目录

第一章　零知识证明

随着网络技术的进步，一个具有广泛意义的问题需要关注，即如何进行身份证明，以在相互陌生且互不信任的双（多）方之间建立基本的信任。该问题的复杂性在于证明者既要证明自己的身份，又不能出示表明自己身份的私有信息，否则证明者自己的身份很容易被冒用。因此，如何“安全地”证明一个事实或命题就成为密码学中最具有意义的问题之一，而零知识证明正可解决此类问题。

零知识证明的概念诞生于上世纪80年代，最早由Goldwasser等人提出[1]，被誉为20世纪最重要的概念之一，是连接密码学与计算复杂性理论的重要纽带，在基于计算复杂性的现代密码学中占有重要地位。[2]给出了任何利用密码学将交互证明转化为零知识证明的方法，从而证明了任意交互可证明命题都可以零知识地证明，而Shamir证明了IP=PSPACE[3]，由此可知零知识证明强大的能力。[4, 5] Goldwasser、Micali、Goldreich 等一批密码学家通过对包括零知识证明等的研究，建立了现代密码学的可证明安全理论。

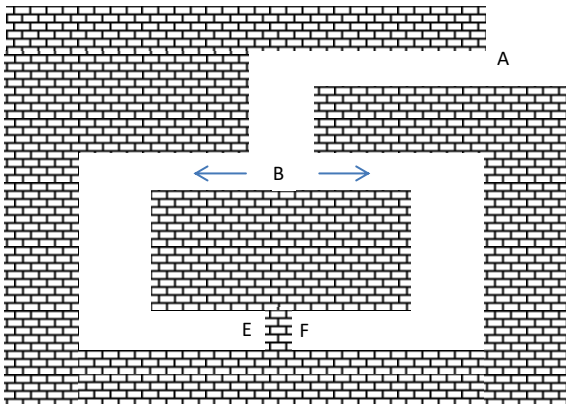
简单地说，零知识证明包含有两层含义，首先是对某个命题或声明的交互证明，其次要求证明是“零知识”。交互证明是证明者与验证者之间的协议，满足证明的两个基本要求：若命题正确，证明者一定可以使（诚实的）验证者接受这个命题；若命题不正确，证明者无论如何都不能诱骗验证者接受这个错误的命题。证明是零知识的，是要求除了命题本身的真假外，验证者不能获得任何其它额外的知识。因此，直观上看，零知识证明可同时实现了两个似乎完全矛盾的任务，这为需要兼顾认证与保密的密码学任务提供了强有力的工具。

在数学上，对命题的证明表现为长度有限的陈述或推理，将命题的正确性归约到无需证明的公理，验证者可以在有限时间内阅读完证明，从而可验证证明的正确性。显然，这种证明并不具有零知识性，因为拿到证明的验证者可以向另外的人证明该命题的正确性。与传统数学意义上的证明相比，零知识证明增加了两个元素：交互性与随机性，但也要保证验证者的验证一定可有效完成。交互性是允许证明时一个动态的交互过程，验证者根据需要提问（尽可能获得命题正确的证据），证明者回答提问（同时要确保没有知识泄露）。随机性是允许双方的行为（验证者的提问与证明者的回答）依赖于随机投币。交互性与随机性的结合，使得证明可以兼顾证明者与验证者双方的利益，验证者不必担心会被欺骗，证明者也相信只让验证者得到了命题的结论。在现实生活中，法庭上原被告之间的控辩过程就是一个交互证明的过程，双方既是证明者，也是验证者，都要证明自己的主张合法。

零知识证明可以用一个关于阿里巴巴与洞穴的经典故事来解释，这个洞穴如下图，它有一个入口A，进入洞穴后在B处分叉，各自通到E与F处就不通了。

故事是说在很久以前，有个人叫阿里巴巴。某日，阿里巴巴看到盗贼从A口进入了洞穴，于

是他就尾随而进，但到了B分叉处，他不知道盗贼从哪一个方向进去，只好随机猜测一个方向走进去，但结果是到了E（或F）处，发现此路不通，而且盗贼也不见了。对此，阿里巴巴想一定是自己选错方向了，盗贼走的是另一条路，就此作罢。一次，两次，…，当这样的事情一再发生，起初阿里巴巴认为自己运气不好，但当重复了很多次后，阿里巴巴心中有了疑问，为什么自己每次都会选错方向呢？是不是里面有什么机关呢？理论上，一次遇不到盗贼的概率是1/2，是大概率事件，但n次都遇不到盗贼的概率，随着n增大将是一个发生概率可忽略的事件。盗贼多次成功逃脱这个小概率事件的发生，实际上是（盗贼）向阿里巴巴证明了这样一个事实：这个洞穴一定存在隐蔽的出口供盗贼逃脱，但阿里巴巴除了相信这样一个事实外，没有得到更多的信息。故事的后来是阿里巴巴为了探求原因，他就事先把自己隐藏在E处，等到盗贼再一次来到这里时，它听到盗贼打开E与F之间的暗道的密语，终于知道了盗贼逃脱的原因。



1.1　交互证明

1.1.1　定义

交互证明（interactive proof）是一个两方协议，协议中的一方（称为证明者）向另一方（称为验证者）证明某个命题或论断成立，其中的命题（或论断）表示为x属于某个语言类L，即$x \in L$。交互证明的证明者P与验证者V是交互的多带Turing机，它们之间有通信带，一方的通信输出带是另一方的通信输入带，共同输入为欲证明的命题x，协议记为$\langle P, V\rangle(x)$。除了共同输入x 外，双方都会有各自的随机输入，要求双方的交互计算在多项式时间内（关于命题长度$|x|$的多项式）终止。协议运行终止后，输出验证者的输出，若验证者V 接受证明者的证明，命题正确输出1；若验证者拒绝证明者的证明，命题错误，输出0，记为$\langle P, V\rangle(x) = 1/0$。

定义 1.1.1　（交互证明）交互系统$\langle P, V\rangle$称为语言L的成员识别问题的交互证明系统，若V是多项式时间的，且满足如下两个条件。

1) 完备性*(Completeness)*：对任意$x \in L$，有

$$\Pr[\langle P, V\rangle(x) = 1] \geq \tfrac{2}{3}。$$

2) 合理性*(Soundness)*：对任意$x \notin L$以及任意交互图灵机B，有

$$\Pr[\langle B, V\rangle(x) = 1] < \tfrac{1}{3}。$$

其中的概率是关于*P*与*V*的随机输入（投币）。

在交互证明中，一般以命题的大小$|x|$作为协议的安全参数。与传统证明相比，交互证明允许错误发生，如在上面的定义中，允许以不超过1/3的概率出错，这大大提高了交互证明的能力。事实上，若L 具有确定性的非交互证明，则$L \in NP$。

进一步，上面语言L是否存在交互证明与完备性及可靠性的错误概率1/3有关，但实际上在一个较为广泛意义下却是无关的，我们有更一般的定义和定理。

定义 1.1.2 （交互证明）设函数$c, s : N \to [0,1]$满足$c(n) < 1 - 2^{-poly(n)}$、$s(n) > 2^{-poly(n)}$，且$c(n) > s(n) + \frac{1}{p(n)}$，其中$poly(\cdot)$为（不确定）任意多项式。交互协议$\langle P, V\rangle$ 称为语言L 的成员识别问题的交互证明系统，若V 是多项式时间的，且满足如下两个条件。

1) 完备性*(Completeness)*：对任意$x \in L$，有

$$\Pr[\langle P, V\rangle(x) = 1] \geq c(|x|)$$

2) 合理性*(Soundness)*：对任意$x \notin L$以及任意交互图灵机B，有

$$\Pr[\langle B, V\rangle(x) = 1] < s(|x|)$$

其中的概率是关于*P*与*V*的随机输入（投币）。通常称$e(|x|) = max\{s(|x|), 1 - c(|x|)\}$为交互证明错误概率。

定理 1.1.1 若语言*L*存在满足定义*3.7.1*的交互证明系统当且仅当*L*存在满足定义*3.7.2*的交互证明系统。

交互证明比传统的证明具有更强的能力，也就是可以证明传统证明不能证明的命题。如，向色盲者证明两个大小重量完全相同的两个球具有不同的颜色。色盲者（计算能力有限）不能区分颜色，这在传统意义下是无法证明。但是，采用交互式证明，色盲者就可以确信自己未被欺骗：色盲者两手各持一个球让证明者察看，然后放双手在身后，脑子里随机选0或者1，若选择0，将左右手的球互换；否则保证不变。结束后在出示给证明者，令其说出自己刚才脑子里想的是什么。重复多次，如果色盲者每次都能得到正确的答案，则他可以相信颜色确实不同（以极大的概率，与次数有关）。

定义 1.1.3 所有具有交互证明系统的语言组成的语言类记为IP。

显然，根据复杂性分类，我们有$NP \subseteq IP$与$BPP \subseteq IP$。Shamir在1992 证明了IP=PSAPCE，这也说明了交互证明远比传统的数学证明具有更强大的能力。下面是两个交互证明的例子。

1.1.2　图非同构问题

设$G_i=(E_i,V_i),i=1,2$是两个图，其中V_i是顶点集合，E_i 是边集合。称G_1,G_2同构，若$V_1=V_2$，且存在顶点集合上的置换π，使得$G_1=\pi(G_2)$，记为$G_1\cong G_2$。令

$$GNI=\{(G_1,G_2):G_1\text{与}G_2\text{不同构}\}$$

定理 1.1.2 $GNI\in IP$。

证明：对任意$x=(G_1,G_2)$，交互证明协议$\langle P,V\rangle(G_1,G_2)$如下：

协议 1.1.1

- 验证者V首先随机选择顶点集合上的置换π及$\sigma\in_R\{1,2\}$，令$G=\pi(G_\sigma)$，然后将G发送给证明者P。
- 证明者选择σ'，使得$G_{\sigma'}$同构于G，发送σ'给验证者。
- 验证者比较σ与σ，若$\sigma=\sigma'$，接受证明（输出1）；否则，拒绝证明（输出0）。

则协议1.1.1是GNI的交互证明系统。

完备性：当$x=(G_1,G_2)\in GNI$时，则G只与G_σ同构，所以证明者选择的σ'一定满足$\sigma'=\sigma$，因此，验证者一定接受证明，即有$\Pr[\langle P,V\rangle(G_1,G_2)=1]=1$。

可靠性：若$x=(G_1,G_2)\notin GNI$，即$G_1\cong G_2$，则G与G_1,G_2都同构，而σ是验证者随机选取的，因此无论证明者如何选择，均有$\Pr[\langle P,V\rangle(G_1,G_2)=1]\le\frac{1}{2}$。

在图非同构的交互证明中，可以$n=|V_1|=|V_2|$作为安全参数，为使协议具有可忽略的错误概率，可将上述协议运行n次，只有当所有证明都被接受时，验证者最终才接受，此时验证者被欺骗的概率就为2^{-n}。

1.1.3　非二次剩余问题

设$Z_n^*=\{x:x<n,gcd(x,n)=1\}$，记

$$QR=\{(x,n):x<n,\exists y,\text{使得 }y^2=x\ mod\ n\}$$

$$QNR=\{(x,n):x<n,\forall y,\text{ 都有 }y^2\neq x\ mod\ n\}$$

定理 1.1.3 $QNR\in IP$。

证明：对任意给定(x,n)，设$k=|n|$（n的二进制表示的长度），构造QNR的交互证明协议$\langle P,V\rangle(x,n)$如下：

协议 1.1.2

- 验证者V随机选择$b_i \in_R \{0,1\}, z_i \in_R Z_n^*$，并计算$w_i = x^{b_i} \cdot z_i^2 \ mod\ n$，$i = 1, \cdot, k$。然后，将$(w_1, \cdots, w_k)$发送给证明者$P$。
- 证明者收到$(w_1, \cdots, w_k)$后，判定w_i是否为二次剩余。若$w_i \in QR$，令$a_i = 0$，否则$a_i = 1$，$i = 1, \cdots, k$。最后将$a = (a_1, \cdots, a_k)$发送给验证者。
- 验证者比较a与$b = (b_1, \cdots, b_k)$，若$a = b$，接受证明（输出1）；否则，拒绝证明（输出0）。

则协议1.1.2是QNR的交互证明系统。

完备性：当$(x, n) \in QNR$时，则$w_i \in QNR$当且仅当$b_i = 0$，因此，证明者获得的a_i一定满足$a_i = b_i$，从而可使验证者接受证明，即有$\Pr[\langle P, V\rangle(x) = 1] = 1$。

可靠性：若$(x, n) \notin QNR$，则对任意$b_i \in \{0,1\}$，$w_i \in QR$，因此，对验证者随机选取的b_i，均有$\Pr[\langle P, V\rangle(x) = 1] = \Pr[a_1 = b_1, \cdots, a_k = b_k] = 2^{-k}$。

1.2 零知识证明

知识与通常所说的信息相关，但不等同于信息。信息代表了事件的发生，而知识从计算意义上讲与计算能力相关，拥有知识意味着具有对应的计算能力。如，甲投币后把结果告诉乙，对乙来说获得了信息（投币这个事件的结果），但未能获得知识（投币结果对它无任何帮助，即不会提升计算能力）。再如，甲给了乙一个Hamilton图，这样乙就获得这个图的所有信息，若甲在告诉其中的Hamilton圈，则对乙来说并未由此获得任何信息（图本身包含该信息），但它获得了知识（因为寻找其中的Hamilton圈在计算上不可能）。

交互证明称为是零知识的，就是说验证者从交互证明过程得不到任何知识（零知识），也就是交互证明过程不能改变验证者的计算能力。由于验证者为获得知识会采取恶意的交互策略，因此零知识实际上要求无论验证者如何选择交互策略，都不能获得足以提高计算能力的知识。粗略地，交互证明的零知识性可以表示为“对任意PPT的验证者，它通过与证明者交互能完成的任何任务都可以独自完成”。Goldwasser等人将思想形式化为“模拟范式”(simulation paradigm)，并由此定义了交互证明的零知识性。

1.2.1 定义

直观上，验证者V^*通过交互所能完成的任务可以用V^*的输出$\langle P, V^*\rangle(x)$来表示，而交互是一个随机计算的过程，因此$\langle P, V^*\rangle(x)$是依赖于双方的随机投币（或均匀分布的随机输入)的随机分布，而V独自可完成的任务就是用PPT算法可实现的任务。于是，零知识性可表示为：对任意PPT的V^*，存在PPT算法$\mathcal{S}$（用于模拟了实际交互过程，故称为模拟器或模拟算法），使得$\langle P, V^*\rangle(x)$与$\mathcal{S}(x)$具有相同的分布。但是，由于交互证明本身也允许错误发生，简单地要求$\langle P, V^*\rangle(x)$与$\mathcal{S}(x)$分布相同就过于苛刻，因此可适当放宽对$\mathcal{S}$的要求，即允许模拟算法M失败输出$\perp$，只要求$\mathcal{S}$在不失败的条件下，其输出与$\langle P, V^*\rangle(x)$同分布。

再回到阿里巴巴的故事，以解释定义零知识性的“模拟范式”。

很多很多年以后，阿里巴巴的后代从他留下的记录中发现这个洞穴的秘密，于是就向人们炫耀，说自己知道洞穴在E与F之间存在秘密通道密语。人们对这个洞穴都很熟悉，自然认为他说的不是真的。为了向人们证明自己没有说谎，但又不愿意告诉打开通道的密语，他想出了一个办法：他让人们在入口A等着，自己先走进去，然后让人们来到B处，并在左右两个方向中随机指定一个，然后他就会从指定方向走出来。正如阿里巴巴一样，反复多次后，人们终于相信他说的是真的。某电视台知道此事后，就找到阿里巴巴的后代商谈做一个节目，通过电视向更多的人展示这个洞穴的奇妙，但最终因阿里巴巴的后代要价太高终止。后来，有个聪明人想到一个办法，让电视台在没有阿里巴巴的后代的情况下完成这个电视节目的录制。

电视台找来一个酷似阿里巴巴的后代的人，摄像机放置在入口A处拍摄他从入口A走进洞穴。此人来到B处后随机投硬币，若是正面则从左边进去，否则就从右边进去。当他进去后，摄像机来到B处，同样也随机投硬币，若是正面则要求他从左边出来，否则就让他从右边出来。显然，只有当两次独立投币的结果相同时（概率为1/2），他才能从指定的方向走出来。于是，当摄像机在B处随机投硬币与他不一样时，就把刚才录的删掉，再重新投币，直到与他当初走进去时的投币一样时，再让他从投币指定方向走出来。这样，在电视节目上看着他走进洞穴，然后投币指定方向，他总可以从指定的方向走出来，这与阿里巴巴的后代向人们证明时完全一样。

定义 1.2.1 设$\langle P,V\rangle$是L的交互证明系统，如果对任意*PPT*的验证者V^*，存在一个*PPT*算法$\mathcal{S}^*$，使得对任意$x\in L$，满足：

1. $Pr[\mathcal{S}^*(x)=\perp]\leq\frac{1}{2}$。
2. 对任意$x\in L$，令$\mathcal{S}^*(x)$为条件分布$\mathcal{S}^*(x)=\alpha|\mathcal{S}^*(x)\neq\perp$，即

$$Pr[\mathcal{S}^*(x)=\alpha]=Pr[\mathcal{S}^*(x)=\alpha|\mathcal{S}^*(x)\neq\perp],\forall\alpha\in\{0.1\}^*$$

则$\{\mathcal{S}^*(x)\}$与$\{\langle P,V^*\rangle(x)\}$具有相同的分布，记为$\{\langle P,V^*\rangle(x)\}_{x\in L}\equiv\{\mathcal{S}^*(x)\}_{x\in L}$。

称$\langle P,V\rangle$是L完美零知识证明（*Perfect zero knowledge*），$\mathcal{S}^*$为完美模拟器。

注：定义中的条件$Pr[\mathcal{S}^*(x)=\perp]\leq\frac{1}{2}$可加强为$Pr[\mathcal{S}^*(x)=\perp]$可忽略。

一般的交互证明并不具有完美零知识性，如GNI与QNR的交互证明系统，但有些计算复杂类就具有简单的完美零知识证明系统，如复杂类BPP所包含的语言，事实上，一些人们相信不属于P的语言类也存在完美零知识证明系统。

GI的零知识证明系统。图同构语言$GI=\{(G_1,G_2):G_1\cong G_2\}$有完美零知识证明系统。$GI\in NP$，简单的证明是将同构映射交给验证者，但这样的证明当$GI\notin P$时将不是零知识证明。GI问题与阿里巴巴洞穴故事类似，G_1,G_2代表两个出口，二者之间的同构映射就是秘密通道。因此，可类似得到GI的零知识证明。

定理 1.2.1 GI存在完美零知识证明系统。

证明：对任意给定(G_1, G_2)为一对图，构造证明$(G_1, G_2) \in GI$的交互证明协议$\langle P, \rangle(G_1, G_2)$如下：

协议 1.2.1

1. 证明者P随机选择顶点集合$V_1 = V_2$上的置换π及$\tau \in \{1, 2\}$，并将$G = \pi(G_\tau)$发送给验证者V。
2. 验证者V随机选择$\sigma \in \{1, 2\}$，并将σ提交给P。
3. 若$\sigma \notin \{1, 2\}$，将$\perp$返回给验证者V；否则，令$\psi = \begin{cases} \pi, & \sigma = \tau \\ \pi \circ \varphi, & \sigma \neq \tau \end{cases}$，其中$\varphi$为$G_\sigma$到$G_\tau$的同构映射。最后，并将$\psi$返回给验证者$V$。
4. 验证者V验证$G = \psi(G_\sigma)$。若验证通过，接受证明，否则拒绝。

则协议1.2.1是GI的零知识证明。

完备性：当$(G_1, G_2) \in GI$时，则G既同构于G_1，也同构于G_2，因此对任意$\sigma \in \{1, 2\}$，证明者可求得ψ，使得$G = \psi(G_\sigma)$，从而有$\Pr[\langle P, V\rangle(G_1, G_2) = 1] = 1$。

可靠性：若$(G_1, G_2) \notin GI$，则无论证明者如何选择G，G一定不同构于G_1与G_2中的某一个，设为G_δ。这样，当$\sigma = \delta$，证明者不可能找到满足要求的同构映射ψ，从而验证者最终必然拒绝接受证明。因此，有$\Pr[\langle P, V\rangle(x) = 1] \leq \frac{1}{2}$。

由此证明了协议是GI的交互证明系统，下面证明零知识性。

零知识性：根据定义，需要构造模拟算法M满足条件。记$x = (G_1, G_2)$。模拟器$M^*(x)$以验证者交互策略V^*为子程序，其策略如下：

- 为V^*选择均匀分布的随机输入：设$p(\cdot)$为V^*的多项式时间上界，$M^*(x)$随机选择$r \in_R \{0, 1\}^{p(|x|)}$，将$r$写在$V^*$的随机输入带上。
- 模拟证明者的第一个消息：随机选择$\tau \in_R \{1, 2\}$ 以及置换ψ，计算$H = \psi(G_\tau)$。
- 模拟验证者消息：将H写入V^*的通信输入带上，运行V^*，最后读取V^*的通信输出带，获σ。
- 模拟证明者消息：若$\sigma \notin \{1, 2\}$，将$\perp$写在验证者V通信输入带上；否则，当$\tau = \sigma$时，将ψ写通信输入带上，而当$\tau \neq \sigma$时，模拟失败。
- 若模拟失败，输出$\perp$；否则，输出V^*的输出。

由于τ随机选取，因此对任意$x \in GI$，有

$$\Pr[M^*(x) = \perp] = \Pr[(\sigma \neq \tau) \vee (\sigma \notin \{1, 2\})] \leq \frac{1}{2}$$

当$\sigma \notin \{1, 2\}$时，M^*与V^*的（模拟）交互与真实交互无任何区别，因此其输出与V^*的输出也无区别。

当$\tau = \sigma$时，M^*与V^*的（模拟）交互与真实交互时的$\tau = \sigma$时的情形完全一样，因此其输出与V^*的输出也无区别。

M^*与V^*的交互与真实交互唯一不同是当$\sigma \neq \tau$时，M^*失败终止。但是，在$\sigma \neq \tau$情况下，证明者实际上仍然是发送一个顶点集上的随机置换（将G 解释为$\pi \circ \varphi(G_\sigma)$），因此，交互产生的随机分布与$\sigma = \tau$时产生的分布完全相同，因此$V^*$的在两种情况下的输出也完全相同，从而也与$M^*$ 的输出相同。交互证明的零知识性本质上是要求：PPT的验证者V^*通过运行模拟器，可以得到从交互过程得到任何信息（知识），完美零知识性要求$\{m^*(x)\}$与$\{\langle P, V^*\rangle(x)\}$具有相同的分布显然满足，但由于假设验证者$V^*$是PPT 时间的算法，因此可以适当放宽条件，即只要求$\{m^*(x)\}$与$\{\langle P, V^*\rangle(x)\}$ 是计算或统计不可区分，此时得到的称为（计算）零知识（computational zero knowledge）或统计零知识。

定义 1.2.2 设$\langle P, V\rangle$是L的交互证明系统，如果对任意*PPT*的验证者V^*，存在一个*PPT*算法M^*，使得对任意$x \in L$，满足：

1. $Pr[M^*(x) = \perp] \leq \frac{1}{2}$。
2. 对任意$x \in L$，令$m^*(x)$为条件分布$M^*(x) = \alpha | M^*(x) \neq \perp$，即

$$Pr[m^*(x) = \alpha] = Pr[M^*(x) = \alpha | M^*(x) \neq \perp], \forall \alpha \in \{0.1\}^*$$

则$\{m^*(x)\}_x$与$\{\langle P, V^*\rangle(x)\}_x$计算不可区分或统计不可区分

称(P, V) 是L计算零知识证明（*computational zero knowledge*，简称为零知识）或统计零知识证明（*statistical zero knowledge*）。

由于$M^*(x)$的失败概率可以通过重复运行多次降低至可忽略，这时在计算（统计）不可区分意义下便可忽略M^*的失败概率，直接要求$\{M^*(x)\}_x$与$\{\langle P, V^*\rangle(x)\}_x$计算不可区分或统计不可区分。因此，定义1.2.2有如下可以等价的形式。以下也经常会用$\mathcal{S}$表示模拟器。

定义 1.2.3 设$\langle P, V\rangle$是L的交互证明系统，称$\langle P, V\rangle$是（统计）零知识证明，如果对任意*PPT* 的验证者V^*，存在一个称为模拟器（*Simulator*）的*PPT*算法$\mathcal{S}$，使得$\{\langle P, V^*\rangle(x)\}_{x\in L}$与$\{\mathcal{S}(x)\}_{x\in L}$计算（统计）不可区分。

恶意验证者V^*的输入完全由它的随机输入（投币）以及在交互过程中收到的所有消息所决定，用$View_V^P(x)$表示验证者的随机输入r以及收到的所有消息$\{m_1, \cdots, m_k)\}$，即$View_V^P(x) = \{(x; r_v; m_1, \cdots, m_k)\}$，于是$V^*$交互完成后的输出就可以记为$V^*(View_{V^*}^P(x))$。计算零知识性就是要求存在$\mathcal{S}$，使得$\{\mathcal{S}(x)\}_{x\in L}$与$\{V^*(View_{V^*}^P(x))\}_{x\in L}$计算不可区分。由于$V^*$也是一个PPT的算法，因此，为满足计算零知识性，只需存在$\mathcal{S}$，使得$\{\mathcal{S}(x)\}_{x\in L}$ 与$\{View_{V^*}^P(x)\}_{x\in L}$计算不可区分即可。由此得下面的等价定义。

定义 1.2.4 设$\langle P, V\rangle$是L的交互证明系统，如果对任意*PPT*的验证者V^*，存在一个称为模拟器的*PPT*算法$\mathcal{S}$，使得$\{View_{V^*}^P(x)\}_{x\in L}$与$\{\mathcal{S}(x)\}_{x\in L}$计算不可区分，则称$\langle P, V\rangle$是（计算）零知识证明。

– 若存在一个*PPT*算法$\mathcal{S}$，使得$\{View_{V^*}^P(x)\}_{x\in L}$与$\{\mathcal{S}(x)\}_{x\in L}$统计不可区分，则称$(P,V)$是$L$的统计零知识证明。

– 若存在一个*PPT*算法$\mathcal{S}$，使得$\{View_{V^*}^P(x)\}_{x\in L}$与$\{\mathcal{S}(x)\}_{x\in L}$同分布，称$(P,V)$是$L$的完美零知识证明。

通常，将随机分布总体$\{X_i\}$与$\{Y_i\}$计算不可区分记为$\{X_i\} \stackrel{c}{=} \{Y_i\}$，类似地，统计不可区分记为$\{X_i\} \stackrel{s}{=} \{Y_i\}$；而同分布记为$\{X_i\} \stackrel{d}{=} \{Y_i\}$。（为简便，有时也写为$X_i \stackrel{c}{=} Y_i$或$X_i \stackrel{s}{=} Y_i$）

以下若无特别说明，零知识均指计算零知识。

1.2.2 Diffie-Hellman问题的零知识证明

设有PPT算法$\mathcal{G}$，$(G,g) \leftarrow \mathcal{G}(1^n)$，其中$G$为$q$（素数）阶循环群，$|q| = n$，$g$为$G$的生成元。$(g, g^a, g^b, g^{ab})$称为Diffie-Hellman四元组，对应的判定性假设（DDH假设）为

$$\{(g, g^a, g^b, g^{ab})\}_{n\in\mathbb{N},a,b\in_R\mathbb{Z}_q} \stackrel{c}{=} \{(g, g^a, g^b, g^c)\}_{n\in\mathbb{N},a,b,c\in_R\mathbb{Z}_q}$$

与之相关的问题是：如何不泄露a,b，并证明(g, g^a, g^b, g^c)是Diffie-Hellman四元组，即证明$c = ab$。为方便，将Diffie-Hellman四元组记为$(g, h = g^a, y_1 = g^b, y_2 = h^b)$。问题可叙述为：给定四元组：$(g, h, y_1, y_2)$，证明存在$\alpha$，使得$y_1 = g^\alpha, y_2 = h^\alpha$。

