Zero Knowledge Mechanisms

A Brief Introduction

minxuan mei

2025-06-01

Zhejiang University

Preliminaries

1.1 Standard Definitions

1. 参与者，类型，结果与效用函数

   a. 参与者：有限多个参与者 $i = 1, ..., n$;

   b. 类型：每个参与者 i 有一个从有限类型集 T_i 中获取的私有类型 t_i;

   c. 结果：一个有限集 X 包含了所有可能的结果;

   d. 效用函数：$u_i : T_i \times X \to \mathbb{R}$.

2. 机制和激励

   记 $T = \times_{i=1}^{n} T_i$.

   机制：随机化函数 $M : T \to \Delta(X)$，可视为确定性函数 $M : T \times \{0,1\}^{n_e} \to X$，第二个参数为一 n_e 比特长的均匀随机采样比特串.

a. 个体理性（IR）：$\forall i = 1, ..., n, t \in T, u_i(t_i, M(t)) \geq 0$

b. 占优策略激励相容（DSIC）：$\forall i = 1, ..., n, t \in T, t'_i \in T_i, \mathbb{E}[u_i(t_i, M(t))] \geq \mathbb{E}[u_i(t_i, M(t'_i, t_{-i}))]$

1.2 Languages for Mechanisms and Proofs

机制解释器

$$I : \{0,1\}^* \times T \times \{0,1\}^* \times \mathbb{N} \to X \cup \{\text{error}\}$$

输入：

1. 机制描述 $A \in \{0,1\}^*$
2. 类型组合 $t \in T$
3. 机制随机数比特串 $r_e \in \{0,1\}^*$
4. 最大运行时间 $R \in \mathbb{N}$

输出为结果或出错，出错有三种情况

1. 机制描述 A 不合法
2. 没有指定足够的随机比特
3. 运行时间超过 R

固定 I，对任意合法机制描述 A，如果其描述的机制的评估时间 $\leq R$，并且需要的随机比特数 $\leq n_e$，便记 M_A 为 A 所描述（I 所解释）的机制，$M_{A(t,r_e)} \stackrel{\text{def}}{=} I(A,t,r_e,R)$.

1.3 The Protocol

1. 传统的提交后运行协议流程：

   a. 提交阶段：机制设计者公开宣布机制的描述（需要至多 n_e 个随机比特），并且提供一个证明该机制是 IR 和 DSIC 的证明.每个参与者均可通过验证该证明来验证机制的正确性.

   b. 直接披露阶段：每个参与者 i 直接披露其类型 t_i.

   c. 评估随机数生成阶段：自然抽取一个长度为 n_e 的均匀随机比特串并且将其公开.

   d. 评估阶段：机制设计者公开机制运行结果.每个参与者都可以依据公开的机制描述、类型、随机数比特串来验证机制的运行结果.

2. 第三方介导的提交后运行协议通过引入可信第三方隐藏机制信息，流程如下：

a. 提交阶段: 机制设计者向第三方提供机制描述 A, 并且提供一个证明该机制是 IR 和 DSIC 的证明.

b. 提交验证阶段：第三方验证机制描述 A 以及证明的合法性，并将验证结果公开.

c. 直接披露阶段：每个参与者 i 直接披露其类型 t_i，形成的类型组合记为 t.

d. 评估随机数生成阶段：自然抽取一个长度为 n_e 的均匀随机比特串 r_e 并且将其公开.

e. 评估阶段：给定输入 t, r_e，机制设计者公开结果 x.

f. 评估验证阶段：第三方评估 $M_A(t, r_e)$ 并公布其与 x 的一致性.

Mechanism Hiding via Zero-Knowledge Proofs

2.1 Trusted Commitment to Hidden Mechanism: Overview

为了能够不借助可信第三方而实现提交后运行协议，需要将第三方介导协议中的一些步骤利用密码学技术进行替换.

1. 替换提交阶段: 机制设计者向参与者发送信息 D_c. 第一部分为**机制描述的密码学提交**，可以理解为机制描述的加密版本，记为 c；第二部分为一个证明 c 所描述的机制是 IR 和 DSIC 的**非交互式零知识证明**，其不会解密 c，也不会泄露机制的信息.

2.1 Trusted Commitment to Hidden Mechanism: Overview

为了能够不借助可信第三方而实现提交后运行协议，需要将第三方介导协议中的一些步骤利用密码学技术进行替换.

