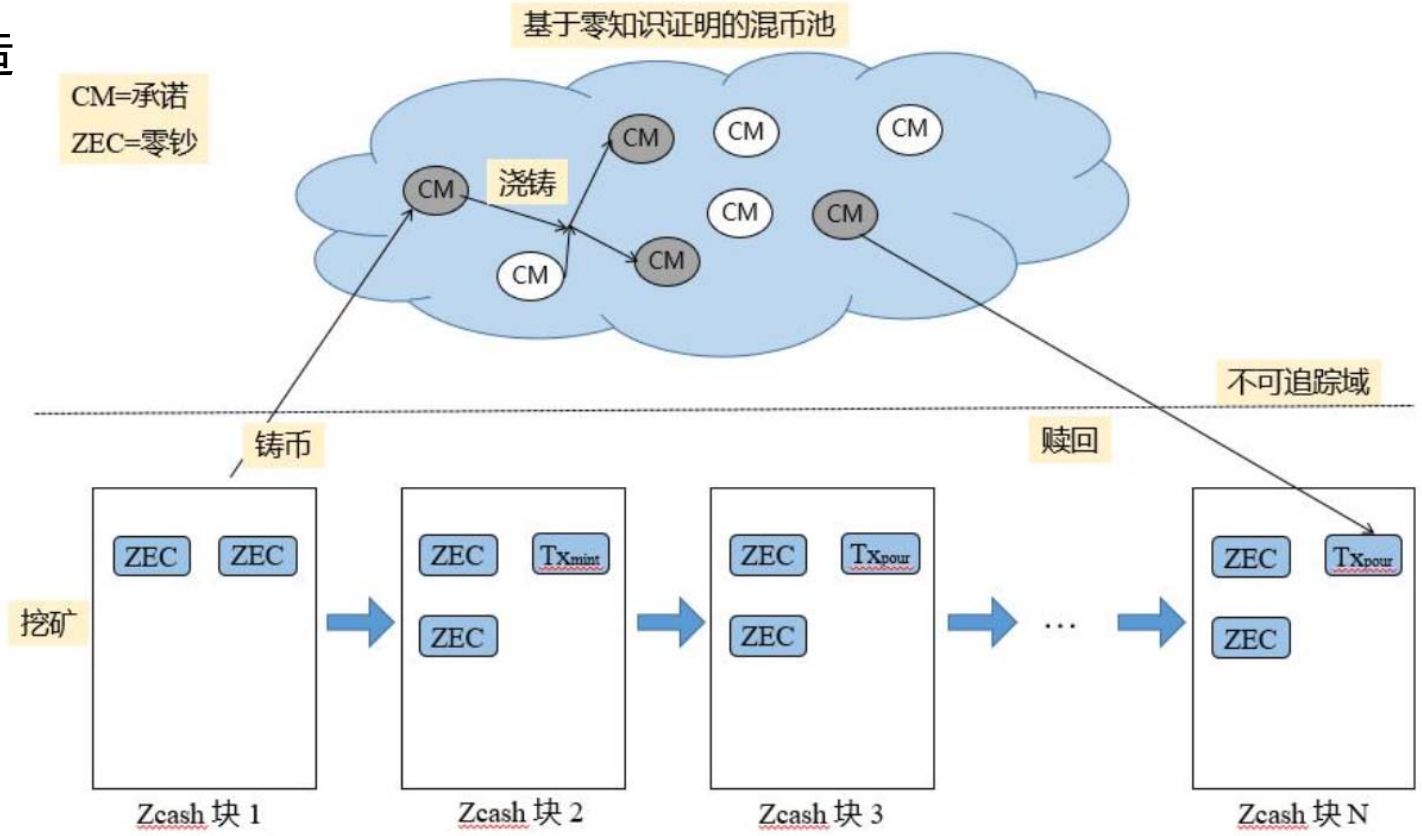
**全网透明**

- 反编译
- 参数泄露
- 各类漏洞

# 10.5 零知识证明在区块链中的应用

## 2. ZeroCash (零钞)

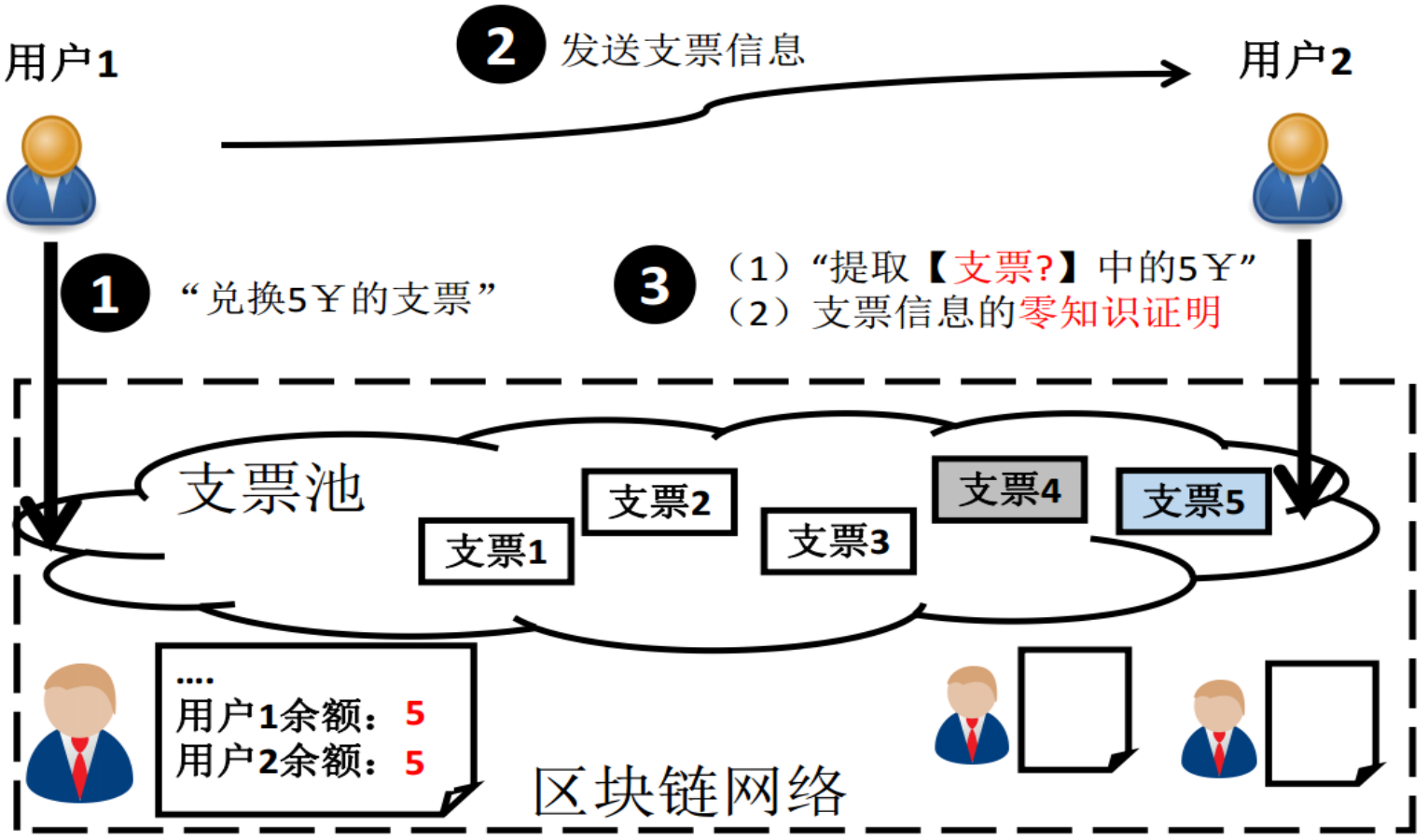
- 基于zkSNARK构造的混币池
- 隐私性强



# 10.5 零知识证明在区块链中的应用

## 2. ZeroCash (零钞)

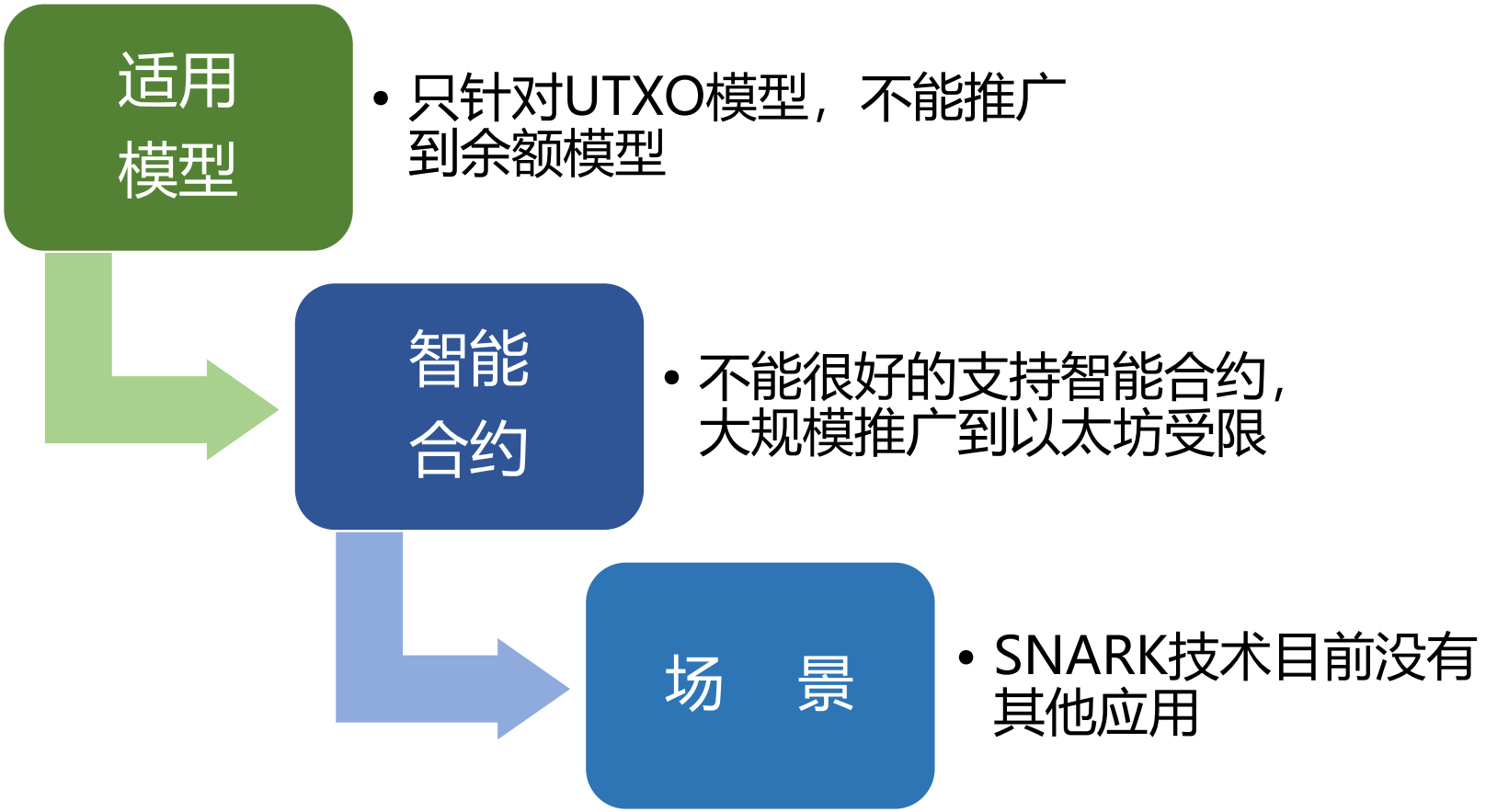
**运行原理：**



# 10.5 零知识证明在区块链中的应用

## 2. ZeroCash (零钞)

**面临挑战：**



# 10.5 零知识证明在区块链中的应用

## 3. Mimblewimble

- **工具：**同态承诺、聚合签名、零知识范围证明(Bulletproofs)；