记$DH = \{x = (g, h, y_1, y_2) : (G,g) \leftarrow \mathcal{G}(1^n); h = g^a, y_1 = g^b, y_2 = h^b\}$，下面给出$DH$的零知识证明系统如下：

协议 1.2.2

公共输入：$x = (g, h, y_1, y_2)$;
证明者拥有辅助输入：α，满足$y_1 = g^\alpha, y_2 = h^\alpha$。

– P随机选择$u \in \mathbf{Z}_q^*$，计算$A = g^u, B = h^u$，并将A, B发送给V。

– V随机选择$\sigma \in \{0,1\}$，并将σ提交给P。

– P计算$s = \sigma \cdot \alpha + u \ mod \ q$，并将$s$返回给$V$。

– V验证$g^s = Ay_1^\sigma, h^s = By_2^\sigma$。若验证通过，接受证明，否则拒绝。

则协议1.2.2 是DH的零知识证明。

完备性：当$x = (g, h, y_1, y_2) \in DH$时，由于对任意$\sigma \in \{0,1\}$，$g^s = g^{\sigma\alpha+u} = Ay_1^\sigma, h^s = h^{\sigma\alpha+u} = By_2^\sigma$，因此，$\Pr[\langle P, V\rangle(x) = 1] = 1$。

可靠性：若$x = (g, h, y_1, y_2) \notin DH$，即$y_1 = g^{\alpha_1}, y_2 = h^{\alpha_2}$，$\alpha_1 \neq \alpha_2$，则无论证明者如何生成$A, B$及$s$，关系$g^s = Ay_1^\sigma, h^s = By_2^\sigma$不可能对$\sigma = 0, 1$都成立。事实上，若$A = g^{u_1}, B = h^{u_2}$且$u_1 \neq u_2$，则对$\sigma = 0$，无论证明者如何选择$s$，$Ay_1^\sigma = A \neq g^s$与$By_2^\sigma = B \neq h^s$必有一个成立，因此验证者拒绝；若$A = g^u, B = h^u$，则对$\sigma = 1$时，$Ay_1^\sigma = g^{\alpha_1+u} \neq g^s, By_2^\sigma = h^{u+\alpha_2} \neq h^s$必有一个成立，因此验证者拒绝。总之，$\Pr[\langle P, V\rangle(x) = 1] \leq \frac{1}{2}$。

零知识性：模拟器$M^*(x)$如下：

1) 首先随机选择V^*的随机输入r，然后选择$\tau \in_R \{0,1\}$以及$z \in_R Z_q^*$。(1) 若$\tau = 0$，计算$A = g^z, B = h^z$。(2) 若$\tau = 1$，令$A = g^z/y_1, B = h^z/y_2$。

2) 以A, B提交给V^*（将A, B写入V^*的通信输入带上，运行V^*），并获得回复σ。

3) 若$\tau \neq \sigma$，输出$\perp$，失败终止；否则，输出$(x, r, (A, B), z)$并终止。

明显地，$\Pr[M^*(x) = \perp] = \Pr[\sigma \neq \tau] = \frac{1}{2}$，下面只需说明当$\tau = \sigma$时，$M^*(x)$的输出，与$View_{V^*}^P(x)$具有相同的分布。

依据协议，证明者的消息$(A = g^z, B = h^z)$在$G \times G$上均匀分布。当$\tau = 0$时，模拟器$m^*(x)$的输出$(x, r, (g^z, h^z), z)$与$View_{V^*}^P(x)$没有区别；当$\tau = 1$时，模拟器计算$A = g^z/y_1 = g^{z-\alpha}$，$B = h^z/y_2 = h^{z-\alpha}$，若$z$均匀分布，则$z' = z - \alpha$也是均匀分布，由此可得模拟器的输出

$$(x, r, (g^{z'}, h^{z'}), z = \alpha + z')$$

与$View_{V^*}^P(x)$依然没有区别。因此，当$\tau = \sigma$时，模拟器$m^*(x)$的输出$(x, r, (A, B), z)$与$View_{V^*}^P(x)$具有相同的分布。 ∎

1.2.3　黑盒模拟零知识

交互证明$\langle P, V\rangle$的零知识性要求对任意的V^*，存在PPT模拟器$\mathcal{S}$，满足$\{\mathcal{S}(x)\}_{x\in L}$与$\{View_{V^*}^P(x)\}_{x\in L}$不可区分，因此模拟器$\mathcal{S}$与验证者的策略$V^*$相关，为明确它们之间的相关性，可把模拟器记为$\mathcal{S}_{V^*}$。传统上，实际构造模拟器$\mathcal{S}_{V^*}$都以黑盒方式使用$V^*$，即模拟器仅以黑盒方式使用验证者的策略$V^*$，这种模拟方式称为黑盒模拟，所得到的零知识性也就称为黑盒零知识（通常就简称为零知识）。为了强调黑盒模拟的特殊性，黑盒零知识可采用下述定义。

定义 1.2.5 设$\langle P, V\rangle$是L的交互证明系统，如果存在一个称为模拟器的*PPT*算法$\mathcal{S}$，使得对任意*PPT*的验证者V^*，$\{View_{V^*}^P(x)\}_{x\in L}$与$\{\mathcal{S}^{V^*}(x)\}_{x\in L}$（统计）计算不可区分，则称$\langle P, V\rangle$是（统计）黑盒零知识证明。

由定义可以看出，模拟器的策略（算法）$\mathcal{S}$与验证者V^*无关，仅将V^*作为子程序（Oracle）调用，这使得模拟器的构造变得比较简单。例如在协议1.2.1与协议1.2.2的零知识性证明中，所构造的模拟器均为黑盒模拟器。实际上，黑盒模拟是最为普遍的模拟方式，以往给出的零知识证明大都黑盒零知识。

虽然黑盒模拟在零知识性上取到了极多的成果，但由于模拟方式简单也带来一些局限，如Goldreich等人[52]证明了BPP以外的复杂类不存在3轮的黑盒零知识证明和常数轮黑盒Arthur-Merlin证明，而Canetti等人[16]证明常数轮可并发黑盒零知识证明仅对BPP类存在。这些黑盒模拟意义下的否定性结论促使人们寻求更复杂的模拟方式，于是在2001年，Barak给出了一种不以黑盒方式使用验证者策略V^*的模拟方法，并由此得到了NP问题的常数轮可（有界）并发零知识论证系统，得到了黑盒模拟意义下不可能的结果。

定理 1.2.2 设$\langle P,V\rangle$是L的交互证明系统，若其错误概率可忽略，且满足下列条件：

– 常数轮的黑盒零知识；

– 公开投币证明（*Arthur-Merlin*证明，验证者只需发送自己的随机投币结果）。

则必有$L \in BPP$。

现在，人们将不是黑盒模拟统称为非黑盒模拟，得到的零知识性也就称为非黑盒零知识。在Barak之后，非黑盒模拟得到很多的关注，也得到许多重要的结果。

1.2.4 零知识证明的复合

在设计交互证明协议时，为使协议尽可能简单，常常允许其错误概率是常数，例如前面构造的关于GNI与GI的证明协议，可靠性错误概率均为1/2，这在实际中显然是不能接受的。为降低一个交互证明的错误概率，一般可采用串行或并行复合的方法，即将基本协议运行多次，验证者接受当且仅当所有的证明都接受，这样使可靠性错误概率随运行次数快速下降。例如，n次运行GNI证明协议，其错误概率为2^{-n}。

1.2.4.1 顺序复合

顺序复合交互证明可以降低可靠性错误，但对零知识证明来说，带来的问题就是复合协议还是否保持零知识性，人们自然希望顺序复合能保持基本协议零知识性，但不幸的是结论是否定的。因此，为使零知识性在顺序复合下具有封闭性，需要更强的零知识性定义。

直观上，协议顺序复合产生的问题是，后一次运行是在前一次运行的基础上进行的，也就是说，协议除了欲证命题作为公共输入外，验证者还可以得到前一次的运行结果。

设伪随机集合序列$S_1 \subseteq \{0,1\}^4, \cdots, S_n \subseteq \{0,1\}^{4n}, \cdots$ 满足对任意的PPT算法A及常数c，存在N，当$n > N$时，对任意的$x \in \{0,1\}^n$，有$Pr[A(x) \in S_n] < n^{-c}$（其存在性可以证明）。设$K(x)$是满足$L_K = \{x : K(x) = 1\} \notin BPP$的布尔函数。对$L = \{0,1\}^*$，考虑如下协议：

$P(x)$		$V(x)$
	$\xleftarrow{r}$	验证者随机选择$r \in_R \{0,1\}^{4n}$
证明者检测$r \in S_n$		
若$r \in S_n$	$\xrightarrow{K(x)}$	
否则	$\xrightarrow{s_0 \in_R S_n}$	验证者接受证明

单个协议运行时，除去一个可忽略的概率，验证者只得到了一个伪随机s_0，因此协议是零知识的（模拟器只需随机选择s_0输出即可）。但若两个协议顺序运行，验证者在第一个协议结束后，以极大的概率得到$s_0 \in S_n$，这样在第二个协议中用s_0就可得到$K(x)$。根据假设，任意PPT算法都不可能以不可忽略的概率由x得到$K(x)$，因此两个顺序运行的协议不再具有零知识性。

为使零知识性在顺序复合下得以保持，需要略微强化零知识性的定义。

定义 1.2.6 设交互图灵机P,V除共同输入外，各有辅助输入y与z，交互协议$\langle P(y),V(z)\rangle(x)$是$L$的交互证明系统，令

$$P_L(x)=\left\{y:\forall z\in\{0,1\}^*,\Pr[\langle P(y),v(z)\rangle(x)=1]>\frac{2}{3}\right\}$$

如果对任意*PPT*的验证者V^*，存在*PPT*（关于$|x|$）模拟器$\mathcal{S}$，使得对任意$y\in P_L(x)$，满足

$$\left\{View^{P(y)}_{V(z)^*}(x)\right\}_{x\in L,z\in\{0,1\}^*}\stackrel{\triangle}{=}\{S(x,z)\}_{x\in L,z\in\{0,1\}^*}$$

则称$(P(y),V(z))(x)$是L的带辅助输入的（计算）零知识证明系统（当$\triangle=c$）、或带辅助输入的统计零知识证明系统（当$\triangle=s$）、或带辅助输入的完美零知识证明（当$\triangle=d$）。 ∎

注：验证者与模拟器的运行时间均以$|x|$来度量。

定理 1.2.3 设交互协议$\langle P(y),V(z)\rangle$是L的带辅助输入的交互证明系统，对任意多项式$Q(|x|)$，令$\langle P_Q,V_Q\rangle(x)$为顺序运行$Q$次基本协议$\langle P(y),V(z)\rangle(x)$，其中$P_Q$是$Q$个独立证明者$P$，而$V_Q$充当验证者完成$Q$次基本协议。若基本协议$\langle P(y),V(z)\rangle(x)$是带辅助输入的（统计、完美）零知识证明，则$\langle P_Q,V_Q\rangle(x)$也是带辅助输入的（统计、完美）零知识证明。

证明： 假设第i个基本协议记为$\langle P_i,V_i\rangle$。复合证明者P_Q在每个协议都是一个独立的基本协议的证明者，为方便，在第i协议中的证明者记为$P_i(x,y)$。对一个不诚实的复合验证者V_Q^*来说，$V_1^*(x,z),\cdots,V_Q^*(x,z)$可能并不独立，尤其在关注零知识性时，总需假设$V_i^*$依赖于$V_{i-1}^*$的运行结果。因此，第$i$个协议中的$V_i^*(x,z)$可以用$V^{**}(x,i,z_{i-1})$来表示，其中$z_{i-1}$是$V_{i-1}^*$的输出，$i=1,\cdots,Q$，最后$V^{**}(x,Q,z_{Q-1})$的输出记为$V_Q^*$的输出，这里$z_0=z$。

$$\begin{array}{rcl}\underline{P_Q(x,y)} & \rightleftharpoons & \underline{V_Q^*(x,z)}\\ P_1(x,y) & \rightleftharpoons & V_1^*\\ & \vdots & \\ P_i(x,y) & \rightleftharpoons & V_i^*\\ & \vdots & \\ P_Q(x,y) & \rightleftharpoons & V_Q^*\to z_Q\end{array}\quad\Longrightarrow\quad\begin{array}{rcl}\underline{P_Q(x,y)} & \rightleftharpoons & \underline{V^{**}(x,z)}\\ P_1(x,y) & \rightleftharpoons & V^{**}(x,z_0)\to z_1\\ & \vdots & \\ P_i(x,y) & \rightleftharpoons & V^{**}(x,z_{i-1})\to z_i\\ & \vdots & \\ P_Q(x,y) & \rightleftharpoons & V^{**}(x,z_{Q-1})\to z_Q\end{array}$$

这里第Q个协议里的验证者仍用V_Q^*表示，其输出即为复合协议的输出。由于$\langle P(y),V(z)\rangle(x)$是带辅助输入的零知识证明，因此，存在模拟器$\mathcal{S}$，对任意$i=1,\cdots,Q$，使得$\{\mathcal{S}(x,z_{i-1})\}_{x,z_{i-1}}$与$\{\langle P_i(y),V^{**}(z_{i-1})\rangle(x)\}_{x,z_{i-1}}=\{z_i\}_{x,z_{i-1}}$计算不可区分。当$i=Q$时，便有

$$\{\mathcal{S}(x,z_{Q-1})\}_{x,z_{Q-1}}\stackrel{c}{=}\{\langle P_i(y),V^{**}(z_{Q-1})\rangle(x)\}_{x,z_{Q-1}}=\{z_Q\}_{x,z_{Q-1}}$$

为简明，以下略输出分布中的下标x, z_{i-1}。尽管$\mathcal{S}(x, z_{Q-1})$的输出与z_Q不可区分，但由于$\mathcal{S}(x, z_{Q-1})$利用z_{Q-1}（真实交互产生的中间输出）作为辅助输入，$\mathcal{S}(x, z_{Q-1})$并不能作为复合协议的模拟器。令

$$z_i' \leftarrow \mathcal{S}(x, z_{i-1}), i = 1, \cdots, Q, z_0' = z$$

由$\{z_i\}$与$\{z_i'\}$不可区分，而$\mathcal{S}$是PPT算法，从而根据计算不可区分的性质可知，$\{z_{i+1}' \leftarrow \mathcal{S}(x, z_i)\}$与$\{\mathcal{S}(x, z_i')\}$也不可区分。即

$$\{z_i\} \stackrel{c}{=} \{z_i'\} \Rightarrow \{z_{i+1}' \leftarrow \mathcal{S}(x, z_i)\} \stackrel{c}{=} \{\mathcal{S}(x, z_i')\},\ i = 0, \cdots, Q-1$$

于是就有

$$\{\mathcal{S}(x, z_{Q-1})\} \stackrel{c}{=} \{\mathcal{S}(x, z_{Q-1}')\}$$

由此可以看出，复合协议的模拟器可由$\mathcal{S}$递归定义：令

- 令$z_0' = z$，$z_i' \leftarrow \mathcal{S}(x, z_{i-1}')$，$i = 1, \cdots, Q$；
- 输出z_Q'。

为证明z_Q'与z_Q的不可区分性，定义混合分布

$$H^{(i)}(x, z_0) = \mathcal{S}_{Q-i}(x, z_i), i = 0, 1, \cdots, Q$$

$$\mathcal{S}_0(x, z) = z, \mathcal{S}_k(x, z) = \mathcal{S}_{k-1}(x, \mathcal{S}(x, z))$$

则容易验证：$H^{(0)} = \mathcal{S}_Q(x, z_0)$，$H^{(Q)} = \mathcal{S}_0(x, z_Q) = z_Q$。若$\mathcal{S}_Q(x, z_0)$与$z_Q$可区分，即$H^{(0)}$与$H^{(Q)}$可区分，则必存在$1 \le i \le Q$，使得$H^{(i-1)}$与$H^{(i)}$是可区分的。

$H^{(i-1)}(x, z_0)$

$V^{**}(x, z_0)$	$\rightarrow$	z_1
	$\vdots$	
$V^{**}(x, z_{i-2})$	$\rightarrow$	z_{i-1}
$\mathcal{S}(x, z_{i-1})$	$\rightarrow$	$\underline{z_i'}$
$\mathcal{S}(x, z_i')$	$\rightarrow$	z_{i+1}'
	$\vdots$	
$\mathcal{S}(x, z_{Q-1}')$	$\rightarrow$	z_Q'

$H^{(i)}(x, z_0)$

$V^{**}(x, z_0)$	$\rightarrow$	z_1
	$\vdots$	
$V^{**}(x, z_{i-2})$	$\rightarrow$	z_{i-1}
$V^{**}(x, z_{i-1})$	$\rightarrow$	$\underline{z_i}$
$\mathcal{S}(x, z_i)$	$\rightarrow$	z_{i+1}'
	$\vdots$	
$\mathcal{S}(x, z_{Q-1}')$	$\rightarrow$	z_Q'

这里需要注意到$\mathcal{S}(x, z_i')$与$\mathcal{S}(x, z_i)$输出均记为z_{i+1}'，这是因为z_i'与z_i是不可区分的，从而$\mathcal{S}(x, z_i')$与$\mathcal{S}(x, z_i)$也是不可区分的，因此可不加区别。

若存在算法D区分z_Q与$z_Q\prime$，即存在z及多项式p，使

$$|Pr[D(x, z, \langle P_Q, V^*(z)\rangle(x)) = 1] - Pr[D(x, z, \mathcal{S}_Q(x, z)) = 1]| > \frac{1}{p(|x|)}$$

则必有

$$|Pr[D(x, z, H^{(i)}(x, z)) = 1] - Pr[D(x, z, H^{(i-1)}(x, z)) = 1]| > \frac{1}{Q(|x|)p(|x|)}$$

即

$$|Pr[D(x,z,\mathcal{S}_{Q-i}(x,\langle P,V^{**}(z_{i-1})\rangle(x)))=1]-Pr[D(x,z,\mathcal{S}_{Q-i}(x,\mathcal{S}(x,z_{i-1})))=1]|>\frac{1}{Q(|x|)p(|x|)}$$

于是，由D与$\mathcal{S}$可构造算法$D\prime$(需要有z及i)，区分$(x,z_{i-1},\langle P,V^{**}(z_{i-1})\rangle(x))$与$(x,z_{i-1},M^{**}(x,z_{i-1}))$。这与假设$\langle P,V\rangle$是带辅助输入零知识相矛盾。命题得证。 ∎

注：前面给出的GI问题与DH问题的零知识证明都是带辅助输入的零知识证明，所以可以利用顺序复合降低错误概率。

1.2.4.2　并行复合

顺序运行可以降低错误概率，但带来的问题是协议的交互次数随运行次数而增加，也就是轮数不再是常数。并行是协议复合的另一种形式，它可以在降低错误概率的同时保持协议交互次数（称为轮数）不变，但它带来的问题更严重，因为（带辅助输入）零知识性在并行复合下的封闭性无法证明，而且人们也找不到一个可在并行下保持不变的零知识的一般定义。

在顺序复合时，在带辅助输入的条件下，我们可以用基本协议的模拟器构造复合协议的模拟器，但在并行环境下，由基本协议的模拟器无法构造并行复合协议的模拟器。下面以GI 问题的协议1.2.1为例进行说明。

假设证明者P与验证者V并行地运行协议1.2.1的k个拷贝，可简要描述如下：

$$\begin{array}{ccc} P(G_1,G_2) & & V(G_1,G_2) \\ & \xrightarrow{\pi_1(G_{\tau_1})}\ \cdots\ \xrightarrow{\pi(G_{\tau_i})}\ \cdots\ \xrightarrow{\pi_k(G_{\tau_k})} & \\ & \xleftarrow{\sigma_1}\ \cdots\ \xleftarrow{\sigma_i}\ \cdots\ \xleftarrow{\sigma_k} & \\ & \xrightarrow{\psi_1}\ \cdots\ \xrightarrow{\psi_i}\ \cdots\ \xrightarrow{\psi_k} & \end{array}$$

其中$(\pi_i(G_{\tau_i}),\sigma_i,\psi_i)$对应于第$i$个拷贝的交互过程。若通过$k$次调用基本协议的模拟器$\mathcal{S}$构成并行协议的模拟器，就需要独立地正确猜测每一个$\sigma_i$，其成功概率仅为$2^{-k}$。若希望模拟器的成功概率大于1/2，平均来说，模拟器需要运行多于2^{k-1}次，这说明要使模拟器在多项式时间内完成，则k应该满足$k=O(logn)$（这里n是安全参数），即保证2^{-k}不可忽略，而这也正是P欺骗成功的概率。因此，$k=O(logn)$次并行与单个协议的运行并无本质区别，并不能满将错误概率降低至可忽略的目标。

上面的例子只标明该并行协议的零知识性不能证明，并不意味着它一定不是零知识的。一般来说，协议的是否可并行（保持零知识性）需要具体分析，这既与协议有关，也与模拟器的构造方式有关，Goldreich等人在[52]证明了若采用黑盒模拟，非平凡问题不会有3轮零知识证明和常数轮Arthur-Merlin证明。

1.2.4.3　并发复合

协议的并发运行远比并行时的情况复杂，并行知识并发的特例，因此并发运行自然也不能保持零知识性。零知识证明的顺序与并行复合，主要是为降低错误概率，而协议的并发却是协议运行一般形态，因此协议的并发安全性（零知识性）极为重要。Feige[11]最早开始研究零知识

证明的并发安全性，此后，并发零知识的研究受到了极大的关注，人们对可并发的零知识证明的存在性，以及如何设计可并发的零知识协议进行了大量的研究。早期的研究首先给出了否定性结论，[52]证明只有BPP语言类才存在3轮的黑盒零知识证明和常数轮Arthur-Merlin证明，[12]给出了可并发零知识证明轮复杂性下界，证明黑盒模式下4轮可并发的零知识证明只有BPP才存在，而Rosen[15]将下界提高至7 轮。后来Canetti等人[16]证明非平凡语言类不存在常数轮可并发黑盒零知识证明。Richardson等人[13]首次给出NP问题的可并发零知识证明，其轮复杂性为$O(n^{\varepsilon})$，后来[18]给出了NP问题的轮复杂性为$O(\alpha(n)\log n)$的可并发零知识证明系统。

1.2.5 交互与随机性的作用

交互与随机性的引入不仅使证明具有了更强大的能力，IP=PSPACE，而且也使证明可以具有看似与“证明”矛盾的性质，零知识性，如果没有交互与随机性，零知识证明将只能是平凡的证明，即只能对BPP类进行证明。

定理 1.2.4 若L具有单向零知识证明，则$L \in BPP$。

证明： 设$\langle P, V\rangle$是L的单向证明，即证明者P发送证明π，验证者V验证π。若$\langle P, V\rangle$是零知识的，则根据定义，存在PPT的模拟算法$\mathcal{S}$，使得$\{\mathcal{S}(x)\}$与$\{x, r, \pi \leftarrow P(x)\}$计算不可区分（这里$r$为$V$的随机输入）。定义$L$的判定算法$D$如下：

> 对输入x，首先运行$\mathcal{S}(x)$，$(x, r, \pi') \leftarrow \mathcal{S}(x)$，其中$r \in \{0,1\}^{\ell}$随机均匀选择，然后运行$V(x, \pi'; r)$。若$V$接受，则输出1，否则输出0。

显然，D是PPT算法。

若$x \in L$，根据证明的完备性，$\Pr[V(x,\pi) = accept : \pi \leftarrow P(x)] > 2/3$，而$\{\mathcal{S}(x)\}$与$\{x, r, \pi \leftarrow P(x)\}$计算不可区分，因此，$\Pr[D(x) = 1] > 2/3 - negl(n)$；若$x \notin L$，根据证明的可靠性，必有$\Pr[D(x) = 1] < 1/3$，否则，证明者可利用$\mathcal{S}$欺骗$V$。由此即得$L \in BPP$。 ∎

定理 1.2.5 设$\langle P, V\rangle$是L的一个零知识证明系统，而且验证者V是确定性的，则$L \in BPP$。

证明： 由于V是确定性算法，P完全可以首先确定每一轮要发送的消息。因此，P可以先计算出所有的交互消息，然后发送给验证者，这就将交互证明转化为单向证明，而且同样也是一个零知识证明。根据定理1.2.4，必有$L \in BPP$。 ∎

定理 1.2.6 设$\langle P, V\rangle$是L的一个零知识证明系统，而且证明者P是确定性的，则$L \in BPP$。

证明： 假设协议$\langle P, V\rangle(x)$首先由验证者发送第一个消息，交互过程如下：

$$
\begin{array}{ccc}
P(x) & & V(x) \\
 & \xleftarrow{\alpha_1} & \alpha_1 \leftarrow V(x) \\
\beta_1 \leftarrow P(x,\alpha) & \xrightarrow{\beta_1} & \\
 & \cdots & \\
 & \xleftarrow{\alpha_k} & \alpha_k \leftarrow V(x, \beta_1, \cdots, \beta_{i-1}) \\
\beta_k \leftarrow P(x, \alpha_1, \cdots, \alpha_k) & \xrightarrow{\beta_k} & V(x, \beta_1, \cdots, \beta_k) = 0/1
\end{array}
$$

由于P是一个确定性算法，因此对任意的i，$\beta_i = P(x, \alpha_1, \cdots, \alpha_i)$就由$x, \alpha_1, \cdots, \alpha_i$唯一确定。若$\langle P, V\rangle(x)$是零知识的，则对任意的$V^*$，存在PPT的模拟算法$\mathcal{S}_{V^*}(x)$，使得$\mathcal{S}_{V^*}(x)$与$View^P_{V^*}(x)$不可区分，这意味着$\mathcal{S}_{V^*}(x)$输出中的消息$\beta'_i$，也由$x, \alpha_1, \cdots, \alpha_i$唯一确定。由此，定义$L$的判定算法$D$如下：

1) 对输入x，首先随机选取V^*的随机输入。

2) $D(x)$充当证明者P与V^*按照协议过程进行交互，从$i = 1$到k完成如下过程：

   - 以β'_{i-1}（β'_0为空）调用V^*并获得消息α_i;
   - 运行$\beta'_i = \mathcal{S}_{V^*}(x, \alpha_1, \cdots, \alpha_i)$，$(x, r, \pi') \leftarrow \mathcal{S}(x)$，其中$r \in \{0,1\}^\ell$ 随机均匀选择，

3) 最后，运行$V^*(x, \beta'_1, \cdots, \beta'_k; r)$。若$V$ 接受，则输出1，否则输出0。

当$x \in L$时，由$\mathcal{S}(x)$与$View^P_{V^*}(x)$不可区分可知

$$|\Pr[D(x) = 1] - \Pr[\langle P, V\rangle(x)] = 1| = |\Pr[V^*(x, \beta'_1, \cdots, \beta'_k) = 1] - \Pr[\langle P, V\rangle(x)] = 1|$$

可忽略，从而得$|\Pr[D(x) = 1] > 2/3 - negl(n)$；当$x \notin L$时，由$\langle P, V\rangle(x)$的完备性可得$|\Pr[D(x) = 1] \leq negl(n)$。因此，$L \in BPP$。 ∎

1.3　NP问题的零知识证明

零知识证明之所以在密码学中具有十分广泛的应用，是源于Goldreich等人证明的如下定理：

定理 1.3.1 若单向函数存在，则任意*NP*问题都存在零知识证明系统。

为证明这个定理，我们只需在单向函数存在的假设下，证明某个NPC问题存在零知识证明系统。为此，需要另外一种协议——承诺方案。

1.3.1　承诺方案

承诺方案是一个双方（承诺者$\mathcal{C}$，接收者$\mathcal{R}$）之间的协议，承诺者拥有私有输入x，协议可记为$\langle \mathcal{C}(x), \mathcal{R}\rangle$。承诺协议的运行分为两个阶段：承诺阶段与公开阶段。在承诺阶段，承诺者给出（可能有交互）对私有数据x（或消息）的承诺，以说明自己确实拥有该数据但又不泄露该证据；在公开阶段，承诺者给出该数据和承诺阶段的相关交互信息，接收者由此可确认公开的数据是否是此前承诺的数据x。承诺阶段需要具有隐藏性，接收者看到承诺后并不能获取被承诺数据的任何信息；公开阶段需要绑定性，承诺者不能把此前的承诺解释为两个不同的数据的承诺。承诺方案两个阶段安全性分别为：