1. 替换提交阶段: 机制设计者向参与者发送信息 D_c. 第一部分为**机制描述的密码学提交**，可以理解为机制描述的加密版本，记为 c；第二部分为一个证明 c 所描述的机制是 IR 和 DSIC 的**非交互式零知识证明**，其不会解密 c，也不会泄露机制的信息.
2. 替换评估阶段: 机制设计者向参与者发送信息 D_e，这是一个证明公开的结果 x 是使用提交阶段中 c 所描述的机制，以直接披露阶段中的类型组合 t 为输入评估得到的非交互式零知识证明.

2.1 Trusted Commitment to Hidden Mechanism: Overview

显然, 如果攻击者具有近乎无限的计算资源, 他便可以通过穷举的方法攻破加密手段; 即便其没有计算资源, 他也可以猜测密钥, 尽管成功的概率很小.

所以, 现代密码学抵御的是有限资源的攻击者, 并且保证这样的攻击者攻破的概率是可忽略的. 前沿加密技术安全性保证一般都会规约到一个难解问题（如大数分解, 离散对数求解）, 这也会在协议中体现.

2.2 Formal Definition: Commit-and-Run Protocols

接下来通过参数形式化先前提到的提交后运行协议.

由参数 $n_c, L_c, \psi_c, n_e, L_e, \psi_e$ 参数化的直接披露的提交后运行协议按以下步骤执行:

1. 提交随机数生成阶段: 自然抽取一个长度为 n_c 的均匀随机比特串 r_c 并且将其公开.
2. 提交阶段: 机制设计者以 r_c 作为输入, 选择一个提交信息 $D_c \in \{0,1\}^{L_c}$ 并且将其公开.
3. 提交验证阶段: 每个参与者都可以评估 $\psi_c(D_c, r_c) \in \{\text{True}, \text{False}\}$ 来验证其是否成立. 如果其成立那么参与者便可以相信机制设计者提交了一个 IR 并且 DSIC 的机制的描述 A.
4. 直接披露阶段: 每个参与者 i 直接披露其类型 t_i, 形成的类型组合记为 t.
5. 评估随机数生成阶段: 自然抽取一个长度为 n_e 的均匀随机比特串 r_e 并且将其公开.

6. 评估阶段：机制设计者以 t 和 r_e 作为输入，公开一个结果 x 作为机制运行结果，并且选择一条评估信息 $D_e \in \{0,1\}^{L_e}$ 并且将其公开.

7. 评估验证阶段：每个参与者都可以评估 $\psi_e(r_c, D_c, t, r_e, x, D_e) \in \{\text{True}, \text{False}\}$ 来验证其是否成立. 如果其成立那么参与者便可以相信 $x = M_{A(t,r_e)}$，其中 A 是提交阶段中固定的机制描述.

2.2 Formal Definition: Commit-and-Run Protocols

提交策略: $S_c : \{0,1\}^{n_d} \times \{0,1\}^{n_c} \to \{0,1\}^{L_c}$, 第一个参数是机制设计者私有的随机数比特串 r_d, 第二个参数是提交随机数比特串 r_c, 输出提交信息 D_c, n_d 便是机制设计者的私有随机数比特串长度.

评估策略: $S_e : \{0,1\}^{n_d} \times \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \to X \times \{0,1\}^{L_e}$, 对应输入参数分别为:

1. 机制设计者私有的随机数比特串 r_d,
2. 提交随机数比特串 r_c,
3. 提交信息 D_c,
4. 类型组合 t,
5. 评估随机数比特串 r_e.

输出结果 x 和评估信息 D_e.

以上二者 (S_c, S_e) 统称为机制设计者策略.

2.2 Formal Definition: Commit-and-Run Protocols

Notation	Meaning
Setting	
n	# of players
X	possible outcomes
T	possible type profiles
$M : T \to \Delta(X) / M : T \times \{0,1\}^{n_e} \to X$	direct-revelation mechanism
Mechanisms and Proofs	
A	mechanism description
$\mathcal{A}_{L_a, n_e, R}$	descriptions of length $\leq L_a$ with required randomness bits $\leq n_e$ and evaluation time $\leq R$
M_A	mechanism defined by description A
P	Proof
$\mathcal{P}_{L_p}$	proof of length at most L_p

Notation	Meaning
Protocol: Commitment	
n_c	# of commitment random bits
r_c	commitment random bits
L_c	length in bits of commitment message
$D_c \in \{0,1\}^{L_c}$	commitment message
$S_c : \{0,1\}^{n_d} \times \{0,1\}^{n_c} \to \{0,1\}^{L_c}$	mechanism designer commitment strategy
$\psi_c : \{0,1\}^{n_c} \times \{0,1\}^{L_c} \to \{\text{True}, \text{False}\}$	commitment verifier
Protocol: Evaluation	
n_e	# of evaluation random bits
$r_e \in \{0,1\}^{m_e}$	evaluation random bits
L_e	length in bits of evaluation message
$D_e \in \{0,1\}^{L_e}$	evaluation message