- **地址：** $C = vH + rG$ ，密钥：$(v, r)$
- **交易格式：** $\left(s, \widehat{\boldsymbol{C}}_{\boldsymbol{i}}, \widehat{\boldsymbol{C}}_{\boldsymbol{o}}, \pi, E, \sigma\right)$，其中， $s$为挖矿金额，$\widehat{\boldsymbol{C}}_{\boldsymbol{i}} = (C_{i1}, \dots, C_{im})$为输入地址列表，$\widehat{\boldsymbol{C}}_{\boldsymbol{o}} = (C_{o1}, \dots, C_{on})$为输出地址列表， $\pi$为范围证明， $E = \Sigma_j C_{ij} - \Sigma_k C_{ok} = (r_{i1} + \cdots + r_{im} - r_{o1} - \cdots - r_{on})G$为公钥， $\sigma$为聚合签名（私钥为$r_{i1} + \cdots + r_{im} - r_{o1} - \cdots - r_{on}$）
- **账本格式：** $\left(\boldsymbol{s}, \widehat{\boldsymbol{C}}_{\boldsymbol{o}}, \boldsymbol{\pi}, \boldsymbol{E}, \boldsymbol{\sigma}\right)$：其中， $\boldsymbol{s}$为所有挖矿金额总和，$\widehat{\boldsymbol{C}}_{\boldsymbol{o}}$为所有的输出地址列表，$\boldsymbol{\pi}$为所有交易的范围证明集合， $\boldsymbol{E}$为所有交易的公钥集合， $\boldsymbol{\sigma}$为聚合签名集合。备注：每笔交易的输入一定会是账本的输出列表，由于直通性，账本中没有输出列表。