(1) 隐藏性（hiding）：双方在承诺阶段完成对x的承诺后，接收者不能得到关于x的任何信息。若隐藏性对PPT的接收者成立，则称为计算隐藏性；若隐藏性对计算能力无限的接收者也成立，则称为统计隐藏性。

(2) 绑定性（binding）：在公开阶段，发送者不能使接收者接受一个不同于x的消息$x\prime$。若绑定性对PPT的承诺者成立，则称为计算绑定性；若绑定性对计算能力无限的承诺者也成立，则称为统计绑定性。

定义 1.3.1 承诺方案$\langle \mathcal{C}, \mathcal{R} \rangle$由承诺与公开两个阶段组成，记为

$$\langle \mathcal{C}, \mathcal{R} \rangle = (\langle \mathcal{C}(commit, x), \mathcal{R} \rangle, \langle \mathcal{C}(decom, x), \mathcal{R} \rangle)$$

其中$commit$与$decom$分别表示承诺阶段与公开阶段。下面为简单，将$\mathcal{C}(commit, \cdot)$记为$\mathcal{C}_1(\cdot)$，$\mathcal{C}(decom, \cdot)$记为$\mathcal{C}_2(\cdot)$（若不引起误解，则可统一记为$\mathcal{C}$）。双方在承诺阶段的输出为$ok$（承诺正常完成）或$\perp$（承诺失败）。接受者在公开阶段输出$accept$或$reject$。

– 隐藏性: 承诺阶段运行$\langle \mathcal{C}_1(x), \mathcal{R} \rangle(1^n)$，接收者收到的所有消息记为$View_{\mathcal{R}^*}^{\mathcal{C}_1(x)}(1^n)$，其中$n$为安全参数。若对任意$x \neq x'$以及任意*PPT*的$\mathcal{R}^*$，下面两个分布

$$View_{\mathcal{R}^*}^{\mathcal{C}_1(x)}(1^n) \text{与} View_{\mathcal{R}^*}^{\mathcal{C}_1(x')}(1^n)$$

计算（统计）不可区分，称承诺是计算（统计）隐藏的。

若$View_{\mathcal{R}^*}^{\mathcal{C}_1(x)}(1^n)$与$View_{\mathcal{R}^*}^{\mathcal{C}_1(x')}(1^n)$同分布，则称是完美隐藏的。

– 绑定性：公开阶段运行$\langle \mathcal{C}_2(x), \mathcal{R}(v) \rangle(1^n)$，承诺者$\mathcal{C}^*$把承诺解释为对$x$的承诺，接收者的输入为$v = View_{\mathcal{R}^*}^{\mathcal{C}(commit, \cdot)}(1^n)$，输出记为$Out_{\mathcal{R}}(\mathcal{C}_2^*(x), \mathcal{R}(v))$。若对任意（*PPT*的）承诺者$\mathcal{C}^*$以及确定$v$，记

$$Pr\left[Out_{\mathcal{R}}(\mathcal{C}_2^*(x), \mathcal{R}(v)) = accept, Out_{\mathcal{R}}(\mathcal{C}_2^*(x'), \mathcal{R}(v)) = accept, x \neq x'\right] = \mu(n)$$

若μ可忽略，则称承诺是（计算）绑定的（当$\mu = 0$，也称为完美绑定的）。 ∎

承诺方案的隐藏性与绑定性是相互矛盾的两个性质，统计隐藏性与统计绑定不可能同时满足。一般来说，承诺方案要么是计算隐藏统计绑定的（简称为计算隐藏承诺），要么是统计隐藏计算绑定（简称为统计隐藏承诺），当然，也可以是计算隐藏与计算绑定的（取决于应用需要）。

另外，大多数承诺方案的公开阶段都是非交互的，承诺者公开被承诺的x与承诺时的随机输入，验证者验证正确性。如果承诺方案的承诺阶段和公开阶段都是非交互的，称为非交互承诺方案，它的应用很广泛。

定义 1.3.2 非交互计算隐藏比特承诺方案$\langle C, R \rangle$。在承诺阶段，发送者使用*PPT*算法Com生成对x承诺，即$c \leftarrow Com(x; r)$，并将对x的承诺c发送给接收者；在公开阶段，承诺者发送x与承诺时的随机数r，验证者验证承诺。其满足如下两个条件：

1) 统计绑定性：对任意$r, s \in_R \{0,1\}^{poly(n)}$，$Com(0; r) \neq Com(1; s)$，其中$n \in \mathbb{N}$为安全参数，$poly(.)$为任意多项式。

2) 计算隐藏性：$\{Com(0; U_{poly(n)})\} \stackrel{c}{=} \{Com(1; U_{poly(n)})\}$。 ∎

定理 1.3.2 若*1-1*单向函数存在，则存在计算隐藏完美绑定的非交互比特承诺方案。

证明：设f是1-1单向函数，b是其hard-core谓词。定义比特承诺算法$Com(\cdot,\cdot)$如下：对$\forall\sigma\in\{0,1\}$，承诺者随机选择$r\in\{0,1\}^*$，计算承诺

$$c=Com(\sigma;r)\stackrel{def}{=}(f(r),\sigma\oplus b(r))$$

公开时，承诺者发送(σ,r)，接收者验证上述等式成立。

由f是1-1单向函数易知完美绑定性成立，而计算隐藏性则可由hard-core谓词b的性质得到。 ∎

比特承诺的实现相对简单，而对任意$x\in\{0,1\}^*$的承诺可由比特承诺方案实现，即只需并行地对x的每一比特逐一进行承诺即可。

从实现来看，计算隐藏的承诺方案使用单向函数比较容易实现，而统计（完美）隐藏的承诺方案的实现要困难许多，但在一些更强的假设下，如Claw-free函数的存在，可以实现，而在一些具体的困难性假设下，也可以构造出简单的统计（完美）隐藏的承诺方案，如Pederson承诺方案。

Pederson承诺方案基于离散对数假设，是非交互承诺方案。设G是素数q阶的循环群，$g,h=g^{\alpha}$是两个生成元，要求α保密。承诺方案如下：

承诺阶段：承诺者对$x\in\mathbb{Z}_q^*$承诺。承诺者随机选择$r\in\mathbb{Z}_q^*$，计算承诺$c=g^xh^r$，并发送给接收者。

公开阶段：承诺者发送(x,r)，接收者验证$c=g^xh^r$。验证通过，则接受，否则拒绝。

Pederson承诺方案是完美隐藏的，因为对x的承诺$c=g^xh^r$在G上均匀分布。对任意的$x'\neq x$，存在$r'\in\mathbb{Z}_q^*$，使得$c=g^xh^r=g^{x'}h^{r'}$。方案的绑定性依赖于离散对数假设，若承诺者可以将对对$x\in\mathbb{Z}_q^*$承诺公开为对$x'\neq x$的承诺，则它可求出h的离散对数α。

统计隐藏计算绑定的承诺方案与统计零知识证明密切相关，因此它存在的条件自然会被关注，人们进行了许多的研究，希望能在单向函数存在的条件下实现。Halevi与Micali[6]在96年使用抗碰撞Hash函数构造了统计隐藏承诺方案，随后Naor[7]给出了基于单向置换的构造。基于单向函数的构造最早来自于Haitner等人的工作，他们基于正则（regular）单向函数给出统计隐藏承诺方案[8]，最终由Haitner与Reingold[9]在2006年给出了基于单向函数的构造，解决了基于单向函数的存在性问题。单向函数存在是承诺方案的最低假设，因为[10]证明安全承诺方案存在意味着单向函数存在。

下面给出Naor等人在[7]给出的统计隐藏的承诺方案。设f是$\{0,1\}^n$上的单向置换。

协议 1.3.1

输入：承诺者$\mathcal{C}$拥有输入$b \in \{0,1\}$。
承诺阶段：

- 承诺者$\mathcal{C}$随机选择$x \in \{0,1\}$，计算$y = f(x)$。
- 对$i = 1, \cdots, n-1$，双方完成以下过程：
  - 接收者$\mathcal{R}$随机选择$z_i \in \{0,1\}^{n-i}$，令$h_i = 0^{i-1}1z_i$，并将h_i发送给$\mathcal{C}$。
  - 承诺者$\mathcal{C}$计算$c_i =< h_i, y >$，并返回给$\mathcal{R}$。
- 承诺者$\mathcal{C}$求解线性方程$c_i =< h_i, y >$，$i = 1, \cdots, n-1$，得到的解记为y_0, y_1（字典次序），则y必是其中之一，不妨设$y = y_c$。承诺者$\mathcal{C}$发送$d = c \oplus b$给$\mathcal{R}$。

公开阶段：

承诺者$\mathcal{C}$发送x, b。接受者$\mathcal{R}$计算$c = d \oplus b$及y_0, y_1（字典次序），并验证：

$$y_c = f(x);\ c_i = \langle h_i, y_c \rangle, i = 1, \cdots, n-1$$

验证通过则接受，否则则拒绝。

隐藏性： 由于y_0, y_1都满足$c_i =< h_i, y >$（$i = 1, \cdots, n-1$），因此，$d = c \oplus b$不泄露b的任何信息，即承诺是完美隐藏的。

绑定性： 若发送者可以求出y_{1-c}在f下的原像，则它可违反绑定性，因此承诺是计算绑定的。

承诺方案与零知识证明有着十分密切的关系，一方面，承诺方案可以用来构造零知识协议，另一方面，零知识证明的存在也意味着某种承诺方案存在。承诺方案除了上述的基本形式外，还有许多变形或扩展形式，如陷门承诺、不可延展承诺、实例依赖承诺等，这里无法一一赘述。下面介绍利用承诺方案构造NP问题的零知识证明。

1.3.2 HC问题的零知识证明系统

设$G = (V, E)$为无向图，其中V与E分别为顶点集与边集，若存在E的子集$E_1 \subseteq E$，满足E_1中所有的边恰好组成一个封闭的环，而且恰好过V中每个顶点一次，则称图G包含Hamilton圈（回路）。令

$$HC = \{G : G\text{是无向图且包含Hamilton圈}\}$$

根据计算复杂性理论，HC问题：判定给定无向图G是否包含Hamilton圈，是一个NPC问题。Blum在1986年证明了若单向函数存在，则HC问题存在零知识证明系统。

假设图$G = (V, E)$包含n个顶点，其邻接矩阵记为$A = (a_{i,j})_{i,j=1}^{n}$，当$a_{i,j} = 1$时，表示$V$中存在链接顶点$i, j$的边，而$a_{i,j} = 0$则意味着$V$中不存在链接顶点$i, j$的边。设$H$为图$G$的一个Hamilton圈，为简单$H$中的边$e_{i,j}$记为$(i, j) \in H$。设$Com$为非交互计算隐藏统计绑定的承诺方案，HC问题的零

知识证明系统（称为Blum 协议）$\langle P, V\rangle$如下：

协议 1.3.2

公共输入：$G=(V,E)$，其邻接矩阵表示为$A=(a_{i,j})_{i,j=1}^{n}$。

- 证明者P对邻接矩阵A进行承诺：随机选择顶点集上的置换π，然后对$\forall i,j\in[n]$，随机选择$r_{ij}\in\{0,1\}^*$，计算$c_{i,j}=Com(a_{i,j};r_{i,j})$，最后将$\{c_{i,j}\}_{i,j=1}^{n}$发送给验证者$V$。
- V随机选择$\sigma\{0,1\}$，并将σ提交给P。
- 若$\sigma\notin\{0,1\}$，终止。若$\sigma=0$，P公开对A的承诺及置换π，即将$\{(a_{i,j},r_{i,j})\}_{i,j}$与$\pi$发送给$V$；若$\sigma=1$，$P$公开一个Hamilton 圈H 对应的承诺，即将$\{(a_{i,j},r_{i,j})\}_{(i,j)\in H}$发送给$V$。
- 若$\sigma=0$时，V验证所有公开的承诺是对$\pi(G)$的邻接矩阵的承诺；若$\sigma=1$时，V验证所有公开的承诺是对圈H的承诺。若验证通过，接受证明，否则拒绝。

定理 1.3.3 若承诺方案$Com(\cdot;\cdot)$是计算隐藏的，则协议*3.7.5*是*HC*问题的（计算）零知识证明系统。

证明： 设图$G=(V,E)$有$n=|V|$个顶点。

完备性： 显然，当$G\in HC$时，$\Pr[\langle P,V\rangle(x)=1]=1$。

合理性： 若$G\notin HC$，则无论证明者如何生成$\{c_{i,j}\}$，要么$\{c_{i,j}\}$所承诺的图不与G同构，要么$\{c_{i,j}\}$所承诺的图不包含Hamilton圈，所以若验证者随机选择$\sigma\in\{0,1\}$，则必有$\Pr[\langle P,V\rangle(x)=1]\leq\frac{1}{2}$。

零知识性： 首先定义算法$\mathcal{S}_0(G)$如下：

1) 首先为V^*选择随机输入r。

2) 随机选择$\tau\in\{0,1\}$，

   - 若$\tau=0$，按照证明者策略计算第一个消息，即随机选择顶点集上的置换π，然后对$\forall i,j\in[n]$，随机选择$r_{ij}\in\{0,1\}^*$，计算$c_{i,j}=Com(a_{i,j};r_{i,j})$；
   - 若$\tau=1$，随机确定一个Hamilton圈H，对其邻接矩阵（记为$H=(h_{i,j})$）进行承诺，即随机选择$r_{i,j}\in\{0,1\}^*$,计算$c_{i,j}=Com(h_{i,j};r_{i,j})$。

   令$m_1=\{c_{i,j}\}$。

3) 将$\{c_{i,j}\}_{i,j=1}^{n}$提交给验证者V^*，并收到V^*的回复$\sigma\in\{0,1\}$。若$\sigma\notin\{0,1\}$，输出(G,r,m_1)，终止。

4) 若$\tau\neq\sigma$，失败，输出$\perp$；否则，

– 若$\tau=\sigma=0$，令$m_2=(\{(a_{i,j},r_{i,j})\}_{i,j},\pi)$（公开对$\pi(A)$的承诺及置换$\pi$）；

– 若$\tau=\sigma=1$，令$m_2=\{(h_{i,j},r_{i,j})\}_{h_{i,j}=1}$（公开一个Hamilton圈H 对应的承诺）。

输出(G,r,m_1,m_2)，终止。

模拟器$\mathcal{S}(G)$定义为：重复运行$\mathcal{S}_0(G)$，直至$\mathcal{S}_0(G)$成功输出或运行次数达到n。

由于$\mathcal{S}_0(G)$失败的概率为$\Pr[\tau\neq\sigma]=1/2$，因此，$\mathcal{S}(G)$失败的概率为2^{-n}可忽略。当$\mathcal{S}(G)$成功时，$\mathcal{S}(G)$与$View^P_{V^*}(G)$的唯一不同在于$\{c_{i,j}\}_{h_{i,j}=0,a_{i,j}=1}$是对不同值的承诺，根据承诺方案的隐藏性，$\mathcal{S}(G)$的输出$(G,r,m_1,m_2)$或$(G,r,m_1)$与$View^P_{V^*}(G)$是计算不可区分的。 ∎

1.3.3 G3C问题的零知识证明

称图$G=(V,E)$是3色图，如果存在顶点的3着色方案，即用三种颜色为图的顶点着色，使得任意一条边的两个顶点的颜色不同。记$V=\{1,\cdots,n\}$，用$\{1,2,3\}$表示3种颜色，一个着色方案ψ就是由$V=\{1,\cdots,n\}$到$\{1,2,3\}$的映射，满足$(i,j)\in E\Rightarrow\psi(i)\neq\psi(j)$。令

$$G3C=\{G=(V,E):G\text{是三色图}\}$$

下面介绍G3C的零知识证明协议。证明者与验证者的公共输入为$G=(V,E)$，$|V|=n$，$\psi(\cdot)$是G的一个着色方案。设Π是一个$\{1,2,3\}$上的全体置换，$Com(\cdot,\cdot)$是一个非交互的承诺方案。

协议 1.3.3

– P首先随机选取$\pi\in_R\Pi$，得到新着色方案$\phi(i)=\pi(\psi(i))$。然后，随机选取$r_i\in\{0,1\}^*$，计算$c_i=C(\phi(i),r_i)$，$i=1,\cdots,n$。最后将$c_1,\cdots,c_n$发送给验证者V。

– V随机选择$(u,v)\in E$，并将u,v提交给P。

– P公开承诺c_u,c_v：即将$\{(\phi(u),r_u),(\phi(v),r_v)\}$发送给$V$。

– V验证承诺公开正确且$\phi(u)\neq\phi(v)$，若验证通过，接受证明，否则拒绝。

定理 1.3.4 若承诺方案$C(\cdot,\cdot)$具有*non-uniformly hiding*性质，则该协议是*3*色图的带辅助输入的零知识证明。

证明：完备性：若$G\in G3C$，则$Pr[\langle P(\psi),V(z)\rangle(G)=1]=1$.

可靠性：若$G\notin G3C$，则对任意的P^*，$Pr[\langle P^*(\psi),V(z)\rangle(G)=1]\leq 1-\frac{1}{|E|}$.

零知识性：对任意的V^*，构造基本模拟器$M^*(G)$如下：

– 随机选择$r\in\{0,1\}^*$作为V^*的辅助输入。记公共输入为G、随机输入为r的V^*的算法为$V(G,r;\cdot)$或者简记为V^*。

– 独立随机选择$e_1,\cdots,e_n\in\{1,2,3\}$以及$s_1,\cdots,s_n\in\{0,1\}^*$，然后计算$d_i=Com(e_i,s_i)$，$i=1,\cdots,n$，并以$c_1,\cdots,c_n$调用$V^*(G,r;\cdot)$，并收到$V^*$的回复$(u,v)$。

– 若$e_u \neq e_v$，输出$(G, r, (d_1, \cdots, d_n), (s_u, e_u, s_v, e_v))$，否则输出$\perp$（失败终止）。

下面需要证明$M^*(G)$满足定义1.2.2。

引理 1.3.5 对充分大的$n = |V|$，有$Pr[M^*(G) = \perp] \leq \frac{1}{2}$。

证明：为简便，记$C_{s_i}(e_i) = Com(e_i, s_i)$，$i = 1, \cdots, n$。设当$V^*$的输入为$G, r, (C_{s_1}(e_1), \ldots, C_{s_n}(e_n))$时，$V^*$选择挑战为$(u, v)$的概率为$p_{u,v}(G, r, (e_1, \cdots, e_n))$（关于$s_1, \cdots, s_n$）。

首先，对任意的$p(\cdot), r \in \{0,1\}^{q(n)}$，以及任意的序列$\alpha, \beta \in \{1,2,3\}^n$，当$|V|$充分大时，有

$$|p_{u,v}(G, r, \alpha) - p_{u,v}(G, r, \beta)| \leq \frac{1}{p(n)}$$

否则，有非一致（Non-Uniform）的PPT算法A使其区分不同着色的承诺的概率不可忽略，与假设矛盾。事实上，若存在$G_n, r_n \in \{0,1\}^{q(n)}$，以及$\alpha_n, \beta_n \in \{1,2,3\}^n$，使

$$|p_{u_n,v_n}(G_n, r_n, \alpha_n) - p_{u_n,v_n}(G_n, r_n, \beta_n)| > \frac{1}{p(n)}$$

则可构造算法簇$\{A_n\}$如下，其中A_n以$G_n, r_n, (u_n, v_n), \alpha_n, \beta_n$和$V^*$为辅助输入，设$y$是对$\alpha_n$或$\beta_n$的承诺，定义

$$A_n(y) = \begin{cases} 1 & V^*(G_n, r_n, y) = (u_n, v_n) \\ 0 & others \end{cases}$$

那么，对任意$e_1, \cdots, e_n \in_R \{1,2,3\}$，有

$$Pr[A_n(C_{U_n^{(1)}}(e_1), \cdots, C_{U_n^{(n)}}(e_n)) = 1] = p_{u,v}(G, r, (e_1, \cdots, e_n))$$

由此可知A_n可区分对α_n的承诺与对β_n的承诺，与承诺方案的安全性矛盾。

其次，设$\overline{e} = (e_1, \cdots, e_n)$为由模拟器给出的伪着色，令

$$E_{\overline{e}} = E_{(e_1, \cdots, e_n)} = \{(u, v) \in E : e_u = e_v\}$$

则有

$$\begin{aligned} Pr[M_r^*(G) = \perp] &= \sum_{\overline{e} \in \{1,2,3\}^n} \frac{1}{3^n} \cdot \sum_{(u,v) \in E_{\overline{e}}} p_{u,v}(G, r, \overline{e}) \\ &= \frac{1}{3^n} \cdot \sum_{(u,v) \in E} \sum_{\overline{e} \in B_{u,v}} p_{u,v}(G, r, \overline{e}) \\ &\leq \frac{1}{3^n} \cdot \sum_{(u,v) \in E} |B_{u,v}| \cdot \left(p_{u,v}(G, r, (1, \cdots, 1)) + \frac{1}{p(n)} \right) \end{aligned}$$

这里$B_{u,v} = \{(e_1, \cdots, e_n) \in \{1,2,3\}^n : e_u = e_v\}$，上面推导使用如下关系：

$$\{(\overline{e}, (u, v)) : \overline{e} \in \{1,2,3\}^n, (u, v) \in E_{\overline{e}}\} = \{(\overline{e}, (u, v)) : (u, v) \in E, \overline{e} \in B_{u,v}\}$$

。

由于$|B_{u,v}| = 3^{n-1}$，于是

$$Pr[M_r^*(G) = \perp] \leq \frac{1}{3} \cdot \frac{|E|}{p(n)} + \frac{1}{3} \cdot \sum_{(u,v) \in E} p_{u,v}(G, r, (1, \cdots, 1)) = \frac{1}{3} + \frac{1}{3} \cdot \frac{|E|}{p(n)}$$

Claim 1.3.5.1 对任意的*PPT*算法A和任意的多项式p，当$n=|V|$充分大时，有

$$|Pr[A(M^*(G))=1|M^*(G)\neq\perp]-Pr[A(view^P_{V^*}(G))=1]|<\frac{1}{p(|V|)}$$

证明：对任意算法A，令

$$\varepsilon_A(G)=|Pr[A(m^*(G))=1]-Pr[A(view^P_{V^*}(G))=1]|$$

需证明对任意A，$\varepsilon_A(G)$可忽略。

定义随机变量：

- $\mu_{u,v}(G)$：验证者的挑战为(u,v)时的$m^*(G)$，$q_{u,v}(G)=Pr[\mu_{u,v}(G)]$；
- $\nu_{u,v}(G)$：验证者的挑战为(u,v)时的$View^P_{V^*}(G)$，$p_{u,v}(G)=Pr[\nu_{u,v}(G)]$；

假设命题不成立，即存在算法A及多项式$poly(\cdot)$，对$G_n=(V_n,E_n)\in G3C$，使

$$\varepsilon_A(G_n)=|Pr[A(m^*(G_n))=1]-Pr[A(view^P_{V^*}(G_n))=1]|>\frac{1}{poly(|V_n|)}$$

分两种情况：

(I) 若存在多项式$p(\cdot)$，对无穷多个n，存在$G_n=(V_n,E_n),(u_n,v_n)\in E_n$，使得

$$|p_{u_n,v_n}(G)-q_{u_n,v_n}(G)|>\frac{1}{p(n)}$$

若如此，则可构造算法区分对不同着色的承诺的概率不可忽略，与假设矛盾，因此(I)不可能。

(II) 若(I)不成立，则对任意的多项式$p'(\cdot)$，只要$|V_n|$充分大$(G_n=(V_n,E_n))$，一定有

$$|p_{u,v}(G_n)-q_{u,v}(G_n)|\leq\frac{1}{p'(n)}$$

令$req_{u,v}(\alpha)$表示V^*选择的挑战等于(u,v)（输入为α时），那么存在$(u_n,v_n)\in E_n$，使

$$|Pr[A(m^*(G_n))=1\&req_{u_n,v_n}(m^*(G_n))]-Pr[A(view^P_{V^*}(G_n))=1\&req_{u_n,v_n}(view_{V^*}(G_n))]|\geq\frac{\varepsilon_A(G_n)}{|E_n|}$$

而

$$Pr[A(m^*(G_n))=1|req_{u_n,v_n}(m^*(G_n))]=Pr[A(\mu_{u_n,v_n}(G_n))=1]$$
$$Pr[A(view^P_{V^*}(G_n))=1|req_{u_n,v_n}(view_{V^*}(G_n))]=Pr[A(\nu_{u_n,v_n}(G_n))=1]$$

令$p(|V_n|)\overset{def}{=}\frac{2|E_n|}{poly(|V_n|)}$，从而有

$$|p_{u,v}\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]-q_{u_n,v_n}\cdot Pr[A(\nu_{u_n,v_n}(G_n))=1]|\geq\frac{\varepsilon_A(G_n)}{|E_n|}=\frac{2}{p(V_n)}$$

进一步，

$$\begin{aligned}&|q_{u_n,v_n}\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]-q_{u_n,v_n}\cdot Pr[A(\nu_{u_n,v_n}(G_n))=1]|\\&\geq\tfrac{2}{p(V_n)}-|p_{u_n,v_n}\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]-q_{u_n,v_n}\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]|\\&\geq\tfrac{2}{p(V_n)}-|p_{u_n,v_n}-q_{u_n,v_n}|\end{aligned}$$

记$p\prime = 3p^2$，那么当n充分大时，有$|p_{u_n,v_n} - q_{u_n,v_n}| < \frac{1}{3p(|V_n)^2}$，于是

$$|q_{u_n,v_n} \cdot Pr[A(\mu_{u_n,v_n}(G_n)) = 1] - q_{u_n,v_n} \cdot Pr[A(\nu_{u_n,v_n}(G_n)) = 1]| > \frac{1}{p(V_n)}$$

综上，存在PPT算法A，多项式$p(\cdot)$，无穷多个n以及与之对应的$G_n = (V_n, E_n), (u_n, v_n) \in E_n$，满足

- $q_{u_n,v_n}(G_n) > \frac{1}{p(n)}$
- $|p_{u_n,v_n}(G_n) - q_{u_n,v_n}(G_n)| < \frac{1}{3p(n)^2}$
- $|Pr[A(\mu_{u_n,v_n}(G_n)) = 1] - Pr[A(\nu_{u_n,v_n}(G_n)) = 1]| > \frac{1}{p(n)}$

下面，由此构造可攻击承诺方案的算法，从而导出矛盾。详情如下：

设ψ_n是$G_N = (V_n, E_n)$的一个3着色，$V_n = (1, \cdots, n)$。构造带有辅助输入$V^*, A, G_n, \psi_n, (u_n, v_n)$的算法序列（即为非一致算法）

$$D_n^{(V^*,A,G_n,\psi_n,(u_n,v_n))}$$

使其可以区分对$\alpha = 1^n 2^n 3^n$的承诺与对任意$\beta \in_R \{1,2,3\}^{3n}$的承诺，即存在多项式$p''(\cdot)$，使

$$|Pr[D_n^{(V^*,A,G_n,\psi_n,(u_n,v_n))}(C(\alpha)) = 1] - Pr[D_n^{(V^*,A,G_n,\psi_n,(u_n,v_n))}(C(\beta)) = 1]| > \frac{1}{p''(n)}$$

这里为方便，对任意$(e_1, \cdots, e_t)$，记$C(e_1, \cdots, e_t) = C_{s_1}(e_1), \cdots, C_{s_t}(e_t)$。区分算法$D_n$如下：