Notation	Meaning
$S_e : \{0,1\}^{n_d} \times \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \to X \times \{0,1\}^{L_e}$	mechanism designer evaluation strategy
$\psi_e : \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \times X \times \{0,1\}^{L_e} \to \{\text{True}, \text{False}\}$	evaluation verifier
Specification	
L_a	bound on description lengths
n_e	bound on # of mechanism evaluation random bits
R	bound on mechanism evaluation time
L_p	bound on proof lengths
B	bound on attacker running time
ε	bound on attacker success probability
$\sigma = (L_a, n_e, R, L_p, B, \varepsilon) \in \Sigma$	specification

Notation	Meaning
Playbook	
n_d	# of mechanism-designer private random bits
$r_d \in \{0,1\}^{n_d}$	mechanism-designer private random bits
$\hat{S}_c : \mathcal{A}_{L_a,n_e,R} \times \mathcal{P}_{L_p} \times \{0,1\}^{n_d} \times \{0,1\}^{n_c} \to \{0,1\}^{L_c}$	commitment playbook(recommended strategy for well-behaved mechanism designer)
$\hat{S}_e : \mathcal{A}_{L_a,n_e,R} \times \{0,1\}^{n_d} \times \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \to X \times \{0,1\}^{L_e}$	evaluation playbook(recommended strategy for well-behaved mechanism designer)
Protocol Catalog	
$\left((n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma), \left(\hat{S}_c^\sigma, \hat{S}_e^\sigma\right)\right)_{\sigma\in\Sigma}$	protocol catalog(protocol-playbook pairs)

2.2 Formal Definition: Commit-and-Run Protocols

技术参数：$\sigma = (L_a, n_e, R, L_p, B, \varepsilon)$ 用来表示零知识协议的配置参数，并记所有可能的技术参数所形成的集合为 Σ.

操作手册：给出构建提交信息和评估信息的指令.

1. 提交操作手册：$\hat{S}_c : \mathcal{A}_{L_a,n_e,R} \times \mathcal{P}_{L_p} \times \{0,1\}^{n_d} \times \{0,1\}^{n_c} \to \{0,1\}^{L_c}$，输入为机制描述 A，证明 P，机制设计者私有的随机数比特串 r_d 和提交随机数比特串 r_c，输出提交信息 D_c. 可以理解为该操作手册推荐机制设计者使用提交策略 $\hat{S}_c(A, P, \cdot, \cdot)$.
2. 评估操作手册：$\hat{S}_e : \mathcal{A}_{L_a,n_e,R} \times \{0,1\}^{n_d} \times \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \to X \times \{0,1\}^{L_e}$，输入为机制描述 A，证明 P，机制设计者私有的随机数比特串 r_d，提交随机数比特串 r_c，提交信息 D_c，类型组合 t 和评估随机数比特串 r_e，输出结果 x 和评估信息 D_e. 可以理解为该操作手册推荐机制设计者使用评估策略 $\hat{S}_e(A, \cdot, \cdot, \cdot, \cdot, \cdot)$.

2.3 Theoretical Guarantees and Main Result

2.3.1 Implementing, Committing, and Hiding Protocols

实现性指的是如果机制设计者遵循操作手册的指令，那么协议的结果就是以参与者的类型组合为输入的机制运行结果，并且可以通过验证.

2.3 Theoretical Guarantees and Main Result

Definition 2.3.1.1：一个协议操作手册对 $\left((n_c, L_c, \psi_c, n_e, L_e, \psi_e), \left(\hat{S}_c, \hat{S}_e\right)\right)$ 被称作对参数 $\left(L_a, R, L_p\right)$ 具有**实现性**，如果对于任何机制描述 $A \in \mathcal{A}_{L_a, n_e, R}$ 和合法的 $P \in \mathcal{P}_{L_p}$，其证明 A 是 IR 和 DSIC 的，使用了至多 n_e 个随机比特，且其评估时间不超过 R，并且对于任何合法的机制设计者私有随机数比特串 $r_d \in \{0,1\}^{n_d}$，提交随机数比特串 $r_c \in \{0,1\}^{n_c}$，类型组合 $t \in T$ 和评估随机数比特串 $r_e \in \{0,1\}^{n_e}$，如果机制设计者的策略对为 $\left(\hat{S}_c(A, P, \cdot, \cdot), \hat{S}_e(A, \cdot, \cdot, \cdot, \cdot, \cdot)\right)$，以下均成立的话：