# 10.5 零知识证明在区块链中的应用

## 3. Mimblewimble

**交易过程：**

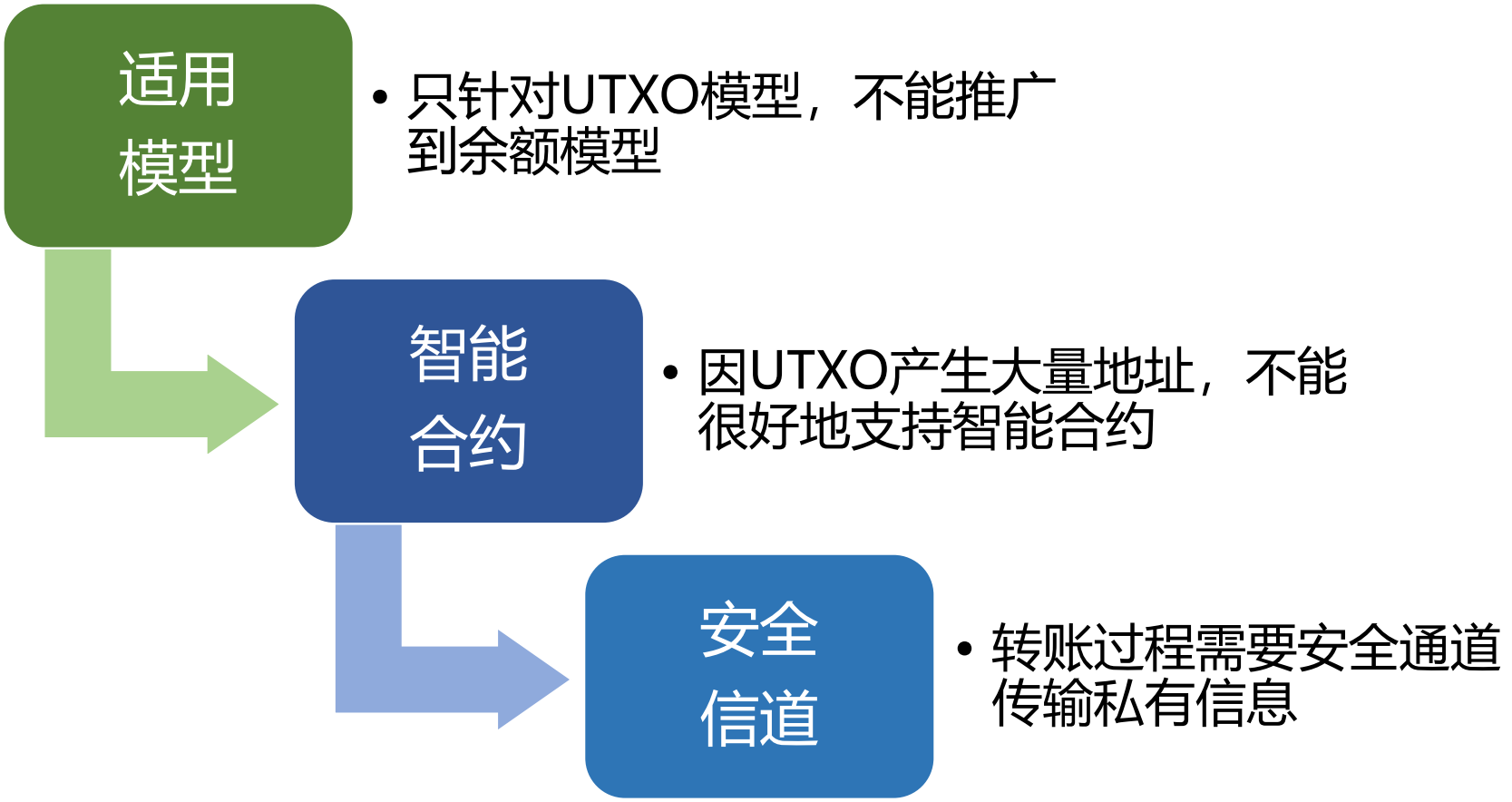
| $Sender$ | | $Receiver$ |
|---|---|---|
| 1. 确定挖矿奖励$s$，选取输入地址列表$\widehat{\boldsymbol{C}}_{\boldsymbol{i}}$，找零地址列表$\widehat{\boldsymbol{C}}_{\boldsymbol{o}}$，临时地址$C = pG + rH$，计算范围证明$\pi$，公钥信息$E$以及签名$\sigma$.<br>2. $pTx = (s, \widehat{\boldsymbol{C}}_{\boldsymbol{i}}, \widehat{\boldsymbol{C}}_{\boldsymbol{o}}, C, \pi, E, \sigma)$ | $pTx, p, r$ → | 选取接收地址列表$\boldsymbol{C}_{\boldsymbol{r}}$，计算交易$pTx' = (C, C_r, \pi', E', \sigma)$，利用直通性，最终得到交$tx = (s, \widehat{\boldsymbol{C}}_{\boldsymbol{i}}, \widehat{\boldsymbol{C}}_{\boldsymbol{o}}, C_r, \{\pi, \pi'\}, (E + E'), \{\sigma, \sigma'\})$ |