$D_n^{V^*,A,G_n,\psi_n,(u_n,v_n)}(y = (y_1, \cdots, y_{3n}))$

1) 随机选择$\{1,2,3\}$上的置换π，计算$\phi(i) = \pi(\psi_n(i))$，$i = 1, \cdots, n$；
2) 对$i \in V_n - \{u_n, v_n\}$，令$c_i = y_{\phi(i)\cdot n - n + i}$；
3) 随机选择s_{u_n}, s_{v_n}，计算$c_{u_n} = C_{s_{u_n}}(\phi(u_n)), c_{v_n} = C_{s_{v_n}}(\phi(v_n))$；
4) 随机选择$r \in \{0,1\}^{q(n)}$，以$G_n, r, (c_1, \cdots, c_n)$调用验证者算法$V^*$，得到$V^*$的输出，记为$m$；
5) 若$m \neq (u_n, v_n)$，输出0；
6) 若$m = (u_n, v_n)$，输出$A\left(G_n, r, (c_1, \cdots, c_n), (s_{u_n}, \phi(u_n), s_{v_n}, \phi(v_n))\right)$。

说明：

- 若y来自于$C(1^n 2^n 3^n)$，算法中得到的$c_1, \cdots, c_n$与证明者给出的承诺相同；
- 若y来自于$C(\alpha), \alpha \in_R \{1,2,3\}^{3n}$，算法中得到的$c_1, \cdots, c_n$与模拟器给出的顶点$u_n, v_n$不同色时的承诺相同；

由D的构造，不难得到

(1) D_n的规模是多项式的；

(2) $Pr[D_n(C(1^n2^n3^n))=1]=q_{u_n,v_n}(G_n)\cdot Pr[A(\nu_{u_n,v_n}(G_n))=1]$;

(3) $Pr[D_n(C(T_{3n}))=1]=p'_{u_n,v_n}(G_n)\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]$;
$|p'_{u_n,v_n}(G_n)-p_{u_n,v_n}(G_n)|<\frac{1}{3p(n)^2}$;

(4) $|p'_{u_n,v_n}(G_n)-q_{u_n,v_n}(G_n)|<|p'_{u_n,v_n}(G_n)-p_{u_n,v_n}(G_n)|+|p_{u_n,v_n}(G_n)-q_{u_n,v_n}(G_n)|<\frac{2}{3p(n)^2}$;

从而得

$$\begin{aligned}
& |Pr[D_n(C(1^n2^n3^n))=1]-Pr[D_n(C(T_{3n}))=1]| \\
= \;& |q_{u_n,v_n}(G_n)\cdot Pr[A(\nu_{u_n,v_n}(G_n))=1]-p\prime_{u_n,v_n}(G_n)\cdot Pr[A(\mu_{u_n,v_n}(G_n))=1]| \\
\geq \;& q_{u_n,v_n}|Pr[A(\nu_{u_n,v_n})=1]-Pr[A(\mu_{u_n,v_n})=1]|-|p'_{u_n,v_n}(G_n)-q_{u_n,v_n}(G_n)| \\
> \;& \frac{1}{p(n)}\cdot\frac{1}{p(n)}-\frac{2}{3p(n)^2}=\frac{1}{3p(n)^2}
\end{aligned}$$

这与承诺方案的安全相矛盾，因此(II)也不可能成立，由此得引理成立。

综上可知M^*是满足定义的模拟器，于是知协议1.3.3是G3C问题的零知识证明。

1.3.4 NP问题的零知识证明

由于HC问题与G3C问题均是NP完全问题，因此，任意的$L\in NP$，都可归约于HC或G3C问题。如归约HC问题，对任意$L\in NP$，存在多项式时间可计算的函数$f_L(\cdot),g_L(\cdot,\cdot)$，使得

(1) $x\in L\Leftrightarrow f_L(x)\in HC$

(2) $(x,e)\in R_L\Rightarrow(f_L(x),g_L(x,w))\in R_{HC}$

因此，可以将对L证明转化为对HC问题的证明，于是由HC问题的零知识证明$\langle P',V'\rangle$得到对L的零知识证明$\langle P,V\rangle$：

- 公共输入：$x\in L$。
- 证明者P与验证者V各自独立计算$G=f_L(x)$。证明者求出G中包含的Hamilton圈H。
- 双方运行协议$\langle P',V'\rangle(G)$，其中P充当P'，其中V充当V'。
- V接受当且仅当V'接受。

定理 1.3.6 若$\langle P',V'\rangle$是*HC*问题的带辅助输入的零知识证明，则以上协议也是L的带辅助输入的零知识系统。

定理 1.3.7 若具有（非一致）单向函数的存在，则任意NP问题都存在带辅助输入的零知识证明系统。而且，若*PPT*的证明者拥有实例的证据，则它可在概率多项式时间完成证明。

定理1.3.7是零知识证明在密码学中有着广泛的应用的基础。一方面，在密码学中的问题大都是NP问题，如证明两个密文是同一个明文的密文、证明承诺值满足某种条件等；另一方面，单向函数存在是密码学的最低假设，因此，诸多密码学问题自然隐含着零知识证明的存在。

例如，设$Com(\cdot)$是计算隐藏的非交互承诺方案，令$c = Com(\alpha; r)$，假如我们希望在不公开承诺的条件下证明c对应的承诺值α大于某个确定值α_0（即满足$\alpha > \alpha_0$）。令

$$L = \{(c, \alpha_0) : \text{存在}\alpha\text{及}r\text{满足：}\ c = Com(\alpha; r), \alpha > \alpha_0$$

证明c对应的承诺值α大于某个确定值α_0就是证明$(c, \alpha_0) \in L$。显然，$L \in NP$，因此根据定理知存在零知识证明（已无需公开承诺的证明）。一般来说，对任意NP问题的证明都归约至某个NPC问题，如HC或G3C问题，进行证明，在理论上可行，但方案往往效率较低，并不适合实际应用。

关于零知识证明的效率，主要取决于协议的通信复杂度与计算复杂度，而通信复杂度包括交互次数（称为轮数）和消息的总长度（比特），其中轮数是一个重要的指标。在实际应用时，总希望根据所证明命题的特殊性，设计出更高效的证明系统，特别是希望得到轮数尽可能低的证明系统。

HC问题的零知识证明（协议1.3.2）的合理性的错误概率为1/2，根据定理1.2.3，n（图的顶点数，作为协议的安全参数）次顺序复合协议1.3.2，即可得到HC问题的错误概率为2^{-n} 的零知识证明系统。协议1.3.2需要交互3次（称为3轮），因此复合协议就要交互$3n$次（$3n$轮），即复合协议的交互次数不是常数，而是随着安全参数的增加而增加。如果使协议1.3.2的n拷贝并行地运行，所得复合协议的错误概率也是2^{-n}，而且复合协议仍交互是3轮，但不幸地是并行复合协议的零知识性无法证明。

1996年，Goldreich等人首先构造出了G3C常数轮零知识证明系统，从而也就得到了任意NP问题的常数轮零知识证明系统。

1.3.5　IP的零知识证明

在单向函数存在的假设下（密码学所需的最低假设），任意$L \in IP$都存在零知识证明系统[2]，即有如下定理。

定理 1.3.8 若具有（非一致）单向函数的存在，则任意IP问题都存在带辅助输入的零知识证明系统。

为证明这个定理，需要引入一类特殊的零知识证明，称为公开投币（public-coin）零知识证明。回顾HC问题的零知识证明（协议1.3.2）与G3C问题的的零知识证明（协议1.3.3），验证者都只是发送随机选择的挑战给证明者。设$\langle P, V\rangle$是L的交互证明系统，若V在与P交互时，总是将随机投币的结果发送给P，则称$\langle P, V\rangle$是一个公开投币交互证明；另外，若其还具有零知识性，则称是公开投币零知识证明系统。

公开投币交互证明也称为Arthur-Merlin Game，最早由Babai与Moran（1988）提出。若用AM表示可以用Arthur-Merlin games证明的语言类，显然$IP \supseteq AM$。Goldwasser与Sipser给出了由一般交互证明到公开投币证明的转化方法，从而证明了公开投币证明与一般交互证明等价，即IP=AM。

设$L \in IP$，根据定义L存在一个交互证明系统，记为$\Pi = \langle P, V \rangle$。由于IP=AM，因此可以假设$\Pi$是一个公开投币证明。下面给出将$\Pi$转化为零知识证明的方法。由于假设单向函数存在，因此存在计算隐藏的非交互承诺方案，记为$Com(\cdot;\cdot)$。假设$\langle P, V \rangle(x)$的交互过程如下：

$$
\begin{array}{rcl}
P(x) & & V(x) \\
 & \xleftarrow{\alpha_1} & \alpha_1 \leftarrow_R \{0,1\}^* \\
\beta_1 \leftarrow P(x, \alpha_1) & \xrightarrow{\beta_1} & \\
 & \cdots & \\
 & \xleftarrow{\alpha_k} & \alpha_k \leftarrow_R \{0,1\}^* \\
\beta_k \leftarrow P(x, \alpha_1, \cdots, \alpha_k) & \xrightarrow{\beta_k} &
\end{array}
$$

将$\langle P, V \rangle(x)$转化为零知识证明的过程如下：

协议 1.3.4 *Transform: from IP to ZKP*

设$\langle P, V \rangle(x)$为L的AM证明，构造新交互证明$\langle P', V' \rangle(x)$如下：

1) 以密文方式运行Π：从$i = 1$到k，完成如下交互过程：

   - V'运行V，发送α_i；
   - P'运行P，$\beta_i \leftarrow P(x, \alpha_1, \cdots, \alpha_i)$，并选择随机数$r_i$，计算$c_i = Com(\beta_i; r_i)$，最后将$c_i$发送给$V'$。

2) 归约并运行NP问题的零知识证明：记$X = ((\alpha_1, c_1), \cdots, (\alpha_k, c_k))$。定义语言类$L'$如下：

$$
L' = \left\{ \begin{array}{ll} (a_1, s_1), \cdots, (a_k, s_k): & \exists (b_1, r_1), \cdots, (b_k, r_r), satisfying \\ & (1)\ s_i = Com(b_i, r_i), i = 1, \cdots, k \\ & (2)\ V((a_1, b_1), \cdots, (a_k, b_k)) = 1 \end{array} \right\}
$$

   显然，$L' \in NP$。由定理知L'存在零知识证明系统，设为$\langle P'', V'' \rangle$。P'与V'分别充当P''与V''，运行$\langle P'', V'' \rangle(X)$证明$X \in L'$。

3) 最后，V'接受当且仅当V''接受。

交互证明$\langle P', V' \rangle(x)$的完备性与可靠性显然成立，要证明定理1.3.8，只需构造合适的模拟器$\mathcal{S}'(x)$即可。设$\langle P'', V'' \rangle(X)$ 的模拟器为$\mathcal{S}''$，由$\mathcal{S}''$构造模拟器$\mathcal{S}'$如下过。

$\mathcal{S}'(x)$完成如下过程：

- 为V'选择随机输入r；
- 充当P'，诚实地完成协议的1)；
- 由协议1)协议得到X，运行$\mathcal{S}_{''}(X)$；

– 最后输出$(x, X, \mathcal{S}_{''}(X))$

下面需要说明$\mathcal{S}'(x)$与$View_{V'}^{P'}(x)$计算不可区分。（略）

1.4　零知识论证及其它

1.4.1　零知识论证

零知识论证（zero knowledge argument）是零知识证明的一种弱化形式。在一般的交互证明中，我们总假定证明者具有无限计算能力，而验证者计算能力是多项式时间的，如果要求双方都只有多项式时间的计算能力，称其为交互论证（interactive argument）。交互论证具有计算可靠性（computaional soundness），而交互证明具有完美可靠性（perfect soundness)。当交互论证还满足零知识性时，即称为零知识论证。

定义 1.4.1（交互论证）交互系统$\langle P, V\rangle$称为语言L的成员识别问题的交互论证系统，若P, V是多项式时间的交互图灵机，且满足如下两个条件。

1) 完备性*(Completeness)*：对任意$x \in L$，有

$$\Pr[\langle P, V\rangle(x) = 1] \geq \frac{2}{3}。$$

2) 计算可靠性*(Computional Soundness)*：对任意$x \notin L$以及任意多项式时间的交互图灵机B，有

$$\Pr[\langle B, V\rangle(x) = 1] < \frac{1}{3}。$$

其中的概率是关于*P*与*V*的随机输入（投币）。

与一般交互证明类似，定义中错误概率上界1/3并无实质意义，定义与下面更一般的定义等价。

定义 1.4.2（交互论证）设函数$c, s : N \to [0, 1]$满足$c(n) < 1 - 2^{-poly(n)}$、$s(n) > 2^{-poly(n)}$，且$c(n) > s(n) + \frac{1}{p(n)}$，其中$poly(\cdot)$为（不确定）任意多项式。交互协议$\langle P, V\rangle$ 称为语言L 的成员识别问题的交互证明系统，若P, V 是多项式时间的交互图灵机，且满足如下两个条件。

1) 完备性*(Completeness)*：对任意$x \in L$，有

$$\Pr[\langle P, V\rangle(x) = 1] \geq c(|x|)$$

2) 计算可靠性*(Soundness)*：对任意$x \notin L$以及任意多项式时间的交互图灵机B，有

$$\Pr[\langle B, V\rangle(x) = 1] < s(|x|)$$

其中的概率是关于*P*与*V*的随机输入（投币）。通常称$e(|x|) = max\{s(|x|), 1 - c(|x|)\}$为交互证明错误概率。

交互论证的零知识性也可分为完美零知识、统计零知识和计算零知识，其定义与交互证明的零知识性一样，无需赘述。若交互论证还满足（完美、统计）零知识性，则称其为（完美、统计）零知识论证（(perfect or statistical) zero knowledge argument）。

零知识性与可靠性是交互证明（论证）的安全性的两个方面。一般相信，所有的NP问题不大可能存在具有完美零知识性与完美合理性的交互协议，而完美零知识论证在一定假设下是存在的。注意到在HC问题的零知识证明（协议1.3.2）中，要求承诺方案是计算隐藏统计绑定的。承诺的隐藏性保证验证者不能从承诺获取任何信息，由于验证者也只有多项式时间的计算能力，而且只要求使模拟器的输出与$View_{V^*}^P(x)$是计算不可区分，所以只需计算隐藏性。但是，证明者具有无限计算能力，所以需要统计绑定性，以保证证明者不能欺骗。如果只需得到HC问题的统计零知识论证，可将协议1.3.2中的承诺方案替换为计算绑定的承诺方案即可。事实上，此时证明者仅有多项式时间的计算能力，因此需要承诺方案具有计算绑定性就可以满足要求。

定理 1.4.1 如果将协议*1.3.2*中的承诺方案替换为计算绑定（统计隐藏）的承诺方案，则得到*HC*问题的统计零知识论证系统。

若希望得到统计零知识的，即模拟器的输出与$View_{V^*}^P(x)$ 是统计不可区分，因此承诺方案应该具有统计隐藏性。由于同时具有统计隐藏性和统计绑定性的承诺方案不存在，因此，统计零知识证明对那些语言类存在尚不清楚，已经证明若NPC问题具有统计零知识证明，则计算复杂性的多项式分层坍塌。

1.4.2 其它

根据证明者的能力，零知识协议分为两类，称为证明（proof）与论证（argument）；根据模拟器的输出与验证者的View的三种不可区分性：同分布、统计不可区分、计算不可区分，每一类的零知识性又可分为三种，分别是完美零知识、统计零知识与计算零知识（通常就称为零知识），计算零知识一般就称为零知识。

$$\left.\begin{array}{l}\{\mathcal{S}(x)\}\equiv\{View_{V^*}^P(x)\}\Leftrightarrow Perfect\\ \{\mathcal{S}(x)\}\stackrel{s}{=}\{View_{V^*}^P(x)\}\Leftrightarrow Statistical\\ \{\mathcal{S}(x)\}\stackrel{c}{=}\{View_{V^*}^P(x)\}\Leftrightarrow Computational\end{array}\right\} Zero{-}Knowledge\left\{\begin{array}{r}Proof\Leftrightarrow Unbounded\ P\\ Argument\Leftrightarrow PPT\ P\end{array}\right.$$

每种零知识协议对应一个复杂类，这样得到6个复杂类：

PZKP、SZKP、CZKP、PZKA、SZKA、CZKA

语言L具有那种零知识协议，就称L属于对应的复杂类，如

SZKP $=\{L:$ 存在统计零知识证明（SZKP）$\}$

通常所说的零知识证明均指CZKP，简记为ZKP。在SZKP、CZKP、SZKA 与CZKA 这四类零知识协议中，SZKP 有着重要意义，由它可以刻画出其它三类的特征。

交互证明所针对的问题是成员判定问题，即证明$x\in L$是成立，这可以用更一般的承诺问题（Promise problem）来表示。设$\Pi=(\Pi_Y,\Pi_N)$为集合二元组，其中$\Pi_Y\cap\Pi_N=\emptyset$，Promise problem就是对已知$x\in\Pi_Y\cup\Pi_N$，判定$x\in\Pi_Y$是否成立，对应的交互证明就可以定义为：

定义 1.4.3（交互证明）交互系统$\langle P,V\rangle$称为$\Pi=(\Pi_Y,\Pi_N)$的交互证明系统，若V是多项式时间的，且满足如下两个条件。

1) 完备性*(Completeness)*：对任意$x\in\Pi_Y$，有

$$\Pr[\langle P,V\rangle(x)=1]\geq\tfrac{2}{3}。$$

2) 可靠性*(Soundness)*：对任意$x\in\Pi_N$以及任意交互图灵机B，有

$$\Pr[\langle B,V\rangle(x)=1]<\tfrac{1}{3}。$$

其中的概率是关于*P*与*V*的随机输入（投币）。

$x\in L$的成员判定问题可以看成$\Pi=(L,\overline{L})$的Promise问题，以下将不再加以区分。显然PZKP $\subseteq$ SZKP $\subseteq$ CZKP，PZKA $\subseteq$ SZKA $\subseteq$ CZKA。

零知识性是针对任意恶意的验证者V^*，如果限制验证者的行为，则可得到较弱的零知识性，如诚实验证者零知识性。所谓诚实验证者零知识，是限制验证者严格按照协议规定执行协议下的零知识（验证者通过在诚实交互过程得不到任何“知识”）。用HVSZKP表示所有具有诚实验证者统计零知识证明的复杂类，类似地可定义HVPZKP、HVCZKP，以及HVPZKA、HVSZKA、HVCZKA。

显然，从零知识性的定义看，不难得到HVSZKP $\subseteq$ SZKP与HVCZKP $\subseteq$ CZKP （其它几类也有类似关系），但Goldreich与Vadhan等人[21, 22]给出了将诚实验证者统计（计算）零知识证明转化为一般的统计（计算）零知识证明的方法，从而证明了HVSZKP=SZKP及$HVCZKP=CZKP$。而且还可以要求SZKP或CZKP 是公开投币的。

根据前面的结论，在单向函数存在的条件下，$NP\subseteq$ CZKP,IP $\subseteq$ CZKP。但对于SZKP来说，由于获得了更强的零知识性，人们并不知道是否所有NP问题都存在SZKP，不过一般相信$NP\subseteq$ SZKP不成立[19, 20]。尽管如此，SZKP仍有许多重要应用，因为仍有许多重要的问题都存在SZKP。特别重要的是，SZKP存在完备问题。

1.4.3　SZKP完备问题

理论计算机科学的重要结果之一是发现NP存在完备问题，如G3C、HC问题等，这样对NP问题的研究就可转化为对某个具体问题的研究。Goldreich与Vadhan（1999）首先定义了熵差异（Entropy Difference）问题，并证明了它是HVSZKP的完全问题。后来，Sahai与Vadhan于2003 年给出HVSZKP的另一个完全问题，称其为统计差（Statistical Difference）。

1.4.3.1　统计差问题

简单地说，统计差问题就是判定两个可有效抽样的随机分布的统计距离是大还是小。设X为具有m比特输入、n比特输出的布尔电路，当输入在$\{0,1\}^m$上均匀分布时，其输出是$\{0,1\}^n$的一个随机分布，也称为由X抽样的分布。为了表示简单，以下直接使用X表示由它抽样的分布。

定义 1.4.4 统计差是一个*Promise*问题，$SD=(SD_Y,SD_N)$，其中

$$SD_Y=\{(X,Y):\Delta(X,Y)\geq\frac{2}{3}\}$$

$$SD_N = \{(X,Y) : \Delta(X,Y) \leq \frac{1}{3}\}$$

这里$\Delta(X,Y)$表示X与Y之间的统计距离。

定理 1.4.2 统计差*Promise*问题$SD = (SD_Y, SD_N)$是*SZKP*的完备问题。

定理的证明需要证明两个结论：

1) $\Pi = (SD_Y, SD_N) \in SZKP$系统；

2) 任意$\Pi \in SZKP$可归约至$\Pi = (SD_Y, SD_N)$。

Promise问题$SD = (SD_Y, SD_N)$定义中的常数2/3与1/3并无实质意义。设$1 \geq \alpha > \beta \geq 0$，令

$$SD_Y^{\alpha,\beta} = \{(X,Y) : \Delta(X,Y) \geq \alpha\}$$

$$SD_N^{\alpha,\beta} = \{(X,Y) : \Delta(X,Y) \leq \beta\}$$

则$SD^{\alpha,\beta} = (SD_Y^{\alpha,\beta}, SD_N^{\alpha,\beta})$是$SD$的更一般的变形。特别地，取$\alpha = 1, \beta = 0$，则得$SD^{1,0}$，即判定两个随机分布是同分布还具有不同的支撑集。

1.4.3.2 熵差异问题

设X, Y为随机分布（同上），称Promise问题$ED = (ED_Y, ED_N)$为熵差异问题，其中

$$ED_Y = \{(X,Y) : H(X) \geq H(Y) + 1\}$$

$$ED_N = \{(X,Y) : H(Y)) \geq H(X) + 1\}$$

$H(\cdot)$为Shannon熵。

定理 1.4.3 熵差异*Promise*问题$ED = (ED_Y, ED_N)$是*SZKP*的完备问题。

1.4.4 零知识证明与单向函数

单向函数存在对许多密码方案来说是必要和充分的，如伪随机生成器、私钥加密及签名等。依据Shamir[3]及Ben-OR[2]的结论，若单向函数存在，则CZKP=IP=PSPACE，但零知识证明是否隐含单向函数的存在成为关注的问题。Ostrovsky[25]在1991年首先证明了若HVSZKP包含困难问题，则单向函数存在，后来Ostrovsky与Wigderson（1993）将此结果推广至HVCZKP。

定义 1.4.5 称*Promise*问题Π是平均困难的（*hard on average*），如果存在可有效抽样分布$\{D_n\}_n$，使得对任意非一致*PPT*算法$\mathcal{A}$，及任意$x \in \Pi_Y \cup \Pi_N$，有

$$\Pr_{x \leftarrow D_n}[\mathcal{A}(x) = \chi(x)] \leq \frac{1}{2} + negl(n)$$

其中$\chi(x) = \begin{cases} 1, & x \in \Pi_Y \\ 0, & x \in \Pi_N \end{cases}$ 这里概率是关于$\mathcal{A}$的随机投币与D_n的抽样。

定理 1.4.4 （*[25]*）若存在平均困难问题L，使$L \in$ *HVSZKP*，则单向函数存在。

定理 1.4.5 （*[26]*）若存在平均困难问题L拥有公开投币的*HVCZKP*，则单向函数存在。

由于CZKP=HVCZKP，定理1.4.5说明单向函数与CZKP的关系，而1.4.4说明对统计零知识证明，单向函数存在是必要的。另一方面，单向函数存在是否意味着统计零知识证明的存在尚不清楚。

1.4.5　零知识协议的特征

SZKP具有比较好的结构，如Okamoto[27]证明了$L \in SZKP \Leftrightarrow \overline{L} \in SZKP$（$\overline{L}$表示$L$的补）。由它可以刻画零知识协议的本质特征。2006 年，Vadhan定义了一个称为SZKP-OWF 的条件，也称之为Vadhan条件（Vadhan condition）。

定义 1.4.6 称*Promise*问题Π满足*Vadhan*条件，如果存在I, J，满足

- $\Pi' = (\Pi_Y \backslash I, \Pi_N \backslash J) \in SZKP$;
- 对任意$x \in I$，由实例x可获得一个单向函数f_x;
- 对任意$x \in J$，由实例x可获得一个单向函数g_x;

Vadhan[28]与Ong[29]给出了下面的定理。

定理 1.4.6 设$\Pi = (\Pi_Y, \Pi_N)$为*promise*问题，则有

- $\Pi \in CZKP$当且仅当Π满足$J = \emptyset$的*Vadhan*条件。*(Vadhan,2004,[28])*
- $\Pi \in CZKA$当且仅当Π满足*Vadhan*条件。*(S.J.Ong,2007,[29])*
- $\Pi \in SZKA$当且仅当Π满足$I = \emptyset$的*Vadhan*条件。*(S.J.Ong,2007,[29])*

1.5　证据不可区分证明

NP问题的交互证明的证据不可区分性（witness indistinguishability）弱于零知识性。设$L \in NP$，根据计算复杂性分类，L 有多项式有界的二元关系R，对任意$x \in L$，存在NP 证据w，使得$R(x, w) = 1$。对任意$x \in L$，令$R(x) = \{w : R_L(x, y) = 1\}$，即成员$x$的证据集合。通常，直接用$R_L$表示$L$对应的NP关系，或者用$L_R$表示NP关系$R$ 对应的语言类，以明确二者的对应关系。

定义 1.5.1 （证据不可区分）设$\langle P, V\rangle$是$L \in NP$的（带辅助输入）交互证明系统。称$\langle P, V\rangle$是关于L的*NP*关系R是证据不可区分的（*witness indistinguishable*），如果对任意的*PPT*交互机V^*和两个证据序列$W^1 = \{w_x^1\}_{x \in L}, W^2 = \{w_x^2\}_{x \in L}$，满足$R(x, w_x^1) = R(x, w_x^2) = 1$，以下两个随机分布计算不可区分;

– $\{\langle P(w_x^1), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$：当证明者用证据$w_x^1$时验证者的输出。

– $\{\langle P(w_x^2), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$：当证明者用证据$w_x^2$时验证者的输出。

如果$\{\langle P(w_x^1), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$与$\{\langle P(w_x^2), V^*(z)\rangle(x)\}_{x\in L, z\in\{0,1\}^*}$同分布，则称$\langle P, V\rangle$是证据独立证明。

直观上，证据不可区分保证验证者不能区分证明者使用x哪一个证据完成证明，这显然比零知识性的要求低。不难证明零知识证明一定是证据不可区分的，但反之就不一定成立。

Okamoto协议(1992)

设$G = \langle g\rangle$是q(素数)阶循环群，$g_1, g_2 \in G$，并且双方都不知道$\log_{g_1} g_2$。定义

$$R = \{(h, w_1, w_2) : \exists w_1, w_2 \in \mathbb{Z}_n, h = g_1^{w_1} g_2^{w_2}\}$$

对应的语言类记为L。对关系R的证据不可区分证明协议（Okamoto协议）如下：

公共输入：h;
证明者P的辅助输入：(w_1, w_2)。

– P随机选择$u_1, u_2 \in_R \mathbb{Z}_n$，计算$a = g_1^{u_1} g_2^{u_2}$，并将$a$发送给验证者$V$;

– V随机选择挑战$c \in_R \mathbb{Z}_n$，并返回给P;