1. 验证成功：$\psi_c(r_c, D_c) = \text{True}$ 和 $\psi_e(r_c, D_c, t, r_e, (x, D_e)) = \text{True}$，其中 $D_c = \hat{S}_c(A, P, r_d, r_c)$ 和 $(x, D_e) = \hat{S}_e(A, r_d, r_c, D_c, t, r_e)$.
2. 运行的机制为 M_A，即 $x = M_A(t, r_e)$.

2.3 Theoretical Guarantees and Main Result

提交性指的是无论机制设计者是否遵循操作手册的指令，其提交信息要么唯一对应一个机制，并且该机制的输出就是最终结果，要么某些验证步骤会失败. 更精确的说，虽然机制设计者并没有公开声明该机制，但验证的成功意味着机制设计者实际上已提交遵守该机制.

需要注意的是尽管在提交后运行协议中，机制设计者可以依据 r_c 和 r_d 选择机制，但这是被允许的, 因为这些机制随机数生成的时间节点在参与者披露类型之前, 并且生成过程与参与者的类型无关. 定义核心便是如果验证成功, 该机制就无法依赖于协议运行期间机制设计者获取的其他信息. 唯一的例外在于 r_c 所属的集合太小，机制设计者可以预先针对这些 r_c 制定策略，所以在定义中需要限制.

Definition 2.3.1.2：一个协议 $((n_c, L_c, \psi_c, n_e, L_e, \psi_e)$ 被称作对攻击者的运行时间上限 B 和攻击成功概率 ε 具有**承诺性**，如果对于任何可在时间 B 内计算的机制设计者策略 (S_c, S_e) 以下条件成立：

对每个机制设计者私有随机数比特串 $r_d \in \{0,1\}^{n_d}$，存在一个测度至少为 $1-\varepsilon$ 的集合 $R_c \subset \{0,1\}^{n_c}$，其中元素为提交随机数比特串 r_c，使得对于每个 $r_c \in R_c$，存在一个合法的机制描述 A_{r_c}，其描述的机制是 IR 和 DSIC 的，使用了至多 n_e 个随机比特，且其评估时间不超过 R，使得对于每个类型组合 $t \in T$ 和评估随机数比特串 $r_e \in \{0,1\}^{n_e}$，如果提交验证和评估验证成功，那么所运行的机制为 $M_{A_{r_c}}$.

形式化的说，对于每个 $r_c \in R_c, t \in T, r_e \in \{0,1\}^{n_e}$，如果 $\psi_c(r_c, S_c(r_d, r_c)) = \text{True}$ 和 $\psi_e(r_c, S_c(r_d, r_c), t, r_e, S_e(r_d, r_c, S_c(r_d, r_c), t, r_e)) = \text{True}$，那么有 $S_e(r_d, r_c, S_c(r_d, r_c), t, r_e)_{\text{outcome}} = M_{A_{r_c}}(t, r_e)$.

2.3 Theoretical Guarantees and Main Result

隐匿性是传统协议中不具有的特性, 其核心目标是对于计算能力受限的参与者, 以高概率满足以下条件:

1. 在参与者披露类型之前，其无法通过协议中已经公开的信息来推断机制任何额外的细节，只能获知已经声明的属性（如是 IR 和 DSIC 的，使用不超过 n_e 个随机比特，评估时间不超过 R 等）.
2. 在协议结束后, 参与者仍然无法推断出机制的任何细节, 只能获知已经声明的属性, 以及结果 $M(t, r_e)$.

Definition 2.3.1.3：一个协议操作手册对 $\left((n_c, L_c, \psi_c, n_e, L_e, \psi_e), \left(\hat{S}_c, \hat{S}_e\right)\right)$ 被称作对参数 $\left(L_a, R, L_p\right)$，攻击者运行时间上限 B 和攻击成功概率 ε 具有**隐匿性**，如果对于任何可在时间 B 内计算的区分器 $\mathcal{D}$，每个机制描述 $A_1, A_2 \in \mathcal{A}_{L_a, n_e, R}$，证明 $P_1, P_2 \in \mathcal{P}_{L_p}$，其中 P_i 一个合法的证明，并且证明了 A_i 是 IR 和 DSIC 的，使用了至多 n_e 个随机比特，且其评估时间不超过 R，以下条件成立：