**直通性：**给定两笔交易$tx_1 = (C_1, C_2, C_3, \pi, E, \sigma)$和$tx_2 = (C_2, C_4, \pi', E', \sigma')$可直接得到$tx = (C_1, C_2, C_4, \{\pi, \pi'\}, E + E', \{\sigma, \sigma'\})$，因此能够有效隐藏中间交易细节。

# 10.5 零知识证明在区块链中的应用

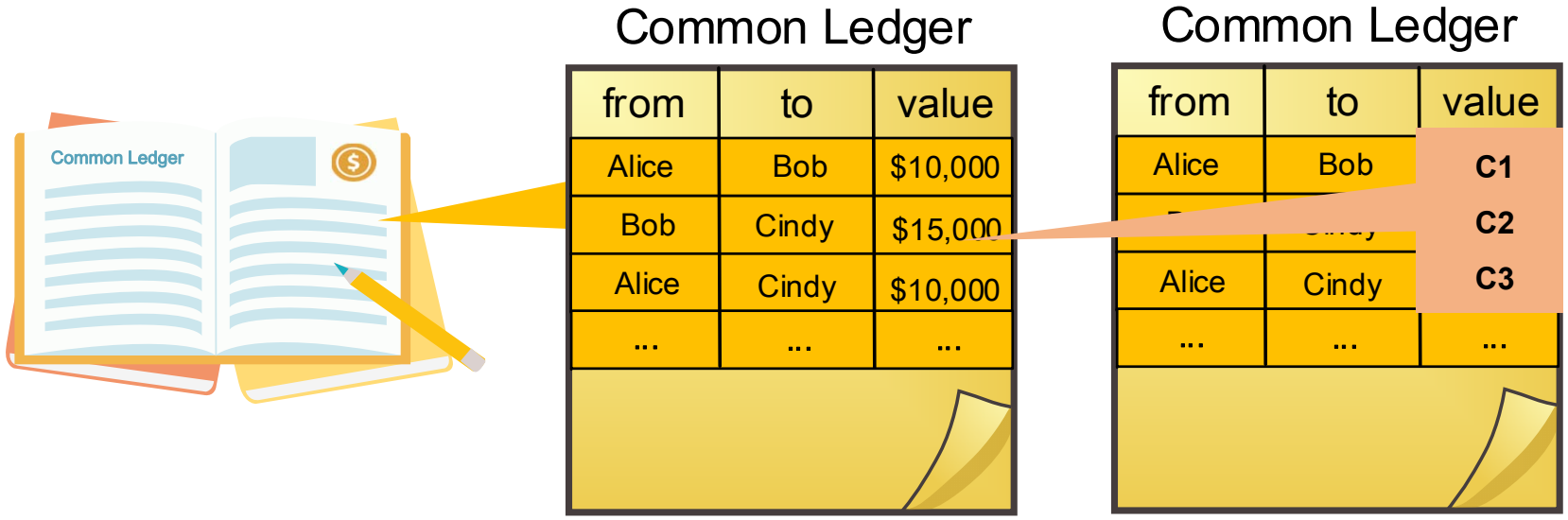