– P计算$z_1 = u_1 + cw_1$，$z_2 = u_2 + cw_2$，然后以z_1, z_2回复V。

– V验证$g_1^{z_1} g_2^{z_2} \stackrel{?}{=} ah^c$，验证通过接受；否则拒绝。

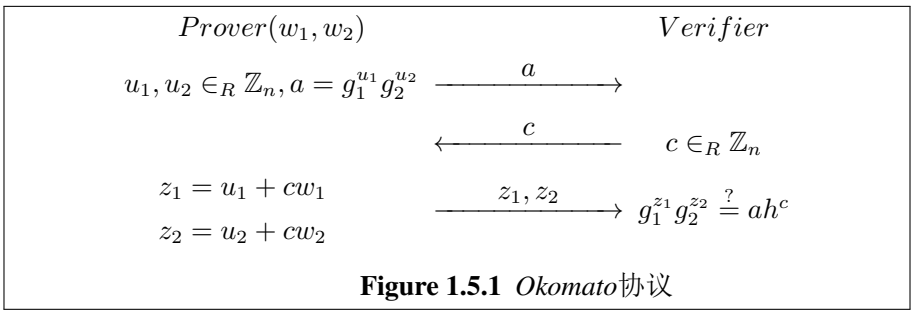


Figure 1.5.1 *Okomato*协议

容易看出，对任意指定的h，满足条件$h = g_1^{w_1} g_2^{w_2}$的(w_1, w_2)有许多，也就是h有许多满足关系R的证据。证明者以某个证据作为辅助输入，可以有效地完成协议，形式上证明者对挑战的回复与证据相关，而实质上二者是相互独立的。

完备性：若双方诚实运行协议，当$h \in L_R$时，条件$g_1^{z_1} g_2^{z_2} = ah^c$必定满足，即$\Pr[\langle P, V\rangle(h) = 1 | h \in L_R] = 1$。

可靠性：在给出a后，如果证明者可以正确回复验证者的两个不同的挑战$c_1 \neq c_2$，则它将有能力求出$h \in L_R$的一个证据。这说明若$h \notin L_R$时，证明者最多只能正确回复一个挑战，因此，$\Pr[\langle P, V\rangle(h) = 1 | h \notin L_R] \leq \frac{1}{n}$。

证据不可区分性：设(a,c,z_1,z_2)是证明者采用证据(w_1,w_2)时的一个证明，只需说明(a,c,z_1,z_2)的分布与证据无关即可。对另一个证据(w_1',w_2')，若取$u_1'=u_1+c(w_1-w_1'),u_2'=u_2+c(w_2-w_2')$，利用$h=g_1^{w_1}g_2^{w_2}=g_1^{w_1'}g_2^{w_2'}$可得：$a'=g_1^{u_1'}g_2^{u_2'}=g_1^{u_1}g_2^{u_2}=a$，而此时有$z_1'=u_1'+cw_1'=z_1,z_2'=u_2'+cw_2'=z_2$，即对证据$(w_1',w_2')$有一个与$(a,c,z_1,z_2)$相同的证明。因此，无论使用哪个证据，交互的信息具有完全相同的分布。

另一方面，注意到G中任意的h均可以表示为$h=g_1^{w_1}g_2^{w_2}$，因此Okamoto协议实际上是要证明者证明自己拥有某个证据。其合理性保证，若$\Pr[\langle P,V\rangle(h)=1|h\notin L]>\frac{1}{n}$，则证明者一定可以拥有某个证据。

证据不可区分性可以进一步加强，允许验证者在得到证据条件下仍然不能区分证明者使用哪个证据，称为强证据不可区分性。

定义 1.5.2（强证据不可区分）设$\langle P,V\rangle$是$L\in NP$的（带辅助输入）交互证明系统。称$\langle P,V\rangle$关于关系R_L是强证据不可区分的，如果对任意的*PPT*交互机V^*和$R_L\times\{0,1\}^*$上的两个随机序列$\{X_n^1,W_n^1,Z_n^1\},\{X_n^2,W_n^2,Z_n^2\}$，若$\{X_n^1,Z_n^1\}$与$\{X_n^2,Z_n^2\}$计算不可区分，则以下两个分布

$$\{\langle P(W_n^1),V^*(Z_n^1)\rangle(X_n^1)\}_{n\in\mathbb{N}}\text{与}\{\langle P(Y_n^2),V^*(Z_n^2)\rangle(X_n^2)\}_{n\in\mathbb{N}}$$

也计算不可区分。

显然，强证据不可区分性隐含一般的证据不可区分性，但反之却不一定。若单项置换存在，则存在交互证明具有证据不可区分性但不具有强证据不可区分性。另一方面，强证据不可区分性也弱于零知识性。

定理 1.5.1 设$\langle P,V\rangle$是$L\in NP$的带辅助输入零知识证明，则$\langle P,V\rangle$是强证据不可区分的。

证据不可区分证明弱于零知识证明，但其可复合性优于零知识证明。前面提到，零知识证明的并行复合可能不再具有零知识性，但证据不可区分证明并行复合后一定是证据不可区分的。设$L\in NP$，对应的NP关系记为R_L，$\langle P,V\rangle$是L证据不可区分证明。设$Q(\cdot)$为任意多项式，用P_Q表示与Q个验证者（并不要求独立，记为V_Q）并行独立地运行$m=Q(n)$个$\langle P,V\rangle$的证明者，其中n为安全参数。复合证明者P_Q的辅助输入为$\overline{w}=(w_1,\cdots,w_m)$，其中$w_i$为第$i$个证明者的辅助输入，而共同输入为$x_i$。定义并行复合NP关系

$$R_L^Q=\{(\overline{x}=(x_1,\cdots,x_m),\overline{w}=(w_1,\cdots,w_m)):\forall i,R_L(x_i,w_i)=1\}$$

定理 1.5.2 若$\langle P,V\rangle$是$L\in NP$的证据不可区分的（证据独立的）证明，对任意多项式$Q(\cdot)$，如上构造关于R_L^Q的并行复合证明系统$\langle P_Q,V_Q\rangle$也是证据不可区分的（证据独立的）。

证明：设$\langle P,V\rangle$是证据不可区分的，而$\langle P_Q,V_Q\rangle$是证据可区分的，即存在V_Q^*，及$\overline{x},\overline{w^1},\overline{w^2}$，满足

$$R_L^Q(\overline{x},\overline{w}^1)=R_L^Q(\overline{x},\overline{w}^2)=1$$

但分布$\{\langle P_Q(\overline{w}^1),V^*(z)\rangle(\overline{x})\}_{\overline{x}\in\overline{L},z\in\{0,1\}^*}$与$\{\langle P(\overline{w}^2),V^*(z)\rangle(\overline{x})\}_{x\in\overline{L},z\in\{0,1\}^*}$是计算可区分的，这里$\overline{L}=\{\overline{x},\forall i,x_i\in L\}$。

首先，构造$\overline{x}$的混合证据序列，即令

$$\overline{w}^i=(w_1^1,\cdots,w_i^1,w_{i+1}^2,\cdots,w_m^2),\ i=0,\cdots,m$$

显然，$\overline{u}^0 = \overline{w}^1, \overline{u}^m = \overline{w}^2$。那么，根据假设，一定存在$0 \le i \le m-1$，使得分布$\{\langle P_Q(\overline{u}^i), V_Q^*(z)\rangle(\overline{x})\}$与$\{\langle P_Q(\overline{u}^{i+1}), V_Q^*(z)\rangle(\overline{x})\}$是计算可区分的（为简便，略掉了分布的下标($x \in \overline{L}, z \in \{0,1\}^*$)）。由此就可构造$\langle P, V\rangle$的验证者$V^*$，使得$\{\langle P(w_{i+1}^1), V^*(z)\rangle(x_{i+1})\}$与$\{\langle P(w_{i+1}^2), V^*(z)\rangle(x_{i+1})\}$是可区分的。这与对$\langle P, V\rangle$的假设矛盾，因此$\langle P_Q, V_Q\rangle$也是证据不可区分的。

证据独立的情形可类似证明。

HC问题的3-轮零知识证明系统（错误概率为1/2，见2.4节）是证据不可区分的，根据上面的复合定理，并行运行协议3.7.5n次，则得到错误概率为2^{-n}的证据不可区分证明，因此有如下定理：

定理 1.5.3 若单向函数存在，则任意*NP*问题都有错误概率可忽略的常数轮证据不可区分证明系统。

1.6 Σ协议

1.6.1 定义

Σ协议是证明者P与验证者V之间的三轮交互证明协议。

定义 1.6.1 设R为*NP*关系，关于R的Σ协议$\langle P, V\rangle$是一个三轮公开投币（*Public coin*）交互证明协议，具有如下形式：

公共输入：$x \in L_R$。

证明者辅助输入：证据w，$R(x, w) = 1$。

– 证明者P计算承诺，$a \leftarrow \mathcal{P}(x, w; r_P)$，并将其发送给验证者，其中$r_P$为随机输入；

– 验证者随机选择挑战$c \in \mathcal{C}$（这里$\mathcal{C}$称为挑战空间），并将c返回给证明者。

– 收到挑战后，证明者计算回复消息$z = P(x, w, c, r_P)$，并发送z给验证者。

– 最后，验证者验证用某个可计算判定函数$\phi(\cdot,\cdot)$决定接受与否。若$\varphi(x, a, c, z) = 1$，输出“*accept*”（接受）；否则输出“*reject*”（拒绝）。

并满足：

(1) **完备性：** 若双方遵守协议，则验证者总接受；

(2) **知识合理性**（*knowledge soundness*）：对任意$x \in L_R$，若有两个使V接受的会话（证明）$(a, c, z), (a, c', z')$，则可以有效地求出w满足$(x, w) \in R$；

(3) **诚实验证者零知识**（*honest-verifier zero-knowledge*）：存在*PPT*模拟器$\mathcal{S}$，对$c \leftarrow_R \mathcal{C}$，$(a', z') \leftarrow S(x, c)$，使得$(a', c, z')$ 与协议$\langle P(w), V\rangle(x)$交互消息$(a, c, z)$ 不可区分。

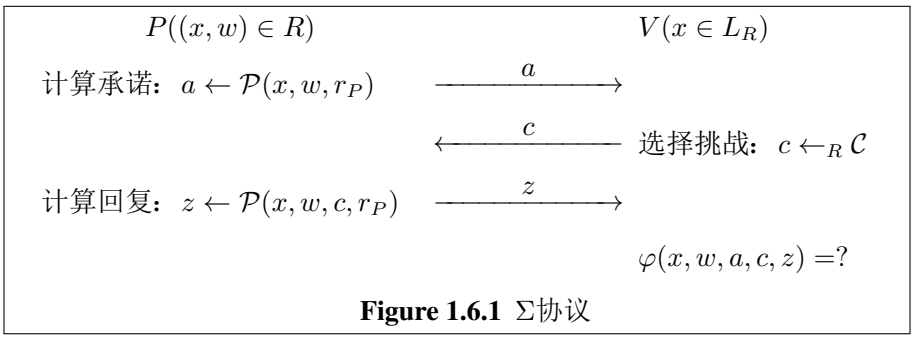


Figure 1.6.1 Σ协议

定义中的模拟器$\mathcal{S}$需要以验证者的挑战c作为辅助输入，也就是要求验证者诚实地随机选择其挑战，故称为诚实验证者零知识。

1.6.2　Σ协议

Okamoto协议： 在2.6中已说明Okamoto协议是一个证据不可区分证明，实际上它还是一个Σ协议。Okamoto协议针对的关系为：

$$R=\{(h,w_1,w_2):\exists w_1,w_2\in\mathbb{Z}_n,h=g_1^{w_1}g_2^{w_2}\}$$

(1) **完备性：**

(2) **知识合理性：** 对任意$h\in L_R$，若对任意的a，有两个使V接受的会话（证明）$(a,c,(z_1,z_2)),(a,c',(z_1',z_2'))$，其中$c\neq c'$，即

$$g_1^{z_1}g_2^{z_2}=ah^c,g_1^{z_1'}g_2^{z_2'}=ah^{c'},\ c\neq c'$$

由此可得出$h=g_1^{\frac{z_1-z_1'}{c-c'}}g_2^{\frac{z_2-z_2'}{c-c'}}$，从而得到一个证据$w_1=\frac{z_1-z_1'}{c-c'},w_2=\frac{z_2-z_2'}{c-c'}$。

(3) **诚实验证者零知识**（honest-verifier zero-knowledge）：对任意挑战$c\leftarrow\mathbb{Z}_n$，定义PPT模拟器$\mathcal{S}(h,c)$如下：

$\mathcal{S}(h,c)$：随机选择$z_1,z_2\in\mathbb{Z}_n$，计算$a=g_1^{z_1}g_2^{z_2}/h^c$，输出$(a,c,(z_1,z_2))$。

显然，模拟器的输出与真实交互时的会话具有相同的分布。

Chaum-Pedersen协议： 设G为q（素数）阶循环群，$n=|q|$（q的长度，安全参数），g为G的生成元。若$c=ab$，(g,g^a,g^b,g^c)称为Diffie-Hellman四元组。2.2 节给出了Diffie-Hellman四元组的零知识证明，因为协议中验证者的挑战$c\leftarrow_R\{0,1\}$只有两种选择，当$c\neq ab$时，证明者不可能同时正确回复两个挑战，因此协议的错误概率为1/2。证明协议的零知识性时，模拟器需要猜测验证者的挑战c，由于$c\leftarrow\{0,1\}$，所以模拟器成功的概率为1/2。为降低错误概率，可以扩大挑战空间，即令$c\leftarrow\mathbb{Z}_q$，则得到如下一般协议：

协议 1.6.1

公共输入：$x=(g,h,y_1,y_2)$；证明者拥有辅助输入：α，满足$y_1=g^{\alpha},y_2=h^{\alpha}$。

- P随机选择$u\in\mathbf{Z}_q^*$，计算$A=g^u,B=h^u$，并将A,B发送给V。
- V随机选择$c\in\mathbb{Z}_q$，并将σ提交给P。
- P计算$z=c\cdot\alpha+u\ mod\ q$，并将z返回给V。
- V验证$g^z=Ay_1^c,h^z=By_2^c$。若验证通过，接受证明，否则拒绝。

$Prover(x,\alpha)$		$Verifier(x)$
$u\in_R\mathbb{Z}_n,A=g^u,B=h^u$	$\xrightarrow{A,B}$	
	$\xleftarrow{c}$	$c\in_R\mathbb{Z}_n$
$z=u+c\alpha$	$\xrightarrow{z}$	验证$g^z=Ay_1^c,h^z=By_2^c$

协议1.6.1

扩大挑战空间是为降低错误概率。若错误概率降低至可忽略，则模拟器随机猜测c的的成功概率也会是可忽略的，所以协议不再具有零知识性。但是，协议仍然是诚实验证者零知识的，也就是说协议是Σ协议。

知识合理性： 若对确定的A,B，存在$(c_1,z_1),(c_2,z_2)$，满足$c_1\neq c_2$，且$V(x,(A,B),c_1,z_1)=V(x,(A,B),c_2,z_2)=1$，则

$$g^{z_1}=Ay_1^{c_1},h^{z_1}=By_2^{c_1},\ g^{z_2}=Ay_1^{c_2},h^{z_2}=By_2^{c_2}$$

从而可得$y_1=g^{\frac{z_1-z_2}{c_1-c_2}},y_2=h^{\frac{z_1-z_2}{c_1-c_2}}$。

诚实验证者零知识： 对任意随机挑战$c\leftarrow\mathbb{Z}_n$，定义PPT模拟器$\mathcal{S}(x,c)$如下：

$\mathcal{S}(x,c)$：随机选择$z\in\mathbb{Z}_n$，计算$A=g^z/y_1^c,B=h^z/y_2^c$，输出$((A,B),c,z)$。

1.6.3 复合关系的Σ协议

Σ协议具有很好的复合性质，由简单命题的Σ协议很容易地构造出一些复合命题的Σ协议，比较简单地有“AND”与“OR”的复合。设有关系R_1和R_2，定义关系R:

$$R=R_1\wedge R_2=\{((x_1,x_2),(w_1,w_2)):(x_1,w_1)\in R_1,(x_2,w_2)\in R_2\}$$

关系R的一个Σ协议是关系R_1的Σ协议与关系R_2的Σ协议的并行，而且两个Σ协议使用同一个(验证者的）挑战。协议1.6.1实际就是一个关于离散对数的Σ协议的“AND”复合，多个关系的“AND”复合类类似得到。

定义关系

$$R'=R_1\vee R_2=\{((x_1,x_2),(w_1,w_2)):(x_1,w_1)\in R_1\ or\ (x_2,w_2)\in R_2\}$$

关系R的Σ协议也可由关系R_1的Σ协议与关系R_2的Σ协议的并行得到，与“AND”复合不同的是对挑战处理方式不同。

对给定的实例(x_1, x_2)，不妨设存在w_1，使$(x_1, w_1) \in R_1$成立。协议如下：

- 首先，证明者首先按照关系R_1的Σ协议中证明者生成a_1，再随机选择挑战c_2，用R_2的Σ协议的模拟器生成a_2。
- 其次，验证者发送挑战c。
- 再次，证明者收到c后，计算$c_1 = c - c_2$，然后分别以c_1, c_2作为挑战计算z_1, z_2，并发送个验证者。
- 最后，验证者验证$c = c_1 + c_2$，并且(a_1, c_1, z_1)与(a_2, c_2, z_2)分别为（关系R_1与R_2的Σ协议）可接受证明。。

不难证明上面的协议是一个Σ协议。同样，多个关系的“OR”的Σ协议也很容易得到。

更一般地，可以考虑特殊线性关系情形。设G为q（素数）阶循环群，$n = |q|$（安全参数），g为G的生成元。对给定$g_{i,j}, h_i \in G$，$i = 1, \cdots, m, j = 1, \cdots, k$，定义函数（布尔公式）$\varphi : \mathbb{Z}_q^k \to \{0, 1\}$为：

$$\varphi(x_1, \cdots, x_k) = \left(\left(h_1 = \prod_{j=1}^{k} g_{1,j}^{x_j} \right) \wedge \cdots \wedge \left(h_m = \prod_{j=1}^{k} g_{m,j}^{x_j} \right) \right), (x_1, \cdots, x_k) \in \mathbb{Z}_q^k$$

设Φ为所有布尔公式集合，由此再定义二元关系R：

$$R = \left\{ (\varphi, (x_1, \cdots, x_k)) \in \Phi \times \mathbb{Z}_q^k : \varphi(x_1, \cdots, x_k) = 1 \right\}$$

对应的语言类

$$L_R = \left\{ \varphi \in \Phi : \exists (x_1, \cdots, x_k)) \in \mathbb{Z}_q^k, \varphi(x_1, \cdots, x_k) = 1 \right\}$$

显然关系R是m个形如$h_i = \prod_{j=1}^{k} g_{i,j}^{x_j}$的命题的“AND”复合，而证明单个$h_i = \prod_{j=1}^{k} g_{i,j}^{x_j}$的$\Sigma$协议可仿照证明离散对数的$\Sigma$ 协议的得到，由此可得R（或L_R)的Σ协议如下：

协议 1.6.2

公共输入：布尔公式φ；证明者拥有辅助输入：$(\alpha_1, \cdots, \alpha_k)) \in \mathbb{Z}_q^k$，满足$\varphi(\alpha_1, \cdots, \alpha_k) = 1$。

- P随机选择$u_1, \cdots, u_k \in \mathbf{Z}_q$，计算$A_i = \prod_{j=1}^{k} g_{i,j}^{u_j}$，$i = 1, \cdots, m$，并将$A_1, \cdots, A_m$发送给$V$。
- V随机选择$c \in \mathbb{Z}_q$，并将c提交给P。
- P计算$z_j = c \cdot \alpha_j + u_j \bmod q$，$j = 1, \cdots, k$，并将$z_1, \cdots, z_k$返回给$V$。
- V验证$\prod_{j=1}^{k} g_{i,j}^{z_j} = A_i h_i^c, i = 1, \cdots, m$。若验证通过，接受证明，否则拒绝。

$Prover(x,\alpha)$		$Verifier(x)$
$u_1,\cdots,u_k \leftarrow_R \mathbf{Z}_q$ $A_i = \prod_{j=1}^{k} g_{i,j}^{u_j}, i=1,\cdots,m$	$\xrightarrow{A_1,\cdots,A_m}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_n$
$z_j = c\cdot\alpha_j + u_j \ mod\ q,$ $j=1,\cdots,k$	$\xrightarrow{z_1,\cdots,z_k}$	验证 $\prod_{j=1}^{k} g_{i,j}^{z_j} = A_i h_i^c,$ $i=1,\cdots,m$

协议1.6.2

定理 1.6.1 协议*1.6.2*是关于L_R的Σ协议

证明： 略。

1.6.4 Σ协议与承诺方案

由Σ协议可以构造承诺方案。设R为NP关系，对应的语言类L_R。$Gen(.)$为R的有效抽样算法，即$(x,w) \leftarrow Gen(1^n)$，$R(x,w)=1$。称关系$R$是困难的，如果对任意PPT算法$A$，

$$\Pr\left[\begin{array}{ll} R(x,w)=1: & (x,*) \leftarrow Gen(1^n) \\ & w \leftarrow A(x) \end{array}\right]$$

是可忽略的。例如，设G为q（素数）阶循环群，g为生成元。令$R=\{(x,w): x=g^w, w\in\mathbb{Z}_q\}$，则在DL假设下，$R$是一个困难NP关系。

设$\langle P,V\rangle$ 是关于R（或L_R）的Σ协议。另外，假设对任意x，判定x是否满足$x\in L_R$是容易的。构造承诺方案如下：

协议 1.6.3

承诺者欲对$m\in\{0,1\}^\ell$进行承诺，双方共享安全参数n。

承诺阶段：

- 接收者运行$(x,w)\leftarrow Gen(1^n)$，并将x发送给承诺者。
- 承诺者验证$x\in L_R$，若验证未通过，终止协议；否则，运行Σ协议的模拟算法$(a,z)\leftarrow S(x,m)$（以m作为挑战），然后将a（对m的承诺）发送给接收者。

公开阶段：承诺者发送(m,z)，接收者接受当且仅当$V(x,a,m,z)=1$。

定理 1.6.2 协议*1.6.3*是统计隐藏计算绑定的承诺方案。

1.7　知识的零知识证明

设$R=\{(x,w)\}\subseteq\{0,1\}^*\times\{0,1\}^*$NP关系，令$R(x)=\{w:(x,w)\in R\}$，若存在多项式$p$，使得$|w|<p(|x|)$，则称$R$多项式有界。NP问题的零知识证明是对$L_R=\{x:\exists\, w\ such\ that\ (x,w)\in R\}$的判定性命题的证明，这只表明证据的存在性，并不意味着交互图灵机P（证明者）“已经拥有”对应的证据（知识）。图灵机P要证明拥有命题的证据（知识），就需说明它具有相应的计算能力。当然，最简单地是要求证明者直接输出证据，但这并不是所期望，需要的是（零知识）证明本身隐含这种能力。

1.7.1　知识的证明

定义 1.7.1 设关系$R=\{(x,w)\}$多项式有界，$\kappa:\mathbb{N}\to[0,1]$。称交互机$V$是关系$R$的具有知识错误概率（*Knowledge error*）为κ的知识验证者（*knowledge verifier*），如果:

- *Non-triviality:* 存在交互机P，使得对任意的$(x,y)\in R$，所有可能与$P(x,y)$（具有公共输入x与辅助输入y的P）交互的V（具有公共输入x）都会接受。
- *Validity (with error κ):* 存在一个多项式$q(\cdot)$和一个概率*oracle*机$\mathcal{K}$，使得对任意的P，$x\in L_R$及$y,r,\in\{0,1\}^*$，$\mathcal{K}$满足下面的条件：

  设P与V的公共输入为x，证明者P的辅助输入与随机输入分别为y,r，记为$P_{x,y,r}$，令$p(x,y,r)$为当V与$P_{x,y,r}$交互时的接受概率。若$p(x,y,r)>\kappa(|x|)$，那么$\mathcal{K}^{P_{x,y,r}}(x)$输出解$w$的期望步数的上界为：

  $$\frac{q(|x|)}{(p(x,y,r)-\kappa(|x|))}$$

  $\mathcal{K}$称为通用知识抽取器。

若$\kappa=0$，则称V是关于R的知识验证者。若P满足*Non-triviality*性，而V是知识验证者，则称$\langle P,V\rangle$是关系R的知识证明系统。

直观上，Validity要求若$p(x,y,r)>\kappa(|x|)$，知识抽取器$\mathcal{K}$通过访问$P_{x,y,r}$便可输出证据，而且以此表明$P_{x,y,r}$拥有证据，而$\mathcal{K}$是否可有效输出证据，取决于$p(x,y,r)-\kappa(|x|)$的大小。Validity还有另外一种等价的表述，即如下的定义：

定义 1.7.2 设$V,P_{x,y,r},p(x,y,r)$意义同定义*1.7.1*。称V满足带错误κ的有效性，如果存在一个概率*oracle*机$\mathcal{K}$和一个正多项式$q(\cdot)$，使得$\mathcal{K}^{P_{x,y,r}}(x)$在期望多项式时间内输出一个$s\in R(x)$的概率至少为：

$$\frac{p(x,y,r)-\kappa(|x|)}{q(|x|)}$$

定理 1.7.1 设R是一个NP关系，V是一个交互机。V对关系R满足定义*1.7.1*的有效性（带错误κ）当且仅当V对关系R满足定义*1.7.2*的有效性（带错误κ）。

证明：设V满足定义1.7.2，$\mathcal{K}, q$分别为满足要求的知识抽取器和多项式。定义新的知识抽取器$\mathcal{K}'$为重复运行$\mathcal{K}$直至成功输出一个解。由于$\mathcal{K}$成功输出的概率不小于$(p(x,y,r)-\kappa(|x|))/q(|x|)$，因此$\mathcal{K}'$重复运行$K$的期望次数为$q(|x|)/(p(x,y,r)-\kappa(|x|))$。

设V满足定义1.7.2，$\mathcal{K}, q$分别为满足要求的知识抽取器和多项式。设对任意的$(x,w)\in R$满足$|w|\leq p_1(|x|)$。定义新的知识抽取器$\mathcal{K}'$ 为；

(1) for $i=1,\cdots,p_1(|x|)$:

- 完成$2^{i+1}\cdot q(|x|)$时间内的$K^{P_{x,y,r}}(x)$的运行，若得到证据w，输出w并终止；否则以$1/2$的概率进入下一轮。

(2) 若(1)未找到证据，则穷尽搜索获得一个证据。

那么其期望运行时间为：

$$\begin{aligned}&\sum_{i=1}^{p_1(|x|)} 2^{-(i-1)}\cdot(2^{i+1}\cdot q(|x|))+2^{-P_1(|x|)}\cdot(2^{P_1(|x|)}\cdot poly(|x|))\\ =\;& 4p_1(|x|)\cdot q(|x|)+poly(|x|)\end{aligned}$$

记$\mathcal{K}^{P_{x,y,r}}$的期望时间为$E(K^{P_{x,y,r}})$，则$K^{P_{x,y,r}}$运行时间为$2E(K^{P_{x,y,r}})$的概率不大于$1/2$，即$K^{P_{x,y,r}}$在$2E(K^{P_{x,y,r}})$找到解的概率不小于$1/2$，而在第$i=-\log_2(p(x,y,r)-\kappa(|x|))$轮，共运行

$$2^{i+1}\cdot q(|x|)=\frac{2q(|x|)}{p(x,y,r)-\kappa(|x|)}$$

于是在第i轮找到解的概率不小于

$$2^{-1}\cdot 2^{-(i-1)}=p(x,y,r)-\kappa(|x|)$$

因此，知识抽取器$\mathcal{K}'$满足定义1.7.1（取$q\equiv 1$）。

例. 图同构证明协议1.2.1是知识的证明。

事实上，在协议1.2.1中，双方共同输入为(G_1,G_2)，证明者的辅助输入为G_1,G_2之间的同构映射φ，即满足$G_1=\varphi(G_2)$的置换φ。根据协议，定义1.7.1的第一个条件显然满足。另一方面，由于验证者随机选择$\sigma\in\{1,2\}$，所以被证明者欺骗的概率为1/2，因此，当验证者接受的概率大于1/2时，G_1,G_2 一定是同构的，现在需要解决的是知识抽取器$\mathcal{K}$如何才能获取G_1,G_2之间的同构映射φ。设想在证明者给出G后，验证者可以询问证明者两次，那么，第一次验证者取$\sigma=1$，获得证明者的回复ψ；第二次验证者取$\sigma=2$，获得证明者的回复ψ'，则必有$\psi(G_1)=\psi'(G_2)$，即$G_1=\psi^{-1}\circ\psi'(G_2)$，从而可求得$G_1,G_2$ 之间的一个同构映射$\psi^{-1}\circ\psi'$。由此可构造知识抽取器$\mathcal{K}$如下：