1. （提交具有隐匿性）当使用 $(A, P) = (A_1, P_1)$ 和 $(A, P) = (A_2, P_2)$ 时，区分器 $\mathcal{D}$ 输入 (r_c, D_c) 后输出“这是 A_1”的概率差异不超过 ε. 形式化可以写作

$$\left|\Pr\left[\mathcal{D}^c\left(r_c, D_c^{A_1, P_1}\right) = 1\right] - \Pr\left[\mathcal{D}^c\left(r_c, D_c^{A_2, P_2}\right) = 1\right]\right| \leq \varepsilon,$$

其中机制设计者采用的提交策略为 $\hat{S}_c(A, P, \cdot, \cdot)$.

2. （评估具有隐匿性）对于每个类型组合 $t \in T$ 和评估随机数比特串 $r_e \in \{0,1\}^{n_e}$，如果 $A_1(t, r_e) = A_2(t, r_e)$，那么当使用 $(A, P) = (A_1, P_1)$ 和 $(A, P) = (A_2, P_2)$ 时，区分器 $\mathcal{D}$ 输入 $(r_c, D_c, t, r_e, x, D_e)$ 后输出“这是 A_1”的概率差异不超过 ε. 形式化可以写作

$$\left|\Pr\left[\mathcal{D}^e\left(r_c, D_c^{A_1,P_1}, t, r_e, x_1^A, D_e^{A_1}\right) = 1\right]\right.$$
$$\left. - \Pr\left[\mathcal{D}^e\left(r_c, D_c^{A_2,P_2}, t, r_e, x_2^A, D_e^{A_2}\right) = 1\right]\right| \leq \varepsilon.$$

2.3 Theoretical Guarantees and Main Result

尽管隐匿性在许多场景下都提供了良好的保密性保障, 但其存在一个关键限制: 只适用于固定机制和固定证明，换言之，该定义没有处理依赖提交随机数 r_c 动态选择机制的情况.

> *Example*: 假如 r_c 的生成方式是对某份报纸的标题进行哈希处理，并且提交的机制中的某些物品的价格依赖于某些原材料的价格, 那么是否可以证明报纸标题和物品的价格之间没有关联性? 比如报纸标题宣布开战，那么某些原材料的价格可能会大幅上涨.

所以隐匿性的定义并不能保证机制中**任何部分**的保密性，攻击者可能根据内部的逻辑相关性获取某些信息. 进而引出了关于“强隐匿性”的讨论.

2.3 Theoretical Guarantees and Main Result

隐匿性是基于零知识协议两次运行的不可区分性，而强隐匿性的想法来源于如果这些信息在第三方介导的提交后运行协议中无法获得, 那么也不能在零知识协议中获得, 实现的具体方法依然是不可区分性. 但是第三方介导的提交后运行协议的运行流程和零知识协议的运行流程相差很大，比如第三方介导的提交后运行协议中不存在 r_c, D_c 和 D_e，这很容易就会导致不可区分性失效.

为了克服这一问题，从现代密码学中引入了**模拟**的概念，为第三方介导的提交后运行协议补充一个可高效计算的算法, 称为模拟器, 其不能访问机制描述, 但是可以用于生成 r_c, D_c 和 D_e. 如果该模拟器所生成的 r_c, D_c 和 D_e 让那些计算能力受限的参与者无法区分第三方介导的提交后运行协议产生的记录 $(r_c, D_c, t, r_e, x, D_e)$ 和零知识协议产生的记录 $(r_c, D_c, t, r_e, x, D_e)$，那么就达到了先前讨论的目标.

2.3 Theoretical Guarantees and Main Result

因为希望即使所有参与者联合都无法区分第三方介导的提交后运行协议和零知识协议，而参与者可以联合决定的便是类型组合 t，定义参与者的联合策略为 $\hat{t}:\{0,1\}^{n_c}\times\{0,1\}^{L_c}\to T$，以 r_c 和 D_c 为输入，输出类型组合 t. 并且，即使机制描述 A 的选择依赖于 r_c，也希望其能保持隐匿性，所以定义机制选择策略为 $\left(\hat{A}:\{0,1\}^{n_c}\to\mathcal{A}_{L_a,n_e,R},\hat{P}:\{0,1\}^{n_c}\to\mathcal{P}_{L_p}\right)$，使得其满足相容性条件：对于每个 $r_c\in\{0,1\}^{n_c}$，$\hat{P}(r_c)$ 是一个合法的证明，并且 $\hat{A}(r_c)$ 是一个合法的机制描述，证明 $\hat{P}(r_c)$ 证明了 $\hat{A}(r_c)$ 是 IR 和 DSIC 的，使用了至多 n_e 个随机比特，且其评估时间不超过 R. 更进一步，希望即使一个技术参数的 r_e 的生成方式与 r_c 相关，仍然能够保持隐匿性，所以定义评估随机数生成策略为 $\hat{n}_e:\{0,1\}^{n_c}\to\{0,1\}^{n_e}$.