## 3. Mimblewimble

**面临挑战：**



# 10.5 零知识证明在区块链中的应用

## 4. 基于同态加密的NIZK协议



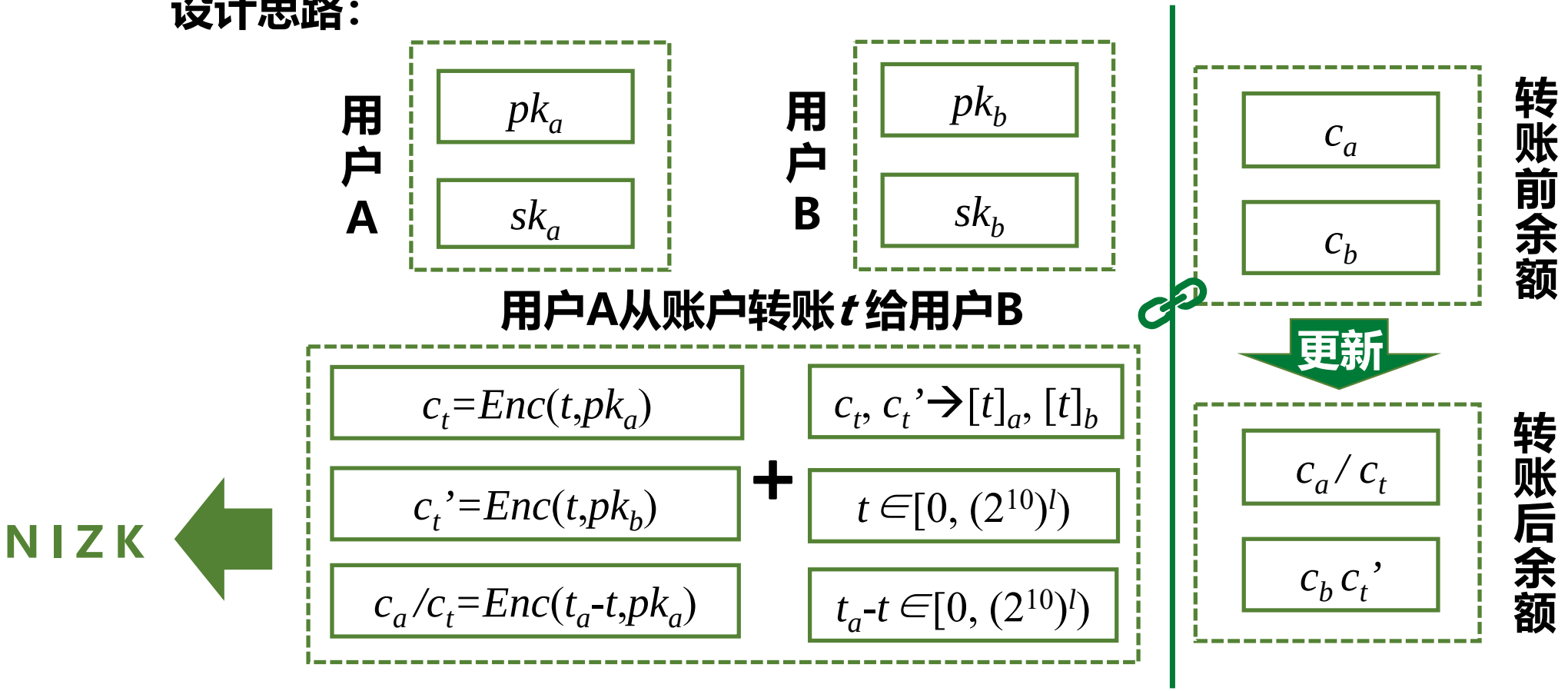

**问题分析:** 1. 交易金额明文； 2. 可验证：账本均衡+余额够减

**如何解决:** 1. 加法同态加密； 2. 零知识证明

# 10.5 零知识证明在区块链中的应用

## 4. 基于同态加密的NIZK协议

**设计思路：**



谢谢！