1) 启动与充当Oracle的P之间的协议，并获取P的回复的第一个消息G。

2) 随机选择$\sigma\in\{1,2\}$，以此次询问P并收到回复ψ；若$\psi=\perp$（P终止），重复此过程。记最后得到正确回复的挑战与回复为(σ,ψ)。

3) 用$\sigma' \in \{1,2\}$且$\sigma' \neq \sigma$询问P并收到回复ψ'，若$\psi \neq \perp$，重复此过程。记最后得到正确回复的挑战与回复为(σ', ψ')。

4) 由φ', ψ计算同构映射。不妨设$\sigma = 1, \sigma' = 2$，则同构映射$\varphi = \psi^{-1} \circ \psi'$。

由于V接受的概率$p > 1/2$，所以第二步获得正确的ψ的期望次数是2。在第3)步，回复ψ'正确的概率大于$2(p-1/2)$，因此3)运行的希望次数不超过$(p-1/2)^{-1}$。

总之，如上的知识抽取算法满足定义1.7.1的要求，即图同构证明协议1.2.1是知识的证明。

1.7.2　顺序复合降低知识错误概率

在一些简单的知识证明协议中，知识错误κ是可靠性的错误概率，一般会是常数。为降低证明的知识错误概率κ，需要重复运行协议。

定理 1.7.2 设R是一个多项式有界的关系，$t : \mathbb{N} \to \mathbb{N}$多项式有界。若$(P,V)$是关系$R$的知识错误为$\kappa$的知识证明系统，那么由顺序运行$(P,V)$证明系统$t(|x|)$次构成了关系$R$的知识错误为$\kappa'(n) = \kappa(n)^{t(n)}$的知识证明系统。

证明： 证明顺序复合协议显然是一个证明系统，故只需证明其满足定义1.7.2的Validity条件，即存在多项式q，使$\mathcal{K}$在期望多项式时间内成功的概率为$\frac{p'(x,y,r)-\kappa'}{q}$，其中$p'(x,y,r)$为$V'$的接受概率（以下简记为$p'$）。

利用$\mathcal{K}$，定义复合协议$\langle P', V'\rangle$的知识抽取器$\mathcal{K}'$如下：

– 选择$i \in \{1, \cdots, t\}$，并按照P'运行前$i-1$个证明$\langle P, V\rangle$。

– 设前$i-1$个证明协议的运行记录为α。运行第i个证明$\langle P, V\rangle$的知识抽取器$\mathcal{K}$。

– 输出$\mathcal{K}$输出。

下面分析$\mathcal{K}$的成功概率。设在第i次证明中，验证者V的接受概率为$c_i(\alpha)$（可能与α相关），其期望值记为c_i，根据定理1.7.1 的证明知，$\mathcal{K}$抽取成功的概率为$c_i - \kappa$。因此，则$\mathcal{K}'$ 在完成前$i-1$ 轮证明后，利用$\mathcal{K}$ 抽取成功的概率为$c_i - \kappa$。

设V在前$i-1$次的运行中接受证明的概率为a_{i-1}，则显然有，$a_i = a_{i-1}c_i$。设$\mathcal{K}'$成功的概率为p，那么，对任意$1 \leq i \leq t$，有$p \geq \frac{1}{t}a_{i-1}(c_i - \kappa)$，即

$$p = \Pr[R(x,w) = 1 : w \leftarrow \mathcal{K}'(\cdot)] \geq \frac{1}{t}a_{i-1}(c_i - \kappa), i = 1, \cdots, t$$

分以下两种情形：

1) 若$a_1 = c_1 \geq \kappa + \frac{1}{t}(p' - \kappa')$，则$p' \geq \frac{1}{t^2}(p' - \kappa')$，从而定理得证。

2) 若$a_1 = c_1 < \kappa + \frac{1}{t}(p' - \kappa')$，考虑函数$f(i) = \kappa^i + \frac{i}{t}(p' - \kappa') - a_i$，由于$f(1) > 0, f(t) = p' - a_t = 0$，故存在$2 \leq i \leq t$，满足$f(i-1) \geq 0, f(i) \leq 0$，即

$$a_i \geq \kappa^i + \frac{i}{t}(p' - \kappa'),\ a_{i-1} \leq \kappa^{i-1} + \frac{i-1}{t}(p' - \kappa')$$

从而就有$\mathcal{K}'$在第i次成功抽取的概率

$$\begin{aligned} a_{i-1}(c_i-\kappa) &= a_i - a_{i-1}\kappa \\ &> \kappa^i + \tfrac{i}{t}(p'-\kappa') - \left(\kappa^i + \tfrac{i-1}{t}(p'-\kappa')\kappa\right) \\ &\geq \tfrac{p'-\kappa'}{t} \end{aligned}$$

从而得知定理成立。

1.7.3 NP问题的知识的零知识证明

若零知识证明$\langle P,V\rangle$同时也是知识的证明，则称为知识的零知识证明。不难证明，前面给出的关于HC问题与G3C问题的零知识证明都是知识的证明，从而由对HC问题（或G3C问题）的零知识证明构造的任意NP问题的证明也将是知识的证明，因此，有如下定理：

定理 1.7.3 若具有（非一致）单向函数的存在，则任意NP问题都存在带辅助输入的知识的零知识证明系统。而且，进一步还可要求其可靠性错误概率（知识错误概率）是可忽略的。

1.7.4 知识的强证明

知识证明要求在$p(x,y,r)>\kappa$的条件下，存在知识抽取器在期望时间$1/\mathcal{O}(p(x,y,r)-\kappa)$（可以是无效的）输出证据，或者一个有效的（期望多项式时间）知识抽取器以$\mathcal{O}(p(x,y,r)-\kappa)$的概率输出证据。知识的强证明要求存在PPT的知识抽取器以接近1的概率输出证据。

定义 1.7.3 称交互函数V是关系R的强知识验证者（*knowledge verifier*），如果:

- *Non-triviality*：同定义*1.7.1*。
- *Strong Validity:* 存在可忽略函数$\mu:\mathbb{N}\to[0,1]$和*PPT*的*oracle*机K，使得对任意的P及$x,y,r,\in\{0,1\}^*$，K满足下面的条件：

  令$p(x,y,r)$与$P_{x,y,r}$的意义同定义*1.7.1*。若$p(x,y,r)>\mu(|x|)$，那么$K^{P_{x,y,r}}(x)$输出解$w\in R(x)$的概率不小于$1-\mu(|x|)$。K称为强知识抽取器。

若P(对V与R)满足*Non-triviality*性，V是强知识验证者，则称$\langle P,V\rangle$是关系R的强知识证明系统。

强知识抽取器的运行时间独立于证明者的成功概率。

1.7.4.1 图同构的知识的强（零知识）证明

图同构证明协议1.2.1是一个知识的证明，其知识错误（可靠性）概率为1/2，根据定理1.7.2，顺序复合n次协议1.2.1，可得到错误概率为2^{-n}的知识证明协议，而且复合协议还是一个知识的强证明。

设P^*是顺序复合n次的协议的证明者，知识抽取器$\mathcal{K}$可以以Oracle的方式访问P^*，具体过程如下：初始令$i=1$，协议运行记录$T=\emptyset$，

1) 询问P^*，获得第i个基本协议（1.2.1）的第一个消息G，即$G\leftarrow P^*(T)$。

2) 使用$\sigma = 1, 2$分别询问P^*，分别得到回复ψ_1, ψ_2。

3) 若ψ_1, ψ_2都是正确的，即ψ_σ是G与G_σ之间的同构映射，$\sigma = 1, 2$，则$G_1 = \psi_1^{-1} \circ \psi_2(G_2)$，输出$\varphi = \psi_1^{-1} \circ \psi_2$，并终止。

4) 若$i < n$而且ψ_1, ψ_2中只有一个是正确的，记为ψ_σ，则令$i := i + 1, T := T \cup \{\sigma\}$，返回1)进行下一轮；否则无输出终止。

若知识抽取器$\mathcal{K}$失败无输出，则意味着是以下两种情形之一：

– 存在$1 \leq i \leq n$，使得在第i轮运行中，ψ_1, ψ_2都不正确；

– 对任意的$1 \leq i \leq n$，在第i轮的ψ_1, ψ_2中，恰好只有一个是正确的。

若为第一种情形，则意味着V接受的概率为0；若为第二种情形，则V接受的概率恰为2^{-n}。由此表明：当V接受的概率大于2^{-n}时，$\mathcal{K}$一定成功输出一个同构映射φ。另外，$\mathcal{K}$显然是多项式时间可完成的，因此是强知识抽取器。

另一方面，由于顺序复合可保证零知识性，因此协议1.2.1的n次顺序复合是GI问题的知识的强零知识证明系统。

1.7.4.2　NP问题的知识的强（零知识）证明

HC问题的零知识证明（协议1.3.2）与协议1.2.1的结构类似，不难证明它也是一个知识的证明，而且也类似地可证明协议1.3.2的n次顺序复合可得到HC问题的知识的强零知识证明系统。由于HC问题是NPC问题，任意NP问题都可Levin归约于HC问题，由此有HC的知识的强零知识证明可得NP问题的知识的强零知识证明，即得如下定理：

定理 1.7.4 若*(*非一致，*non-uniformly)*单向函数存在，任意的NP关系都存在知识的强零知识证明系统（*zero-knowledge system for strong proofs of knowledge*）。

1.8　非交互零知识证明

非交互零知识证明由Blum、Feldman与Micali于1988提出[38]，是一般零知识证明的变形，在加密、签名等非交互场景有着广泛的应用。NP问题的非交互证明是证明者以非交互的方式证明命题，即证明者计算并发送证明，验证者通过检查确定命题真伪。如果不要求证明是“零知识”的，则非交互证明是平凡的。交互证明具有极其强大的能力，$IP = PSPACE$，而且在单向函数存在的条件下，任何可以进行交互证明的命题都可以有零知识证明。显然，在普通模型下，非交互且具有零知识性的证明只对BPP语言类存在，因为$x \in L$的证明π也就是$x \in L$的NP证据，当证明是零知识的，则说明PPT的模拟算法可求得π，也即$x \in L$是多项式时间可判定的。非交互零知识证明有两种模式，其中之一是由Blum等人提出的CRS（common reference string）模型[38]，即证明双方共享一个由可信第三方提供的公共随机串crs，证明者由x及对应的证据w，利用公共随机串crs 计算证明，验证者利用crs验证证明，另一种是由De Santis等人提出的预处理模式，即在证明之前，证明者与验证者通过一个交互过程（独立于证明的命题）生成所需要的参数。在这两种模

型下，任意NP问题都存在非交互零知识证明，但相比之下，CRS模型较弱，下面以CRS模型介绍非交互零知识证明。

1.8.1 定义

定义 1.8.1 （非交互证明,*non-interactive proof*）一对概率图灵机$\langle P,V\rangle$ 称为语言L的非交互证明（*non-interactive proof*）系统，如果V是多项式时间的，且下面两个条件满足：

– 完备性：对任意$x \in L$及证据$w : R_L(x,w) = 1$，有

$$Pr[V(x,\pi) = 1 : \pi \leftarrow P(x,w)] \geq \frac{2}{3}$$

– 合理性：对任意$x \notin L$ 、任意算法B，有

$$Pr[V(x,\pi) = 1 : \pi \leftarrow B(x)] \leq \frac{1}{3}$$

称π为$x \in L$的证明。

定义中对证明者的计算能力无限制，若要求证明者也是多项式时间的，则称为是非交互论证（non-interactive argument）。显然，对任意$L \in NP$，都存在一个平凡的非交互证明系统，x的证据w即可作为$x \in L$的证明。

设$\langle P,V\rangle$是$L \in NP$（对应的NP关系记为R_L）非交互证明系统，要求$\langle P,V\rangle$是零知识的，就是要求证明者输出的证明π不泄露额外的知识。与交互证明时的情形类似，零知识性是要求存在PPT的算法$\mathcal{S}$（模拟器），使得$\mathcal{S}(x)$输出的分布与证明者给出的证明$\pi \leftarrow P(x,w)$的分布是计算不可区分的。

标准模型下的非交互证明（满足定义1.8.1）具有零知识性只能对BBP复杂类存在。事实上，若L的非交互证明$\langle P,V\rangle$是零知识的，则存在PPT的模拟器$\mathcal{S}$，使得$\{\mathcal{S}(x)\}$与$\{\pi \leftarrow P(x,w)\}$计算不可区分，由此可构造$L$的判定算法$M$如下：

1) 对任意x，运行$\mathcal{S}(x)$；

2) 运行验证算法$V(x,S(x))$并输出结果。

由于$\{\mathcal{S}(x)\}$与真实的证明不可区分，因此当$x \in L$时，$\Pr[M(x) = 1] > \frac{2}{3} - negl(n)$；另一方面，当$x \notin L$时，$\Pr[M(x) = 1] \leq \Pr[\langle B,V\rangle(x) = 1] \leq \frac{1}{3}$。因此，$L \in BBP$。

NP问题的非交互零知识证明需要CRS模型，即证明者与验证者共享一个公共参考串（common reference string，CRS）。一般地，公共参考串crs是均匀分布的随机比特串，由可信第三方产生。CRS模型下的非交互证明的定义如下：

定义 1.8.2 （非交互证明,*non-interactive proof*）设$\mathcal{G}$为*PPT*算法，$crs \leftarrow \mathcal{G}(1^n)$（$n$安全参数）。一对概率图灵机$\langle P,V\rangle$ 称为语言L 的非交互证明（*non-interactive proof*）系统，如果V 是多项式时间的，且下面两个条件满足：

– 完备性：对任意$x \in L$及证据$w : R_L(x, w) = 1$，有

$$Pr[V(x, crs, \pi) = 1 : \pi \leftarrow P(x, crs, w)] \geq \frac{2}{3}$$

– 合理性：对任意$x \notin L$、任意算法B，有

$$Pr[V(x, crs, \pi) = 1 : \pi \leftarrow B(x, crs)] \leq \frac{1}{3}$$

称π为$x \in L$的证明。若限制P也是*PPT*的，则称为非交互论证。

定义 1.8.3 语言L的非交互证明系统（论证）$\langle P, V\rangle$称为是零知识的，如果存在*PPT*算法$\mathcal{S} = (S_1, S_2)$，使得以下两个分布

– $\{(x, crs, \pi = P(x, crs))\}_{x\in L}$;

– $\{(x, \sigma, \pi = S_2(x, \tau)) : (\sigma, \tau) \leftarrow S_1(x)\}_{x\in L}$.

计算不可区分。

非交互零知识证明记为NIZK。模拟器实际上包含两个算法，其中S_1输出替代crs的σ及辅助信息τ（陷门），S_2利用辅助信息τ输出模拟的证明。$\mathcal{S}$之所以可以给出与真实证明不可区分的证明，是由于模拟器实际上是可以对CRS进行操作（以σ替代crs）。

由可信第三方生成的crs一般地是$\{0, 1\}^{poly(n)}$上的均匀分布，因此，为简洁有时就直接将非交互证明的CRS写成U_m。

1986年，Fiat与Shamir[30]给出了由Σ协议构造签名方案的方法，Pointcheval等人对其安全性进行了证明[31]，现称为Fiat-Shamir启发式（Fiat-Shamir heuristic）。Fiat-Shamir启发式实际是将3- 轮公开投币交互证明转化为非交互证明的方法。在3轮公开投币协议交互证明中，验证者输出的挑战是公开投币结果（均匀随机分布），如果双方共享有一个随机函数，则验证者发送挑战（投币结果）可由计算共享的随机函数得到，这样证明就可转化为非交互证明。Fiat-Shamir启发式是用Hash函数作为随机函数，过程如下：设$\langle P, V\rangle$是关于L 的Σ 协议，非交互证明$\langle P', V'\rangle$。

公共输入：$x \in L_R$。

证明者辅助输入：证据w，$R(x, w) = 1$。

– 证明者P'计算承诺，$a \leftarrow P(x, w)$，并将其发送给验证者；

– 证明者P'计算$c = H(x, a)$（以此代替验证者V的随机挑战）。

– 证明者P'计算证明，$\pi \leftarrow P(x, w, c)$，发送$\pi$给验证者$V'$。

– 验证者V'验证$V(x, a, c, \pi) = 1$，若验证通过，输出“accept”（接受）；否则输出“reject”（拒绝）。

1.8.1.1 离散对数的非交互零知识证明

设G为q（素数）阶循环群，g为生成元。考虑NP关系$R = \{(x, w) : x = g^w, w \in \mathbb{Z}_q\}$的$\Sigma$协议：

协议 1.8.1

- 证明者P随机选择$u \in \mathbb{Z}_q$，计算承诺，$a = g^u$，并将其发送给验证者；
- 验证者V随机选择$c \in \mathbb{Z}_q$，并发送给证明者P。
- 证明者P计算证明，$z = u + cw \ mod \ q$，发送z给验证者V。
- 验证者V验证$g^z = ax^c$，若验证通过，输出“accept”（接受）；否则输出“reject”（拒绝）。

协议使证明者向验证者证明自己拥有x（公钥）的离散对数x（私钥），采用Fiat-Shamir方法，可将其转化为非交互的证明。假设双方共享的CRS为Hash函数$H : \{0,1\}^* \to \{0,1\}$。协议如下：

协议 1.8.2

公共输入：$x = g^w$及安全参数n。

证明者辅助输入：证据w。

- 证明者随机选择$u \in \mathbb{Z}_q$，计算$a = g^u$、$c = H(x, a)$与$z = u + cw \ mod \ q$，最后将(a, z)发送给验证者。
- 验证者计算$c = H(x, a)$，并验证$g^z = ax^c$。

定理 1.8.1 协议*1.8.2*在*RO*模型下是离散对数的非交互零知识证明。

1.8.2 从隐藏比特证明到非交互证明

为给出NP问题的一般的非交互零知识证明，先引入称为隐藏比特证明（hidden-bits proof）的非交互证明，然后给出NP问题的隐藏比特证明系统，最后再将其转化为非交互零知识证明。

定义 1.8.4 一对图灵机$\langle P, V\rangle$称为是语言L的隐藏比特证明（*hidden-bits proof*），如果V是多项式时间的，且下面两个条件成立：

- *Completeness:*对任意$x \in L$，有

$$Pr[V(x, R_I, I, \pi)) = 1 : (I, \pi) \leftarrow P(x, R)] \geq \frac{2}{3}$$

其中R是$\{0,1\}^{poly(|x|)}$上的均匀分布，R_I是R中下标在$I \subseteq \{1, 2, \cdots, poly(|x|)\}$的比特组成的子串。

– *Soundness:* 对任意$x \notin L$即任意B，有

$$Pr[V(x, R_I, I, \pi)) = 1 : (I, \pi) \leftarrow B(x, R)] \leq \frac{1}{3}$$

其中R与R_I意义同前。

在隐藏比特证明中，R类似于非交互证明中的crs，由可信第三方生成，所不同的是证明者可以阅读全部的R，而验证者只能看到验证者公开的R_I。尽管隐藏比特证明与标准的非交互证明不同，但其零知识性可类似定义。为证明NP问题有隐藏比特的证明，只需对某个NPC问题（HC 问题）进行构造。

协议 1.8.3 *HC*问题的隐藏比特证明

Common input: 有向图$G = (V, E)$，记$n = |V|$。设G中存在Hamilton圈$C \subseteq G$。
隐藏的比特串R: $n^3 \times n^3$的Boolean矩阵$M_{n^3 \times n^3} = \{m_{i,j}\}$，满足$Pr[m_{i,j} = 1] = n^{-5}$。
说明：若M包含一个$n \times n$的Hamilton子矩阵而且其余$n^6 - n^2$个元素全为0，则称M是有用的。

证明者完成：

– Case1：M是有用的：令$H = \{h_{i_k, j_s}\}_{k,s=1}^{n}$为$M$中$n \times n$的Hamilton子矩阵，$C_H$为$H$中的Hamilton回路。

  – 获得从V到H的行与列的1-1映射：

  $$\pi_1 : \{1, \cdots, n\} \to \{i_1, \cdots, i_n\}, \pi_2 : \{1, \cdots, n\} \to \{j_1, \cdots, j_n\}$$

  使得$(u, v) \in C \Leftrightarrow h_{\pi_1(u), \pi_2(v)} = 1$；

  – 公开M的不在H中的所有$n^6 - n^2$个元素和H中不与G的边对应的$n^2 - |E|$个元素；（公开的全是0）

  – 同时输出(π_1, π_2)。

– Case2：M不是有用的：公开M的所有元素；

验证者验证：

– 若证明者未公开M的所有元素，验证M中除与$\{(\pi_1(u), \pi_2(v)) : (u, v) \in E\}$对应元素外，其余元素全已公开且均为0。

– 若证明者公开了M，验证M是无用的。

定理 1.8.2 协议*1.8.4*是*HC*问题存在零知识的*Hidden-bits*证明系统，而且若证明者以*Hamilton*圈为辅助输入，则它还是*PPT*的。

证明：若证明者获得G的Hamilton圈，则其显然可在多项式时间完成协议。

(1) Completeness: （显然成立）

(2) Soundness: 假设$G \notin HC$。由于$Pr[M$是有用的$] = \Omega(n^{-3/2})$，故只需在此条件下进行证明。

- 若证明者按照Case1进行证明：
  - 若$\pi_1(V)\times\pi_2(V)\neq H$，则证明者一定公开了$H$某行或某列，而其一定不全为0，从而知验证者必拒绝；
  - 若$\pi_1(V)\times\pi_2(V)=H$，则G中与$h_{\pi_1(u),\pi_2(v)}=1$对应的边(u,v)存在，从而可知G中存在Hamilton圈，这与假设矛盾。
- 若证明者按照Case2进行证明，验证者一定拒绝。

(3) Zero-knowledge: 构造模拟算法$\mathcal{S}(G)$如下：

- 随机选择矩阵$M_{n^3\times n^3}$，满足$Pr[m_{i,j}=1]=n^{-5}$。
- 若M是无用的，输出(G,M)；若M是有用的，均匀随机选择1-1映射$\pi_1,\pi_2:\{1,\cdots,n\}\to\{1,\cdots,n^3\}$，记
$$I=\{1,\cdots,n^3\}-\{(\pi_1(u),\pi_2(v)):(u,v)\in G\}$$
输出$(G,0^{n^6-|E|},I,(\pi_1,\pi_2))$。

显然，$\mathcal{S}(G)$的输出与交互证明的输出无区别。

由于HC问题是一个NPC问题，因此任意NP问题都存在零知识的隐藏比特证明系统。下面将$L\in NP$的隐藏比特证明转化为标准的非交互证明，而且保持零知识性不变。

设$\langle P,V\rangle$是L的Hidden-bits证明系统，可计算函数$f:\{0,1\}^*\to\{0,1\}^*$是1-1且$|f(x)|=|x|$，$b:\{0,1\}^*\to\{0,1\}$也是可有效计算的。构造L的非交互证明系统$\langle P',V'\rangle$如下：

协议 1.8.4 从隐藏比特证明到非交互证明

公共输入：$x\in L\subseteq\{0,1\}^n$;
$crs\triangleq s=(s_1,\cdots,s_m)$，$m=poly(n)$，$s_i\in\{0,1\}^n$。

证明者P'：

(1) 计算$r_i=b(f^{-1}(s_i))$，$i=1,\cdots,m$;

(2) 调用P，$(I,\pi)=P(x,R)$，$R=r_1\cdots r_m$;

(3) 输出(I,π,p_I)，其中$I=(i_1,\cdots,i_t)$，$p_I=(f^{-1}(s_{i_1}),\cdots,f^{-1}(s_{i_t}))$

验证者V'： 收到P'的证明$(I,\pi,p_I=(p_{i_1},\cdots,p_{i_t}))$后，进行如下验证：

(1) 验证$s_{i_k}=f(p_{i_k})$，$i_k\in I$，若未通过，拒绝。

(2) 计算$r_{i_k}=b(p_{i_k})$，$k=1,\cdots,t$，令$R_I=r_{i_1}\cdots r_{i_t}$。

(3) 用(x,R_I,I,π)调用验证者V，V'接受当且仅当$V(x,R_I,I,\pi)$接受。

定理 1.8.3 （*1*）若$Pr[b(U_n)=1]=\frac{1}{2}$，则以上由$\langle P,V\rangle$构造的$\langle P',V'\rangle$是L的非交互证明系统。（*2*）同时，若$\langle P,V\rangle$是零知识的且b是f的*hard-core*，则$\langle P',V'\rangle$也是零知识的。

证明：(1)由f,b的性质与s_i的均匀分布特性可得r_i是均匀分布的，因此由$\langle P,V\rangle$完备性(completeness)与合理性(soundness)可知$\langle P',V'\rangle$是L的证明系统。

（2）Zero-knowledge: 设$\langle P,V\rangle$的模拟器为$\mathcal{S}$，由$\mathcal{S}$构造$\langle P',V'\rangle$的模拟器$\mathcal{S}'$如下：

- 调用$\mathcal{S}$，即$(x,R_I,I,\pi)\leftarrow S(x)$;
- 独立选择$e_j\in_R\{0,1\}^n$，使得$b(e_j)=r_j$，计算$s_j=f(e_j)$，$j\in I=\{i_1,\cdots,i_t\}$;
- 独立选择$s_j\in_R\{0,1\}^n$，$j\notin I$;
- 输出(x,s,I,π,p_I)，其中$s=s_1\cdots s_m$，$p_I=(e_{i_1},\cdots,e_{i_t})$。

注:(1) 为保证零知识性，f必须是单向函数;

(2) 由于P'需要计算$f^{-1}(s_i)$，P'不可能有效实现。为解决此，可采用陷门置换。

定理 1.8.4 若单向置换族存在，则任意的NP语言都有*NIZK*证明系统；若加强单向陷门置换族存在，则任意的NP语言都有*NIZK*证明系统，并且以NP证据为辅助输入的证明者是多项式时间可实现的。

1.8.3　多命题非交互证明

上面的定义只针对单个命题的证明（称为单命题NIZK），更一般的情况是证明者要证明多个命题，当然，如果每次都进行独立的证明，那问题就变成了单个命题的证明。如果采用多次独立的证明，就需要多个CRS，这不是一个有效的方法，而理想的方法是利用单个CRS证明多个命题（多命题NIZK）。注意到CRS的长度与命题长度有关，为了简化问题的描述，假设所有命题的长度是一样的，比如设其为n^ε，其中$\varepsilon>0$为常数。多命题NIZK的零知识性也称为无界零知识的（unbounded zero-knowledge）。

定义 1.8.5 设$\langle P,V\rangle$是$L\in NP$的非交互证明系统，安全参数为n。称其为无界零知识的（*unbounded zero-knowledge*），如果对任意多项式p，存在*PPT* 模拟器$\mathcal{S}$，使得下面两个随机分布

- $\{((x_1,\cdots,x_{p(n)}),U_m,(P(x_1,U_m),\cdots,P(x_{p(n)},U_m)))\}_{x_1,\cdots,x_{p(n)}\in L_{n^\varepsilon}}$
- $\{M(x_1,\cdots,x_{p(n)})\}_{x_1,\cdots,x_{p(n)}\in L_{n^\varepsilon}}$

是计算不可区分的，这里$m=poly(n)$，$L_{n^\varepsilon}\stackrel{def}{=}L\cap\{0,1\}^{n^\varepsilon}$。

注意到要使隐藏比特证明具有零知识性，其随机比特串R显然是不能重复使用的，因此，由它得到的NIZK系统就不能满足上面的定义。下面给出将单命题的NIZK转化为多命题的NIZK的一种方法。

设$G:\{0,1\}^l\to\{0,1\}^{2l}$，$L_1\in NPC$。对任意的$L\in NP$，定义

$$L_2\stackrel{def}{=}\{(x,p):x\in L\vee\exists u\in\{0,1\}^{|x|}\ s\,t.\,G(u)=p\}$$

则$L_2\in NP$。采用标准归约方法可以将L_2中的任意实例$z_2\in L_2$ (设$|z_2|=3\ell$，即$|x|=\ell$)归约于L_1中的某个实例$z_1\in L_1$，$|z_1|=q(\ell)$，这里q为多项式。设$\langle P,V\rangle$是L_1的NIZK证明系统，其证明

长度为$q(\ell)$的命题需要长为$q'(\ell)$的CRS。记$m = q'(\ell) + 2\ell$。

协议 1.8.5 ***An unbounded NIZK proof system***

Common input: $x \in \{0,1\}^l$;

Common reference string: $r = (p, s) \in \{0,1\}^{2l} \times \{0,1\}^{m-2\ell}$.