2.3 Theoretical Guarantees and Main Result

带有模拟器的第三方介导的提交后运行协议的流程如下:

1. 模拟器生成 $r'_c \in \{0,1\}^{n_c}$ 和 $D'_c \in \{0,1\}^{L_c}$.
2. 机制设计者将机制描述 $\hat{A}(r'_c)$ 和证明 $\hat{P}(r'_c)$ 发送给可信第三方.
3. 参与者类型组合设置为 $t' = \hat{t}(r'_c, D'_c)$，评估随机数设置为 $r'_e = \hat{n}_e(r'_c)$. 基于 t', r'_e，机制设计者宣布结果 $x' = M_{\hat{A}(r'_c)}(t', r'_e)$
4. 模拟器生成 $D'_e \in \{0,1\}^{L_e}$.
5. 模拟的记录为 $(r'_c, D'_c, t', r'_e, x', D'_e)$.

2.3 Theoretical Guarantees and Main Result

区分器定义为 $\mathcal{D}^c : \{0,1\}^{n_c} \times \{0,1\}^{L_c} \times T \times \{0,1\}^{n_e} \times X \times \{0,1\}^{L_e} \to \{\text{True}, \text{False}\}$，其输入为记录 $(r_c, D_c, t, r_e, x, D_e)$，输出为 True 或 False，表示其是否认为记录是由第三方介导的提交后运行协议生成的.

强隐匿性的通俗理解便是无论使用怎样的策略 $\left(\hat{A}, \hat{P}\right), \hat{t}, \hat{n}_e$，如果机制设计者遵照了操作手册的指令，那么不存在区分器可以 $\frac{1}{2} + \varepsilon$ 的概率判断记录是否来自第三方介导的提交后运行协议，其中 ε 不可忽略.

敌手定义为四元组 $\left(\left(\hat{A}, \hat{P}\right), \hat{t}, \hat{n}_e, \mathcal{D}\right)$.

Definition 2.3.1.4：一个协议操作手册对 $\left((n_c, L_c, \psi_c, n_e, L_e, \psi_e), \left(\hat{S}_c, \hat{S}_e\right)\right)$ 被称作对参数 $\left(L_a, R, L_p\right)$，攻击者运行时间上限 B，最大额外运行时间 B'，攻击成功概率 ε 具有**强隐匿性**，如果对于每个可在时间 B 内计算的敌手 $\left(\left(\hat{A}, \hat{P}\right), \hat{t}, \hat{n}_e, \mathcal{D}\right)$，以下概率相差不超过 ε：

1. 区分器认为一个模拟器生成的记录 $(r'_c, D'_c, t', r'_e, x', D'_e)$ 来自第三方介导的提交后运行协议的概率.
2. 区分器认为一个零知识协议生成的记录 $(r_c, D_c, t, r_e, x, D_e)$ 来自第三方介导的提交后运行协议的概率，其中机制设计者遵循了操作手册的指令，即 $D_c = \hat{S}_c\left(\hat{A}(r_c), \hat{P}(r_c), r_d, r_c\right)$，$t = \hat{t}(r_c, D_c)$，$r_e = \hat{n}_e(r_c)$，$(x, D_e) = \hat{S}_e\left(\hat{A}(r_c), r_d, r_c, D_c, t, r_e\right)$.

2.3 Theoretical Guarantees and Main Result

2.3.2 Feasibly Computable Protocols Catalogs

这里的可行计算性是协议族上的定义，旨在处理协议的渐进可追踪性. 具体来说，是定义一类提交后运行协议族，称为**协议目录**，其对于每个可能的技术参数 $\sigma = (L_a, n_e, R, L_p, B, \varepsilon)$，都存在一个对应的协议操作手册对. 在此基础上将实现性、承诺性和隐匿性扩展到协议目录上.

2.3 Theoretical Guarantees and Main Result

Definition 2.3.2.1: **提交后运行协议目录** 指的是一族协议操作手册对 $\left((n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma), \left(\hat{S}_c^\sigma, \hat{S}_e^\sigma\right)\right)_{\sigma\in\Sigma}$，对任意 $\sigma = \left(L_a, n_e, R, L_p, B, \varepsilon\right) \in \Sigma$ 都有 $n_e^\sigma = n_e$.