证明者P'：完成如下过程：

(1) 使用标准归约方法，将$(x, p) \in \{0,1\}^{\ell+2\ell}$归约于$y \in \{0,1\}^{q(\ell)}$，同时由$x \in L$的证据$w_x$ 获得满足$y \in L_1$的证据w_{L_1}。

(2) 以s作为CRS，调用P，输出$\pi = P(y, s)$；（若P'拥有x的证据，则归约时可得y的证据w_{L_1}）

验证者V'：收到P'的证明π后，进行如下验证：

(1) 使用标准归约方法，将$(x, p) \in \{0,1\}^{\ell+2\ell}$归约于$y \in \{0,1\}^{q(\ell)}$。

(2) 以s作为CRS，y作为共同输入，调用V，输出$V(y, s, \pi)$；

定理 1.8.5 设$\langle P, V\rangle$是$L \in NP$的一般的*NIZK*证明系统，并且若有NP证据作为辅助输入，P可在多项式时间内实现。G是伪随机发生器。则如前构造的$\langle P', V'\rangle$是$L \in NP$的*Unbounded NIZK*证明系统，并且当P'以NP证据为辅助输入时，可在多项式时间内实现。

证明：首先，由标准归约方法与P的可实现性，我们容易得出，当P'拥有NP证据，则其可在多项式时间内实现。下面证明协议是Unbounded NIZK证明系统。

Completeness: 由$\langle P, V\rangle$的完备性可得。

Soundness: 由$\langle P, V\rangle$的可靠性可得。

Unbounded Zero-knowledge: 设实例序列$\overline{x} = (x_1, \cdots, x_t)$，其中$x_i \in L$，$i = 1, \cdots, t$。设$\langle P, V\rangle$对应的模拟器为$\mathcal{S}$，构造$\langle P', V'\rangle(\overline{x})$的模拟器$\mathcal{S}'(\overline{x})$如下:

(1) 随机独立选择$u \in_R \{0,1\}^l, s \in_R \{0,1\}^{n-2l}$，计算$p = G(u)$。

(2) 对实例x_i，令$z_i = (x_i, p)$，则有$z \in L_2$，将z_i归约于L_1，获得L_1中的实例y_i，同时由$z_i \in L_2$证据u获得$y_i \in L_1$中的证据$w_{L_1,i}$，$i = 1, \cdots, t$。

(3) 运行$\pi_i \leftarrow P(y_i, w_{L_1,i}, s)$，$i = 1, \cdots, t$。

(4) 输出$(\overline{x}, (p, s), \pi_1, \cdots, \pi_t)$。

假设$x_i \in L$的NP证据为w_i，$i = 1, \cdots, t$，定义算法$\overline{S}$：

运行$\mathcal{S}'(\overline{x})$，在第(2)步中以$w_i$（$z_i \in L_2$的证据）代替$u$完成第(2)步的归约，得到的$y_i \in L_1$的证据记为$w'_{L_1,i}$，然后运行$\pi_i \leftarrow P(y_i, w_{L_1,i}, s)$。其余过程与$\mathcal{S}\prime$完全相同。

由于进行从z_i到y_i的归约时，$\mathcal{S}'$与$\overline{S}$使用了不同的证据，得到$y_i \in L_1$的证据也可能不同。因此，$\mathcal{S}'(\overline{x})$与$\overline{S}(\overline{x})$的区别为二者使用了不同的证据序列使用$P$完成了证明。由于$\langle P, V\rangle$是零知识证明，

因此也是证据不可区分证明，从而就有$\mathcal{S}'(\bar{x})$的输出与$\overline{S}(\bar{x})$的输出计算不可区分，而$\overline{S}$的输出即为真实的交互证明。

定理 1.8.6 若单向陷门置换族存在，则任意的NP语言都有*unbounded NIZK*证明系统。若加强陷门置换（*enhanced trapdoor permutation*）族存在，并且证明者以NP证据为辅助输入，则任意问题都有*unbounded NIZK*证明系统，其中的证明者还是多项式时间可实现的。

非交互零知识证明有广泛的应用，如CCA安全加密方案[33]与安全签名方案[36, 37]的设计。为使加密方案具有CCA的安全性，密文可由对明文的加密与非交互证明两个部分组成，以保证密文生成的合法性。如在ElGamal加密中，m的加密为$(R, C) = (g^r, h^r \cdot m)$，为将其增强为CCA安全的方案，密文需要增加对离散对数$r = \log_g R$的非交互零知识证明。这样，对手在进行解密询问时，密文包含有它知道$r = \log R$的证明，因此它可自己解密得到m，即表明解密询问无助于对安全性的攻击。从归约证明的角度看，就是归约算法可以在不知道解密私钥的情况下回答敌手的解密询问，因此可完成将CCA安全性归约至CPA安全性。

Blum 等人[38] 基于数论的假设给出NP 问题的非交互零知识证明，而Feige 等人[39] 利用陷门置换构造了NP 问题的NIZK，后来Groth 等人[40, 41] 利用双线性映射构造了更为高效的NIZK。

1.8.4　更强的性质

非交互零知识证明在密码学中有广泛的应用，但在有些场景下，标准的合理性或零知识性并不能满足需要，于是提出了更强的定义。

1.8.4.1　Adaptive NIZK Proof

定义 1.8.6 称$\Pi = \langle P, V \rangle$为$L \in NP$的动态非交互零知识证明*(adaptive NIZK proof)*，如果P, V均为多项式时间的机器，且满足：

– Completeness*:* 对任意$x \in L$及证据$w : R_L(x, w) = 1$，有

$$Pr[V(x, crs, \pi) = 1 : \pi \leftarrow P(x, crs, w)] = 1$$

– Adaptive Soundness*:* 对任意*PPT*的$A = (A_1, A_2)$，$A_1(r) \notin L$，有

$$\Pr[V(x, r, \pi) = true : x \leftarrow A_1(r), \pi \leftarrow A_2(r)] < \mu(n)$$

即在证明者获得作为*CRS*的r后选择命题的条件下，证明也是可靠性的。

– Adaptive Zero Knowledge 存在*PPT*算法$\mathcal{S} = (S_1, S_2))$，对任意*PPT*的$A = (A_1, A_2)$，使得

$$|\Pr[Expt_A^S(n) = 1] - \Pr[Expt_A(n) = 1]| < \mu(n)$$

其中

$Expt_A^S(n)$： 1) $(r,\tau) \leftarrow S_1(1^n)$ 2) $(x,w,s) \leftarrow A_1(r)$ 3) $\pi \leftarrow S_2(x,r,\tau)$ 4) *Return* $A_2(\pi,r,s)$

$Expt_A^S(n)$： 1) $r \leftarrow \{0,1\}^{poly(n)}$ 2) $(x,w,s) \leftarrow A_1(r)$ 3) $\pi \leftarrow P(x,w,r)$ 4) *Return* $A_2(\pi,r,s)$

也就是，即使允许验证者选择命题（），证明依然是零知识的，即存在模拟器，其输出与真实证明不可区分。

1.8.4.2 Unbounded Adaptive NIZK Proof

一般的Adaptive NIZK的零知识性只保证证明单个命题的零知识性，如果希望在证明多个命题时仍然具有零知识性，就得到如下的定义：

定义 1.8.7 称$\Pi = \langle P,V\rangle$为$L \in NP$的无限动态非交互零知识证明*(unbounded adaptive NIZK proof)*，如果P,V均为多项式时间的机器，且满足：

- Completeness*:* 对任意$x \in L$及证据$w : R_L(x,w) = 1$，有

$$Pr[V(x,crs,\pi) = 1 : \pi \leftarrow P(x,crs,w)] = 1$$

- Adaptive Soundness*:* 对任意*PPT*的$A = (A_1,A_2)$，$A_1(r) \notin L$，有

$$\Pr[V(x,r,\pi) = true : x \leftarrow A_1(r), \pi \leftarrow A_2(r)] < \mu(n)$$

- Adaptive Zero Knowledge 存在*PPT*算法$\mathcal{S} = (S_1,S_2))$，对任意*PPT*的$A = (A_1,A_2)$，使得

$$\Pr[Expt_A^S(n) = 1] - \Pr[Expt_A(n) = 1]| < \mu(n)$$

其中$Expt_A^S(n)$与$Expt_A(n)$分别为如下两个试验：

$Expt_A^S(n)$： 1)$(r,\tau) \leftarrow S_1(1^n)$ 2)*Return* $A^{S_2(\cdot,r,\tau)}(r)$

$Expt_A(n)$： 1)$r \leftarrow \{0,1\}^{poly(n)}$ 2)*Return* $A^{P(\cdot,\cdot,r)}(r)$

可靠性与Adaptive NIZK Proof一样，零知识性有所加强，即对确定的CRS，允许验证者可以收到无限多个证明。

1.8.4.3 Unbounded Simulation Soundness NIZK

非交互证明的模拟可靠性就是要求即使敌手看到了（多个）模拟的证明，也不能对错误命题给出可接受的证明，形式定义如下：

定义 1.8.8 *[33]*设$\Pi = \langle P, V\rangle$为$L \in NP$的动态非交互零知识证明。称$\Pi$是无限模拟可靠的*(Unbounded Simulation Soundness NIZK)*，如果对任意*PPT* 敌手A，有

$$\Pr[Expt_{A,\Pi}(n) = true] < \mu(n)$$

其中$Expt_{A,\Pi}(n)$定义为如下试验：

(1) $(r, \tau) \leftarrow S_1(1^n)$

(2) $(x, \pi) \leftarrow A^{S_2(\cdot,r,\tau)}(r)$*:* 敌手任意选择实例并询问模拟器$S_2$获得对应的模拟的证明，$Q$为所有的询问和得到的回复。最后，敌手输出实例$x$ 和对x 的证明。

(3) 输出*true*当且仅当$(\pi \notin Q \wedge x \notin L \wedge V(x, r, \pi) = true)$.

模拟可靠的NIZK可用于CCA安全的公钥加密，其实这一概念的提出就源自于CCA2安全加密方案的设计。

在1990年，Naor与Yung给出了有由（被动安全）公钥加密方案Π'构造CCA安全的公钥加密方案Π的一种方法[32]：对明文的加密采用双加密，即明文用两个独立的密钥进行加密，得到两个密文，然后使用NIZK 证明两个密文是同一个明文的密文。方案Π的密文就由两个密文与一个非交互证明组成，其安全性归约于基本方案Π'的安全性。在归约证明时，需要将攻击Π的CCA2安全性的敌手$\mathcal{A}$转换为攻击Π被动安全性的敌手$\mathcal{A}'$，这样在构造$\mathcal{A}'$时，需要回答$\mathcal{A}$ 的询问，这需要给出密文里的非交互零知识证明（模拟证明），为使敌手$\mathcal{A}$在此情形下（看到模拟的证明）仍不能非法地产生可通过验证的密文（给出可接受的“错误”证明）。因此，需要非交互零知识具有这个性质。

文献[43]由一般NIZK构造了Unbounded Simulation Soundness NIZK。

1.8.4.4　Non-malleable NIZK

非交互证明的延展性（Malleability）是指验证者借助于证明者给出的证明，证明一个自己无法证明的命题。不可延展的（Non-malleable）证明要求：验证者借助证明者给出的证明可以证明的任何命题，验证者即使不借助证明者的证明也一定可以完成。当然，验证者简单拷贝证明者的证明的平凡情况不可避免，应当除外。为强化不可延展性的定义，可让敌手（验证者）访问模拟算法得到模拟的证明。

Sahai[33]定义了Adaptive non-malleable NIZK，针对验证者只得到单个证明时的情形，随后，[43]针对验证者看到证明者的多次证明情形，给出较强的定义。

定义 1.8.9 设$\Pi = \langle P, V\rangle$为$L \in NP$的动态无限非交互零知识证明。称$\Pi$是不可延展的（*Non-malleable*），如果存在*PPT*算法M，使得对任意*PPT*敌手A和非一致的多项式关系R，满足

$$\Pr\Pr[Expt^{S}_{A,R}(n) = true] \leq \Pr[Expt_{M^A}(n) = true] + negl(n)$$

其中$Expt^{S}_{A,R}(n)$与$Expt_{M^A}(n)$定义如下：

$$
\begin{aligned}
Expt^{S}_{A,R}(n):\ & 1)\ (r,\tau)\leftarrow S_1(1^n)\\
& 2)\ (x,\pi,aux)\leftarrow A^{S_2(\cdot,r,\tau)}(r).\ \text{令}Q\text{为对}S_2\text{的询问记录}\\
& 3)\ \textit{Return}\ true \Leftrightarrow (\pi\notin Q\wedge R(x,aux)=true\wedge V(x,\pi,r)=true)
\end{aligned}
$$

$$
\begin{aligned}
Expt_{M^A}(n):\ & 1)\ r\leftarrow\{0,1\}^{poly(n)}\\
& 2)\ (x,\pi,aux)\leftarrow M^A(1^n)\\
& 3)\ \textit{Return}\ true \Leftrightarrow (V(x,\pi,r)=true)\wedge R(x,aux)=true)
\end{aligned}
$$

Sahai在99年对Naor-Yung的CCA安全的公钥方案进行修改[33]，用Non-malleable NIZK证明代替了NIZK证明。此外，[33]还给出了由Adaptive NIZK构造不可延展Adaptive NIZK 的方法。

1.8.5 简明非交互证明

基于应用需要，人们关注NP问题的非交互证明的通信复杂度，希望非交互证明的验证比直接验证对应的NP关系更有效，由于[34, 35]已经证明了非平凡语言类不存在具有统计（完美）可靠性非交互证明，从而只能具有计算可靠性的协议，称其简短非交互论证（Succinct non-interactive argument，SNARG）。NP关系R的简短非交互论证（SNARG）一般采用较强的预处理模式，可以表示为一个非交互论证$\Pi=\langle P,V\rangle$和一个PPT的初始化算法$\mathcal{G}$ 组成，为简便有时就直接写为$\Pi=\langle P,\mathcal{G},V\rangle$。初始化算法$(crs,priv)\leftarrow\mathcal{G}(1^n)$ 生成非交互证明$\Pi=\langle P,V\rangle$的公共随机串crs和证明验证密钥$priv$。对任意$(x,w)\in R$，证明者P由x,w 及crs生成对x的证明，验证者由crs及验证密钥$priv$验证证明。如果初始化算法由验证者运行，并将参数交给证明者，协议形式上是2轮。

最早，Kilian[44] 利用抗碰撞Hash函数，构造了NP问题的4轮论证协议，验证时间只是关于$\log t$ 的多项式，其中t 为直接验证NP 关系的时间。Micali[45] 在2000年将Fiat-Shamir方法应用于Kilian协议[44]，在RO模型下得到了NP问题简明非交互论证，而标准模型下的SNARG的存在性以及如何构造随即成为人们关注的问题。Aiello等人[46]首先采用PIR（private information retrieval）与PCP证明（probabilistically checkable proofs），研究了非交互简短证据不可区分证明（succinct WI）而Dwork等人[47]说明这种PIR+PCP的方法不能保证证明者的回答的一致性，故协议可靠性不能保证。后来，Di Crescenzo 等人[48] 采用了Merkle-Tree结构，解决证明者不一致的问题，并基于一个非标准的困难性假设，给出了NP问题的简短（两轮）交互论证，而Groth在2010年基于双线性映射，构造了电路可满足性的非交互零知识证论证，其证明由常数个群元素组成，并可有效验证[50]。最近，Bitansky[49]等人修改了[48] 中的构造，证明了若可抽取抗碰撞Hash 函数（extractable collision-resistant hash function, ECRH）存在，则简短非交互论证也存在。

定义 1.8.10 称非交互论证$\Pi=\langle P,V\rangle$和一个*PPT*的初始化算法$\mathcal{G}$为关系R的简明非交互论证，记为$\Pi=\langle P,\mathcal{G},V\rangle$，如果下列条件成立：

– 完备性（*Completeness*）：对任意$(x,w)\in R$，有

$$
\Pr\left[\begin{array}{ll} V(x,crs,priv)=1: & (crs,priv)\leftarrow\mathcal{G}(1^n)\\ & \pi\leftarrow P(x,w,crs)\end{array}\right]=1
$$

– 可靠性（*Soundness*）：分为*static*与*adaptive*两种

  – *static soundness:* 对任意*PPT*算法P^*与$x \notin L_R$，有

$$\Pr\left[\begin{array}{ll} V(x,crs,priv)=1: & (crs,priv)\leftarrow\mathcal{G}(1^n) \\ & \pi\leftarrow P^*(x,w,crs) \end{array}\right] \leq negl(n)$$

  – *adaptive soundness:* 对任意*PPT*算法P^*，有

$$\Pr\left[\begin{array}{ll} V(x,crs,priv)=1: & (crs,priv)\leftarrow\mathcal{G}(1^n) \\ & (x,\pi)\leftarrow P^*(w,crs), x\notin L_R \end{array}\right] \leq negl(n)$$

– 简短性（*Succinctness*）：由诚实证明者P对$(x,w)\in L_R$给出的证明π的长度满足：

$$|\pi| = poly(n)(|x|+|w|)^{o(1)}$$

其中要求多项式$poly(\cdot)$独立，不依赖于关系R。

在定义1.8.10中，根据验证者的验证需要辅助输入$priv$，可分为公开验证与指定（私密）验证两种。当$priv = crs$时，验证者的验证实质上只需要非交互证明的公共输入，因此称为公开验证SNARG；否则，验证需要特定的验证密钥$priv$，即只有拥$priv$的验证者才能验证证明的可靠性，因此称为指定验证者（designated-verifier）SNARG。

如果将SNARG定义中的可靠性加强为如下的称为知识的证明性，就称为是知识的SNARG（SNARG of knowledge），简记为SNARK。

定义 1.8.11 称一个*SNARG*系统$\Pi = \langle P,\mathcal{G},V\rangle$是一个*SNARK*系统，如果它的可靠性由下面的知识的证明性代替：

– 对任意*PPT*算法P^*，存在*PPT*的算法$\mathcal{E}$，使得对任意的$z\in\{0,1\}^{poly(n)}$及充分大的n，有

$$\Pr\left[\begin{array}{l} (crs,priv)\leftarrow\mathcal{G}(1^n) \\ (x,\pi)\leftarrow P^*(w,crs), x\notin L_R \\ V(x,crs,priv)=1 \end{array} \wedge \begin{array}{l} (x,w)\leftarrow\mathcal{E}(z,crs) \\ (x,w)\notin R \end{array}\right] \leq negl(n)$$

[49]给出的构造实际上是一个SNARK系统。

在外包计算等场景，需要对计算结果进行证明，以保证计算结果的正确性，因此需要证明具有可靠性和有效性可验证性，即SNARG或更强的SNARK系统。有时，还希望简短非交互论证同时还具有零知识性，称为简短非交互（知识的）零知识论证（succinct NIZK argument of knowledge），也记为zk-SNARK。作为知识的零知识论证，zk-SNARK也有广泛应用，相关研究工作也吸引人们的关注。

最近，[51]采用QSP（Quadratic Span Programs）给出了比利用PCP更有效的SNARG系统。

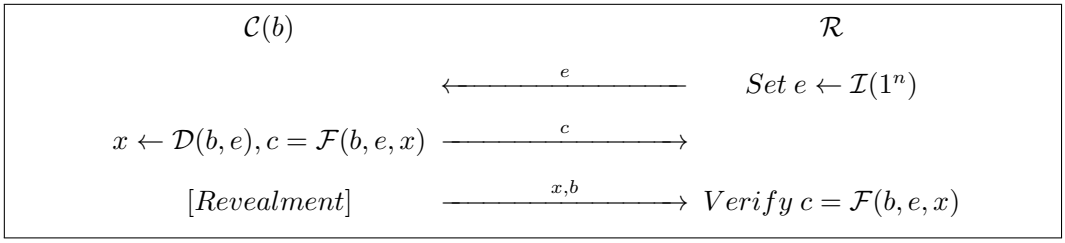
1.9 常数轮零知识证明

交互证明的交互次数（轮数）反映了协议的复杂性，故称它为协议的轮复杂性。在实际应用中，人们总是希望能得到较低轮数的交互证明，轮数最低的即非交互证明（1轮协议），但是它只在特殊模型下存在（如CRS模型）。由协议1.3.2可知，任意NP问题都存在3轮零知识证明，但由其错误概率为1/2，为得到错误概率可忽略的零知识证明，需要顺序运行n次（这里n 是安全参数），这使得协议的轮数与安全参数相关，不再是常数。协议1.3.2的并行可以保证协议的轮数不变，也可以降低错误概率，但复合协议的零知识无法证明。因此，如何获得常数轮且错误概率可忽略的零知识证明称为需解决的一个重要问题，直到1996年，Goldreich等人构造出了G3C常数轮零知识证明系统，从而也就得到了任意NP 问题的常数轮零知识证明系统。Goldreich等人给出的协议其零知识性证明较为复杂，Ross后来(2004，TCC)对HC问题给出一个证明相对简明的常数轮协议。

NP完全问题G3C与HC都存在3轮的错误概率为常数的ZKP，如果将协议并行运行多次，则可以使错误概率降低至可忽略，但问题是零知识性证明遇到了困难。回想HC的零知识证明（协议1.3.2）或者G3C的证明（协议1.3.3），其模拟器的工作方式为：在给出承诺前随机猜测验证者的挑战，然后根据猜测做出承诺，若猜测正确，模拟成功；若猜测错误，重复此过程。由于验证者的挑战只有1比特，模拟器猜测正确的概率为1/2，所以模拟器能1/2概率成功。当n个协议并行时，验证者的挑战是$\sigma \in \{0,1\}^n$，此时模拟器实现猜测成功的概率为2^{-n}，因此在多项式时间内成功的概率是可忽略的。

若模拟器采用黑盒重绕（rewinding）方法产生（证明者的）承诺消息，即模拟器首先计算一个随机承诺，并以此调用验证者V^*，当收到其发出的挑战后，再以此挑战重新计算承诺，保证可以按挑战可以正确公开。这种方法看似可以解决问题，实则仍有问题。当验证者的挑战与承诺无关时，方法是有效的；若验证者的挑战与承诺相关，则将黑盒重绕方法产生承诺交给V^*会使其产生另一个新的挑战，从而导致黑盒重绕方法失效。为解决这个问题，方法之一是强迫验证者V^*的挑战与承诺无关，Goldreich等人采用的方法是，在原3轮协议前，增加一个子协议，让验证者首先对自己的随机挑战进行承诺，使得验证者不能根据承诺改变挑战，从而使黑盒重绕的模拟方法有效。

由于证明者的计算能力无限，验证者对随机挑战的承诺必须是统计（完美）隐藏的（见2.3.1小节）。已经知道，基于单项置换可以实现统计隐藏的承诺，但方案不是常数轮，所以不能用于常数轮零知识证明。基于Claw-free 函数簇，Goldreich 等人（1996）给出了两轮承诺方案。设$(\mathcal{I},\mathcal{D},\mathcal{F})$为一个Claw-free函数簇，其中$\mathcal{I}$ 是指标抽样算法，$\mathcal{D}$是函数定义域上的抽样算法，$\mathcal{F}$ 是函数求值算法。具体地，对任意的$e \leftarrow \mathcal{I}$，对应于一对满射函数f_e^0, f_e^1，定义域分别为D_e^0, D_e^1，而值域同为D。对任意b，算法$\mathcal{D}(b,e)$实现D_e^b上的抽样，即$x \leftarrow \mathcal{D}(b,e)$，而$f_e^b(x) = \mathcal{F}(b,e,x)$。所谓claw-free性质，是对任意$e \leftarrow \mathcal{I}$，寻找$x_0 \in D_e^0, x_1 \in D_e^1$满足$f_e^0(x_0) = f_e^1(x_1)$（称为claw）在计算上是困难的。统计隐藏承诺方案如下：



承诺方案的统计隐藏性和计算绑定性都源于Claw-free函数的性质。函数f_e^0, f_e^1是满射保证了统计隐藏性，寻找claw的困难性保证了计算隐藏性。

设$G = (W, E)$为有向图，$n = |W|$，H是G中的Hamiltonian Cycle。$Com(\cdot,\cdot)$是一个承诺方案。由3轮的基本协议（协议1.3.2），Ross给出的零知识证明如下：

协议 1.9.1 *Constant round ZKP for HC*

step1. 承诺挑战；

P1-1. P启动一个perfect hiding承诺协议$Com_h(\cdot)$，发送消息P11。

V1-1. V首先随机选择$\sigma \in \{0,1\}^n$，$\{\sigma_i^0\}_{i=1}^k$，$\{\sigma_i^1\}_{i=1}^k$，满足$\sigma_i^0 \oplus \sigma_i^1 = \sigma$，$i = 1, \cdots, k$，并对$\sigma \in \{0,1\}^n$，$\{\sigma_i^0\}_{i=1}^k$，$\{\sigma_i^1\}_{i=1}^k$进行承诺：

$$C_\sigma = Com_h(\sigma; r_\sigma),\ C_i^0 = Com_h(\sigma_i^0; r_i^0),\ C_i^1 = Com_h(\sigma_i^1; r_i^1)$$

然后将$V11 = \{C_\sigma\} \cup \{C_i^0, C_i^1\}_{i=1}$发送给$P$。

P1-2. P随机选择k-bit的串$r = r_1 \cdots r_k$并交给V。记$P12 = r$。

V1-2. V公开对$\sigma_1^{r_1}, \cdots, \sigma_k^{r_k}$的承诺。记承诺公开信息为$V12$。

step2. 若公开的承诺正确，双方并行运行n个3轮基本协议，V以$\sigma = \sigma_1 \cdots \sigma_n$为挑战：

P2-1. P独立随机选择顶点集合上的置换π_j，采用Com对$\pi_j(G)$的邻接阵进行承诺，记为$C_j = Com(\pi_j(G))$，$j = 1, \cdots, n$，并将$P21 = \{C_j\}_{j=1}^n$发给V。

V2-1. V公开对$\sigma_1^{1-r_1}, \cdots, \sigma_k^{1-r_k}$及$\sigma$的承诺。记公开的消息为$V21$。

P2-2. P验证公开的承诺与$\sigma_i^0 \oplus \sigma_i^1 = \sigma$，$i = 1, \cdots, k$。若验证通过，则以$\sigma_j$为第$j$个基本协议的挑战，计算第$j$ 个基本协议的回复。记所有的回复为$P22$。

V2-2. V按照基本协议验证P22中的回复，V接受当且仅当验证通过。

引理 1.9.1 协议*1.9.1*是*HC*的交互证明。

证明：

– 完备性：显然。

– 可靠性：若承诺方案$Com_h(\cdot)$是统计隐藏的，则可靠性成立。

引理 1.9.2 协议*1.9.1*是*HC*的零知识证明。

证明： 首先构造模拟器$\mathcal{S}^*(G)$如下：

S1. 为V^*准备适当的随机串s，执行P1-1生成$P11$提供给V^*，得到V^*的承诺$V11 = V^*(G, P11; s)$。

S2. 生成$P12 = r_1 \cdots r_k$执行P1-2，得到V^*对部分承诺的公开：$V12 = V^*(G, P11, P12; s)$。

1) 若$V12$错误（公开承诺失败），输出$(P11, V11)$，终止。
2) 若$V12$正确，执行下一步。

S3. 对$j = 1, 2, \cdots$，完成如下过程：

1) 随机选取$P12_j = r_1 \cdots r_k$，执行P1-2并得到$V12_j = V^*(G, P11, P12_j, s)$（对$\sigma_1^{r_1}, \cdots, \sigma_k^{r_k}$的承诺的公开）。
2) 若$V12_j$正确，执行S4；
3) 若$V12_j$不正确，重复下一轮。