1. 如果对于每个 $\sigma \in \Sigma$，协议操作手册对 $\left((n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma), \left(\hat{S}_c^\sigma, \hat{S}_e^\sigma\right)\right)$ 对于参数 $\left(L_a, R, L_p\right)$ 具有实现性，那么该协议目录具有**实现性**.
2. 如果对于每个 $\sigma \in \Sigma$，协议 $(n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma)$ 对攻击者的运行时间上限 B 和攻击成功概率 ε 具有承诺性，那么该协议目录具有**承诺性**.
3. 如果对于每个 $\sigma \in \Sigma$，协议 $(n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma)$ 对参数 $\left(L_a, R, L_p\right)$，攻击者的运行时间上限 B 和攻击成功概率 ε 具有隐匿性，那么该协议目录具有**隐匿性**.

2.3 Theoretical Guarantees and Main Result

接下来正式定义一个协议目录是可行计算的，即其信息大小和计算需求在渐进意义上和传统协议上本质上是相同的（差异至多为对数因子或次多项式因子）.

Definition 2.3.2.2: 称 $s^\sigma = s^{L_a, n_e, R, L_p, B, \varepsilon} \in \mathbb{N}$ 在基准量 $C(\sigma)$ 上是**本质线性的**，如果存在一个多对数函数 $\mathrm{polylog}(\cdot,\cdot,\cdot)$，以及一个次多项式函数 $\mathrm{subpoly}(\cdot,\cdot)$，使得对任意 $\sigma \in \Sigma$ 有 $s^\sigma \le C(\sigma) \cdot \mathrm{polylog}\big(L_a, R, L_p\big) \cdot \mathrm{subpoly}\big(B, \frac{1}{\varepsilon}\big)$，其中 $\mathrm{polylog}()$ 和 $\mathrm{subpoly}()$ 不能依赖于 σ. 如果 s^σ 在基准量 1 上是本质线性的，那么称 s^σ 是**本质常数的**.

Definition 2.3.2.3: 一个提交后运行协议目录 $\left((n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma), \left(\hat{S}_c^\sigma, \hat{S}_e^\sigma\right)\right)_{\sigma\in\Sigma}$ 是**可行计算的**，如果以下条件成立：

1. 提交随机数 n_c^σ 是本质常数的;
2. 提交信息长度和运行时间在 $L_a + L_p$ 上是本质线性的;
   a. 提交信息长度 L_c^σ 在 $L_a + L_p$ 上是本质线性的;
   b. 提交验证器 ψ_c^σ 的最大运行时间在 $L_a + L_p$ 上是本质线性的;
   c. 提交操作手册 $\hat{S}_c^\sigma$ 的最大运行时间在 $L_a + L_p$ 上是本质线性的.
3. 评估信息长度和运行时间在 R 上是本质线性的;
   a. 评估信息长度 L_e^σ 在 R 上是本质线性的;
   b. 评估验证器 ψ_e^σ 的最大运行时间在 R 上是本质线性的;
   c. 评估操作手册 $\hat{S}_e^\sigma$ 的最大运行时间在 R 上是本质线性的.
4. 机制设计者私有随机数 n_d 在 $L_a + R + L_p$ 上是本质常数的.

2.3 Theoretical Guarantees and Main Result

2.3.3 Commit-then-Prove Protocols

提交后证明协议包含了一个证明器和一个或多个验证器. 协议具有两个阶段: 提交阶段和证明阶段.

1. 提交阶段，证明器提交一些秘密信息 w，该行为不可逆.
2. 证明阶段，证明器给出一个关于 w 的宣称 ϕ，即证明器向验证器保证其知道一个使得 $\rho(w, \phi, P)$ 成立的证明 P，其中 ρ 是一个被用来验证 P 是一个 ϕ 关于 w 的证明的关系.

以上要求可以通过理想函数 $\mathcal{F}_{\text{ctp}}$ 规范化，其捕获期望的结果以及保密性质.

2.3 Theoretical Guarantees and Main Result

$\mathcal{F}_{\text{ctp}}$ 由关系 $\rho(\cdot,\cdot,\cdot)$，证明器 C，以及以下 4 个程序组成.