S4. 设S3结束后，对$P12_j = r'_1 \cdots r'_k$是得到了正确的$V12_j$。

1) 若$P12 = P12_j$，即$r_1 \cdots r_k = r'_1 \cdots r'_k$，输出$\perp$并终止（模拟器无法获得挑战$\sigma$）。
2) 若$P12 \neq P12_j$，则存在i使得$r_i \neq r\prime_i$。记$\sigma = \sigma_i^{r_i} \oplus \sigma_i^{r'_i}$
3) 根据σ生成合适的$P21$完成P2-1，并获得$V21$（V^*在V2-1对承诺的公开，包括σ）。若$V21$正确，根据σ生成$P22$完成P2-2。最后输出$(p11, V11, P12, V12, P21, V21, P22)$并终止。

模拟器$\mathcal{S}^*(G)$的运行时间是多项式的。事实上，设$\zeta = \zeta(G, P11, s)$是V^*在V12正确的概率，$p_i(n)$（$i = 1, 2, 3, 4$）为模拟器完成S1-S4所需要的（期望）时间，注意到$p_1(n), p_2(n)$及$p_4(n)$一定是多项式的，而$p_3(n) = \frac{1}{\zeta} p_2(n)$，于是，模拟器的$\mathcal{S}^*(G)$完成S1-S4的期望时间为：

$$\begin{aligned} p_1(n) + p_2(n) + p_3(n) + p_4(n) &= p_1(n) + (1 - \zeta)p_2(n) + \zeta\left(\frac{1}{\zeta}p_2(n) + p_4(n)\right) \\ &\leq p_1(n) + 2p_2(n) + p_4(n) = poly(n) \end{aligned}$$

Claim 1.9.2.1 若承诺方案Com具有计算隐藏性性，则$\{\mathcal{S}^*(G)\}_{G \in HC}$与$\{View_{V^*}^P(G)\}_{G \in HC}$计算不可区分。

证明： 首先定义一个混合模拟器$\mathcal{S}^*$。$\mathcal{S}^*$与$\mathcal{S}$基本相同，唯一的区别是在进入协议的step 2时（即模拟算法的S4），$\mathcal{S}^*$获得了G中的一个Hamiltonian Cycle H做为辅助输入。若模拟器$\mathcal{S}$输出$\perp$，则$\mathcal{S}^*$也输出$\perp$；否则，$\mathcal{S}^*$按照协议正常运行并输出。混合模拟器$\mathcal{S}^*$的输出记为$\mathcal{S}^*(G, H)$。

(1) $Pr[\mathcal{S}^*(G,H)=\perp]$可忽略。事实上，设$\zeta$的意义同前，有

$$\begin{aligned} Pr[\mathcal{S}^*(G,H)=\perp] &= Pr[\mathcal{S}^*(G,H)=\perp|(\mathcal{S}^* \ reaches \ step \ 3)]\cdot Pr[\mathcal{S}^* \ reaches \ step \ 3] \\ &= Pr[\widehat{M^*}(G,H)=\perp|(\mathcal{S}^* \ reaches \ step \ 3)]\cdot \zeta \\ &= Pr[P12=P12_j]\cdot\zeta \end{aligned}$$

由于$P12, P12_j \in \{0,1\}^k$均匀独立分布，且$\zeta = Pr[V12 \neq abort]$的概率，因此对给定的随机数$s$（$V^*$的随机输入），使$V^*$的回复$V12$正确的$r$共有$2^k\cdot\zeta$，于是$Pr[P12=P12_j]=\frac{1}{2^k\cdot\zeta}$，从而就有

$$Pr[\mathcal{S}^*(G,H)=\perp]=\frac{1}{(2^k\cdot\zeta)}\cdot\zeta=\frac{1}{2^k}$$

(2) $\{\mathcal{S}(G)\}_{G\in HC}$与$\{\mathcal{S}^*(G,H)\}_{G\in HC}$计算不可区分。对$\sigma_j=1$，$\mathcal{S}^*(G,H)$对$\pi_j(G)$的邻接矩阵进行承诺，公开与$\pi_j(H)$的边对应的承诺，而$\mathcal{S}(G)$是对随机Hamilton圈（作为一个图）对应的邻接矩阵进行承诺，公开同样是与随机Hamilton圈的边对应的承诺，这是二者的唯一不同之处。因此，根据承诺方案$Com(\cdot)$的隐藏性，$\{\mathcal{S}(G)\}_{G\in HC}$与$\{\mathcal{S}^*(G,H)\}_{G\in HC}$计算不可区分。

(3) 在不输出$\perp$的条件下，$\{\mathcal{S}^*(G,H)\}_{G\in HC}$与$\{View^P_{V^*}(G)\}_{G\in HC}$同分布。

综上可知协议是一个零知识证明。 ∎

定理 1.9.3 协议*1.9.1*是*HC*问题的常数轮零知识证明系统（具有可忽略的错误概率，*EPT*的模拟器）。

定理 1.9.4 若存在*perfect hiding*承诺方案，则存在*HC*的常数轮零知识证明系统（具有可忽略的错误概率，*EPT*的模拟器）。

定理 1.9.5 若*non-uniformly claw-free* 函数族存在，则对任意的*NP*问题，存在常数轮零知识证明系统（具有可忽略的错误概率，*EPT*的模拟器）。

1.10　非黑盒零知识

交互证明$\langle P,V\rangle$的零知识性要求对任意的V^*，存在PPT模拟器$\mathcal{S}_{V^*}$（与V^*相关），满足$\{\mathcal{S}_{V^*}(x)\}_{x\in L}$与$\{View^P_{V^*}(x)\}_{x\in L}$不可区分。很长时间以来，实际构造模拟器$\mathcal{S}_{V^*}$都以黑盒方式使用$V^*$，即模拟器仅以黑盒方式使用验证者的策略$V^*$，故将所得到的零知识性也就称为黑盒零知识。2001年，Barak等人给出了一种以非黑盒方式使用验证者算法V^*的模拟方法，并由此得到了在黑盒模拟无法得到的结果：常数轮可有界并发的零知识协议。此后，Barak非黑盒模拟方法受到了极大关注，从而也得到了更多黑盒模拟意义下不可能的结果。

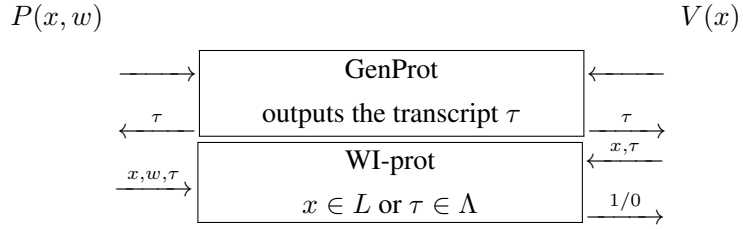
本节介绍Barak在2001年给出的关于NP问题的非黑盒零知识论证（non-black-box zero-knowledge argument）。Barak的协议由两部分组成：预生成子协议$GenProt$和证据不可区分证明子协议$WI-prot$。

– $GenProt$：P与V之间的交互协议（游戏），P欲使其输出$\tau\in\Lambda$，而V的目的是保证$\tau\notin\Lambda$，其中Λ是一个确定的语言。

– $WI-prot$：证据不可区分的证明系统。证明原命题$x \in L$成立或者预生成子协议的输出τ满足$\tau \in \Lambda$。

为保证协议的合理性（soundness），要求在游戏$GenProt$中V一定获胜，即总有$\tau \notin \Lambda$，这样当V在$WI-prot$ 中接受时，有$x \in L$，从而V可以接受证明。协议结构如下：

Structure of Barak's scheme



下面需要定义预生成协议$GenProt$与对应的语言类Λ，为此根据验证者V分为两种情况。

1.10.1 一致的验证者（uniform verifier）

若交互证明的验证者为一致的，定义预生成子协议和对应的Λ如下：

定义 1.10.1 设$GenProt$是两方协议，$\Lambda \subseteq \{0,1\}^*$是$Ntime(T(n))$中的语言，其中$T : \mathbb{N} \to \mathbb{N}$多项式时间可计算*(*例如$T(n) = n^3, T(n) = n^{\log\log n}$*)*，$n$是安全参数。称$GenProt$是生成协议（*generation protocol*）如果满足以下两个条件：

– *Soundness:* 设$GenProt$的输出为τ，若V正确执行协议一，那么对P的任意策略，都有

$$Pr[\tau \in \Lambda] < \mu(n)$$

其中$\mu(\cdot)$是某可忽略函数。

– *Simulation of uniform verifiers:* 存在*PPT*模拟器$S_{GenProt}$，若输入V^*的描述（设V^*是验证者的多项式时间的交互策略，对其的描述不超过$2n$*-bits*），$S_{GenProt}$在多项式时间内输出(v, σ)，满足：

- $View_{GenProt,V^*}(1^n)$与v计算不可区分。
- $(\tau, \sigma) \in R_\Lambda$（其中$\tau$为$GenProt$的输出），且在关于$V^*$的运行时间的多项式时间内可验证。

设$Comm$是perfect-binding承诺方案，对任意$y \in \{0,1\}^*$，记

$$Comm^{-1}(y) = \begin{cases} x & \exists s,\ such\ that\ y = Comm(x,s) \\ \perp & others \end{cases}$$

令

$$\Lambda = \{(z,r) : \Pi(z)在|r|^{\log\log|r|/5}步内输出r, \Pi = Comm^{-1}(z)\}$$

注意到PPT模拟器$S_{GenProt}$以输入V^*的描述为输入，因此可允许以非黑盒方式使用算法V^*，但必须保证协议的可靠性成立。根据Λ的定义，构造预生成子协议实现协议$GenProt$ 如下：

协议 1.10.1 $GenProt$

$P(1^n)$		$V(1^n)$
	$\xrightarrow{z = Comm(0^{3n})}$	
	$\xleftarrow{r \in_R \{0,1\}^n}$	
$Output\ \tau = (z, r)$		$Output\ \tau = (z, r)$

引理 1.10.1 如上构造的协议$GenProt$是一个预生成协议。

证明：

- **Soundness:**显然，因为当V诚实执行协议时，$\Pr[\tau = (z, r) \in \Lambda]$一定可忽略，除非证明者可以以不可忽略的概率预测$V$的输出。
- **Simulation of uniform verifiers:** 设$G : \{0,1\}^{n/2} \to \{0,1\}^m$是伪随机发生器，其中$m$为$V^*$的随机序列的长度。定义模拟器$S_{GenProt}$如下：

  (1) 选择$u \leftarrow_R \{0,1\}^{n/2}$，计算$s = G(u)$。记将$V^*$的随机数$s$“固化”后的算法为$V^{**}$。

  (2) 计算$z \leftarrow Comm(\Pi, \theta)$，其中$\Pi$为$V^{**}$的“next message function”，对其的描述不超过$3n$-bit，θ为随机数。

  (3) 计算$r = \Pi(z) = V^{**}(z)$。

  (4) 输出(v, σ)，其中$v = (z, s)$，$\sigma = (\Pi, \theta)$。

  那么，我们有

  (a) v与$View_{GenProt,V^*}(1^n)$计算不可区分。（由G的伪随机性与承诺方案$Comm$的性质）。

  (b) $(z, r) \in \Lambda$，且$((z, r), \sigma) \in R_\Lambda$，同时，由于$\Pi$为$V^{**}$的“next message function”，而V^{**}又是确定了随机数的V^*，因此$((z, r), \sigma) \in R_\Lambda$可在关于$V^*$的多项式时间验证。 ∎

定理 1.10.2 设$L \in NP$，$WI - prot$是对$NP \cup Ntime(T)$的常数轮证据不可区分的证明（或论证，*argument*），若采用如上定义的$GenProt$与Λ，则*Barak's scheme*是L的一个一致验证者的零知识论证，并且满足：

- *Constant round;*
- *Public coins:*
- *Simulator runs in PPT.*

而且，若$WI - prot$的错误概率可忽略，则协议的错误概率也是可忽略的。

证明：显然，协议是公开投币的常数轮协议，而且协议的错误概率与$WI-prot$的错误概率相同。下面证明它是零知识论证。

- **Completeness:**显然。
- **Soundness:**由欲输出协议$GenProt$及不可区分证明$WI-prot$的可靠性得到。
- **Uniform zero-knowledge:** 设$x \in L$是欲证明的命题，V^*是验证者的交互策略对应的Turing机的描述。假设模拟器$\mathcal{S}$可以获得V^*（希望以非黑箱方式使用V^*），$\mathcal{S}(x, V^*)$按如下策略进行：

  (1) 获得x与算法描述，$\mathcal{S}(x, V^*)$首先将V^*的输入x“固化”，设V^{**}是固化后的算法描述（假设其不超过$2n$-bits）；然后调用$GenProt$的模拟算法$S_{GenProt}(V^{**})$，即$(v, \sigma) \leftarrow S_{GenProt}(V^{**})$。记$V^{***}$为将$V^{**}$中的$v$“固化”后的算法，$\tau$为$v$中的“transcript”。

  (2) 充当证明者与V^{***}运行WI-prot，完成对$x \in L \ or \ \tau \in \Lambda$的证明。记$v' = View^P_{V^{***}}(x, \tau)$。

  (3) 输出(v, v')。

  注意到第一步(1)可以在v^*的运行时间的多项式时间内完成。对于第二步(2)，运行时间与子协议$WI-Prot$相关。当$\Lambda \in NP$时，$WI-Prot$一定可在多项式时间内完成（见证据不可区分证明一节）；而当$\Lambda \notin NP$时，$WI-Prot$采用的“universal arguments”（这里略去$WI-Prot$的具体实现），可保证其也可在多项式时间内完成。总之，$\mathcal{S}(x, V^*)$可在多项式时间内完成。

  $\mathcal{S}(x, V^*)$与实际交互时验证者的“View”的不可区分性由WI-prot的证据不可区分性可得。 ■

1.10.2 非一致验证者（non-uniform verifier）

对非一致验证者，由于其算法不能用统一的描述表示，因此需要重新定义预生成协议。

定义 1.10.2 设$GenProt$是两方协议，$\Lambda \subseteq \{0,1\}^*$是$Ntime(T(n))$中的语言，其中$T : \mathbb{N} \to \mathbb{N}$多项式时间可计算，$n$是安全参数。称$GenProt$是非一致*(non-uniform)*生成协议（*generation protocol*）如果满足以下两个条件：

- *Computational Soundness:* 对任意$T(n)^{O(1)}-size$的P的任意策略，若V正确执行协议，有$Pr[\tau \in \Lambda] < \mu(n)$。其中设$GenProt$的输出为$\tau$，$\mu(\cdot)$是某可忽略函数。
- *Simulation of non-uniform verifiers:* 存在*PPT*模拟器$S_{GenProt}$，若输入V^*的描述（设V^*是验证者的多项式时间的交互策略，对其的描述不超过$2n$-*bits*），$S_{GenProt}$在多项式时间内输出(v, σ)，满足：
  - $View_{GenProt, V^*}(1^n)$与v计算不可区分。
  - $(\tau, \sigma) \in R_\Lambda$且在关于$V^*$的运行时间的多项式时间内可验证。

对非一致验证者，定义语言Λ。设$\mathcal{H}_n$为hash函数族，令

$$\Lambda = \{(h, z, r) : \exists \Pi, z = Comm(h(\Pi)), \Pi(z) \text{ outputs } r \text{ within } |r|^{\log\log|r|/5}\}$$

其中$h \in_R \mathcal{H}_n$。

根据预生成协议及Λ的定义，构造实现协议如下：

协议 1.10.2 *GenProt*

$P(1^n)$		$V(1^n)$
	$\xleftarrow{h \in_R \mathcal{H}_n}$	
	$\xrightarrow{z = Comm(h(0^n))}$	
	$\xleftarrow{r \in_R \{0,1\}^n}$	
$output\ (z, r)$		$output\ (z, r)$

引理 1.10.3 以上协议是关于Λ的生成协议。

证明：

– Computational soundness：设有$n^{O(\log\log n)} - size$的证明者P^*输出$\tau \in \Lambda$的证据的概率ε不可忽略。那么$\mathcal{H}_n$中至少有$\frac{\epsilon}{2} \cdot |\mathcal{H}_n|$使$P^*$获胜的概率不小于$\frac{\epsilon}{2}$，记为$\mathcal{H}'_n$。取定$h \in_R \mathcal{H}'_n$，不妨设$P^*$是确定算法（因是Non-uniform的，可固定随机数），于是P^*的输出z也是确定的。于是，对任意选择的r, r'，P^*将以$\frac{\epsilon^2}{4}$输出相应的Π, Π'，使

$$z = Comm(h(\Pi)) = Comm(h(\Pi')), \Pi(z) = r, \Pi'(z) = r'$$

由$Comm$的perfect-binding性质，必有$h(\Pi) = h(\Pi')$，于是有$n^{O(\log\log n)}$规模的算法以概率$O(\epsilon^2)$求得h的一对碰撞$\Pi \neq \Pi'$，这与对$\mathcal{H}_n$的假设矛盾。(这里要求$\mathcal{H}_n$可以抵御$n^{\log n}$-size的敌手)。

– Simulation of non-uniform verifiers: 定义模拟器$S_{GenProt}(V^*, 1^n)$如下：

(1) $h \leftarrow V^*(\theta)$，V^*的随机数为θ。

(2) 计算$z \leftarrow Comm(h(\Pi), s)$，其中$\Pi$为$V^*$的“next message function”，s为随机数。

(3) 计算$r = \Pi(z)$。

(4) 输出(v, σ)，其中$v = (z, \theta)$，$\sigma = (\Pi, s)$为证据（$\tau = (h, z, r) \in \Lambda$）。

则

- v与$View_{GenProt, V^*}(1^n)$计算不可区分。
- $(\tau, \sigma) \in R_\Lambda$ 且在关于V^*的运行时间的多项式时间内可验证。 ∎

定理 1.10.4 设$L \in NP$，Σ是对$NP \cup Ntime(T)$的证据不可区分的知识的证明（或论证，*argument*），若采用如上定义的$GenProt$与Λ，则关于*non-uniform*验证者的*Barak*方案，定理*1.10.2*的结论依然成立。

证明：

- Completeness:
- Soundness: 若有$x \notin L$，使$Pr[\langle P, V\rangle(x) = 1] = \epsilon$不可忽略，则有知识抽取器$\mathcal{K}$，使$w \leftarrow \mathcal{K}^{P_{WI}}$，这里$P_{WI}$是证明者在$\Sigma$中的策略。由于$x \notin L$，故$w$一定是$\tau \in \Lambda$的一个证据。因此证明者以不可忽略的概率得到$\tau \in \Lambda$的证据，与$GenProt$的性质矛盾！
- Zero knowledge:定义模拟器$\mathcal{S}(x, V^*, 1^n)$如下：

  (1) $(v, \sigma) \leftarrow S_{GenProt}(V^{**})$，其中$V^{**}$是$V^*$将$x$“固化”后的算法描述（假设其不超过$2n$-bits）。记$V^{***}$为将$V^{**}$中的$v$“固化”后的算法，$\tau$为$v$中的“transcript”。

  (2) 在证明者（使用σ）与V^{***}之间运行Σ，证明$x \in L$ *or* $\tau \in \Lambda$。记$v\prime = View_{\sigma, V^{***}}$。

  (3) 输出(v, v')。

  (v, v')与$View_{V^*}$的不可区分性可采用“Hybridargument”方法说明。 ∎

1.11 泄露容忍零知识

参考文献

[1] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. SIAM J. Computing, 18(1):186 – 208, 1989.

[2] M. Ben-Or, O. Goldreich, S. Goldwasser, et al. Everything provable is provable in zero-knowledge. In CRYPTO 1990, pages 37-56.

[3] A. Shamir. IP=PSPACE. In FOCS,1990, pages 11-15.

[4] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. J. ACM, 38(3):690 – 728, 1991.(FOCS 1986, pages 174-187)

[5] M. Blum. How to prove a theorem so no one else can claim it. In Proceedings of the International Congress of Mathematicians, 1986, pages 1444-1451.

[6] S. Halevi and S. Micali. Practical and provably-secure commitment schemes from collision-free hashing. In Crypto, 1996.

[7] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for NP using any one-way permutation. Journal of Cryptology, 11(2):87 – 108, 1998.(preliminary version in CRYPTO 92)

[8] I. Haitner, O. Horvitz, J. Katz, C Y. Koo, R. Morselli, and Shaltiel. Reducing complexity assumptions for statistically-hiding commitment. In EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT, 2005.

[9] I. Haitner, O. Reingold. Statistically-hiding commitment from any one-way function. Thirty-Ninth ACM Symposium on Theory of Computing. ACM, 2007:1-10.

[10] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. Symposium on Foundations of Computer Science. IEEE Computer Society, pages 230 – 235, 1989.

[11] U. Feige. Alternative models for zero knowledge interactive proofs. Ph.D. thesis, Weizmann Institute of Science, 1990.

[12] J. Kalian, E. Petrank, C. Rackoff. Lower bounds for zero-knowledge on internet. In 39th FOCS, 1998, pages 560-569.

[13] R. Richardson, J, Kilian. On the concurrent composition of zero knowledge proofs. In EUROCRYPT99, LNCS 1592, pages 415-431.

[14] O.Glodreich, H. Krawczyk. On the composition of zero knowledge proofs system. SIAM, J. on Computing, 1996, 25(1):169-192.

[15] A. Rosen. A note on the round-complexity of concurrent zero-knowledge. In CRYPTO2000, LNCS 1880, pages 451-468.

[16] R. Canetti, J. Kilian, E. Petrank, A. Rosen. Black-box concurrent zero-knowledge requires $\Omega(logn)$ rounds. ACM Symposium on Theory of Computing. ACM, 2001, pages 570-579.

[17] J. Kilian, E. Petrank, R. Richardson. Concurrent zero-knowledge proofs for NP

[18] M. Prabhakaran, A. Rosen, A. Sahai. Concurrent Zero Knowledge with Logarithmic Round-Complexity. In Symposium on Foundations of Computer Science. IEEE Computer Society, 2002:366-375.

[19] L. Fortnow. The complexity of perfect zero knowledge. In Advances in Computing Research, Volume 5, pages 327-343.

[20] W. Aiello, J. Hastad. Statistical zero knowledge language can be recognized in two rounds. Journal of Computer and Sysytem sciences, 42(3): 327-345, 1991.

[21] O. Goldreich, A. Sahai, S. P. Vadhan. Honest-verifier statistical zero knowledg equals general statistical zero knowledge. In 30th ACM Symposium on Theory of Computing, pages 399-408, 1998.

[22] S. P. Vadhan. A study of statistical zero-knowledge. PhD. Thesis.

[23] O. Goldreich, S. P. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In 14th IEEE Conference on Computations to Complexity, pages 54-73, 1999.

[24] A.Sahai, S.P. Vadhan. A complete problem for statistical zero knowledge. Journal of the ACM, 50(2):196-249, 2003.

[25] R. Ostrovsky. One-way functions, hard on average problems, and statistical zero-knowledge proofs. In Proceedings of the Sixth, Structure in Complexity Theory Conference, 1991, pages 133-138.

[26] R. Ostrovsky, A. Wigderson. One-way fuctions are essential for nontrivial zero-knowledge. Istcs, 1993:3 - 17.

[27] T. Okamoto. On relationships between statistical zero-knowledge proofs. Journal of Computer and System Science, 60(1):47-108, 2000.

[28] Salil P. Vadhan. An unconditional study of computational zero knowledge. SIAM Journal on Computing, 36(4):1160-1214, 2006.(Preliminary version in FOCS2004)

[29] S. J. Ong S J. Unconditional relationships within zero knowledge. Harvard University, 2007.

[30] A. Fiat, A. Shamir: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO 1986, pages 186-194.

[31] D. Pointcheval, J. Stern. Security Proofs for Signature Schemes. In Advances in Cryptology — EUROCRYPT ' 96. EUROCRYPT 1996. LNCS, 1070, pages387-398.

[32] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. Proceedings of the 22nd Annual Symposium on Theory of Computing, ACM, 1990.

[33] A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In FOCS,1999:543-553.

[34] O. Goldreich, J. Hastad. On the complexity of interactive proofs with bounded communication. Information Processing Letters, 67(4):205 – 214, 1998.

[35] O. Goldreich, S. Vadhan, A. Wigderson. On interactive proofs with a laconic prover. Computational Complexity, 11(1-2):1 – 53, 2002.

[36] X. Boyen, B. Waters. Compact group signatures without random oracles. In EUROCRYPT 2006. LNCS, 4004, pages 427 – 444.

[37] N. Chandran, J. Groth, A. Sahai. Ring signatures of sub-linear size without random oracles. In ICALP 2007. LNCS, 4596, pages 423 – 434.

[38] M. Blum, P. Feldman, S. Micali. Non-interactive zero-knowledge and its applications. In: STOC 1988, pages 103 – 112.

[39] U. Feige, D. Lapidot, A. Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. SIAM Journal of Computing 29(1): 1 – 28,1999.

[40] J. Groth, R. Ostrovsky, A. Sahai. Non-interactive zaps and new techniques for NIZK. In CRYPTO 2006. LNCS, 4117, pages 97 – 111.

[41] J. Groth, A. Sahai. Efficient non-interactive proof systems for bilinear groups. In EUROCRYPT 2008. LNCS, 4965, pages 415 – 432.

[42] D. Boneh, E-J. Goh, K. Nissim. Evaluating 2-DNF Formulas on Ciphertexts. International Conference on Theory of Cryptography. Springer-Verlag, 2005:325-341.

[43] A.De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, A. Sahai. Robust Non-interactive Zero Knowledge. In Advances in Cryptology —CRYPTO 2001. CRYPTO 2001. LNCS, 2139, pages 566-598.

[44] J. Kilian. A note on efficient zero-knowledge proofs and arguments. In Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC 1992), pages 723 – 732 (1992).

[45] S. Micali. Computationally sound proofs. SIAM Journal on Computing 30(4), 1253 – 1298 (2000). (Preliminary version appeared in FOCS 1994)

[46] W. Aiello, Sandeep N. Bhatt, R. Ostrovsky, S. Rajagopalan. Fast verification of any remote procedure call: Short witness-indistinguishable one-round proofs for NP. In Proceedings of the 27th International Colloquium on Automata, Languages and Programming, pages 463 – 474, 2000.

[47] C. Dwork, M. Langberg, M. Naor, K. Nissim, O. Reingold. Succinct NP proofs and spooky interactions, December 2004. Available at www.openu.ac.il/home/mikel/papers/spooky.ps.

[48] G. Di Crescenzo, H. Lipmaa. Succinct NP Proofs from an Extractability Assumption. In Proceedings of the 4th Conference on Computability in Europe. 2008, LNCS, 5028, pages 175 – 185, 2008.

[49] N. Bitansky, R. Canetti, A. Chiesa, E. Tromer. From ectractable collision resistance to succinct non-interactive arguments of knowledge, and back again.

[50] J. Groth. Short non-interactive zero-knowledge proofs. In Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security, ASIACRYPT ’ 10, pages 341 – 358, 2010.

[51] R. Gennaro, C. Gentry, B. Parno, M. Raykova. Quadratic Span Programs and Succinct NIZKs without PCPs. In Advances in Cryptology – EUROCRYPT 2013. EUROCRYPT 2013. LNCS 7881, pages 626-645.

[52] O. Goldreich, A. Kahan. How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology, 1996, 9(3):167-189.

[53] A. Rosen. A Note on Constant-Round Zero-Knowledge Proofs for NP. In TCC 2004, pages 191-202, 2004