1. `ChooseCommitmentToken`：不接受输入，输出一个提交令牌 c.
2. `ChooseProofToken`：接受 c 和 ϕ 作为输入，输出一个证明令牌 p.
3. `ChooseCommittedMessage`：接受 c', ϕ', p'，输出一个信息 w' 用来作为提交 c' 的秘密信息.
4. `ChooseProof`：接受 c', ϕ', p'，输出一个证明 P'，使得 $\rho(w', \phi', P')$ 成立或不成立.

2.3 Theoretical Guarantees and Main Result

$\mathcal{F}_{\text{ctp}}$ 的运行流程如下：

1. 提交阶段：接收来自证明器 C 的 $(\texttt{Commit}, w)$ 作为输入，如果 $(\texttt{Commit})$ 信息先前接收过，则对 C 返回 $(\texttt{False})$；否则，令 $c \leftarrow \texttt{ChooseCommitmentToken}()$，如果存在记录 $(\cdot, c)$，返回 $(\texttt{Abort})$；否则，记录 (w, c)，并输出 $(\texttt{Commitment}, c)$ 给 C.
2. 证明阶段：接收来自证明器 C 的 $(\texttt{Prove}, \phi, P)$ 作为输入，没有接收过 $(\texttt{Commit})$ 信息则对 C 返回 $(\texttt{False})$；否则，令 w 为 $(\texttt{Commit})$ 阶段中所接收的，c 为所返回的提交令牌. 如果 $\rho(w, \phi, P)$ 不成立，则对 C 返回 $(\texttt{False})$；否则，令 $p \leftarrow \texttt{ChooseProofToken}(c, \phi)$，记录 $(c, \phi, p, \texttt{True})$ 并输出 $(\texttt{Proof}, p)$ 给 C.
3. 证明确认阶段：接收来自任何一方的输入 $(\texttt{Verify}, c', \phi', p')$：
   a. 如果存在记录 (c'', ϕ'', p'', τ) 满足 $(c'', \phi'', p'') = (c', \phi', p')$，则对该方返回 $(\texttt{Verified}, \tau)$.

b. 如果没有形如 $(\cdot, c')$ 的记录，则选择 $w' \leftarrow \texttt{ChooseCommittedMessage}(c', \phi', p')$，记录 (w', c').

c. 令 $P' \leftarrow \texttt{ChooseProof}(c', \phi', p')$，如果 $\rho(w', \phi', P')$ 成立，则令 $\tau = \texttt{True}$，否则令 $\tau = \texttt{False}$，记录 (c', ϕ', p', τ) 并对该方返回 $(\texttt{Verified}, \tau)$.

2.3 Theoretical Guarantees and Main Result

为了简化提交后证明协议, 限定提交后证明协议中包含的所有程序都满足输出正则性, 即程序的输出长度固定，并且只依赖于安全参数.

Definition 2.3.3.1：**提交后证明协议**是一个满足输出正则性的程序四元组 $\pi = (\text{drawrefstring}, \text{commit}, \text{prove}, \text{verify})$，依赖于安全参数 λ. π 如果满足以下两个要求，则可称为对 ρ, n_w, n_ϕ, n_p 是安全的，其中 ρ 是一个关系，n_w 是提交信息的最大长度，n_ϕ 是宣称的最大长度，n_p 是证明的最大长度.

首先是第一个要求，如果存在

1. 一个参数传递函数：$\tau\mathbb{N} \times \mathbb{R}_{>0} \to \mathbb{N}$，使得 $\lambda = \tau(B, \varepsilon)$ 是 B 和 $\frac{1}{\varepsilon}$ 的次多项式函数.

2. 一个模拟器 $\mathcal{S}$，给定安全参数 λ 和比特串 $r_{\mathcal{S}} \in \{0,1\}^{n_{\mathcal{S}}}$ 作为内置随机源，生成如下 5 个程序：

   a. `SimulateRefString`：不接受输入，输出一个比特串 s_{sim} 用作模拟参考字符串. 要求如果 $r_{\mathcal{S}}$ 是随机均匀抽取的话，s_{sim} 的分布要与 drawrefstring (λ) 的输出分布相同.

   b. `ChooseCommitmentToken`, `ChooseProofToken`, `ChooseCommittedMessage`, `ChooseProof`：用于 $\mathcal{F}_{\text{ctp}}$ 的程序.

$\mathcal{S}$ 以及以上 5 个程序的运行时间需要是 $\lambda, n_w, n_\phi, n_p$ 的多项式函数.

2.3 Theoretical Guarantees and Main Result

而为了定义第二个要求，需要首先定义一个公正的挑战者监督的敌手程序 $\mathcal{A}$ 的挑战.

2.3.3.1 $\mathcal{F}_{\text{ctp}}$ 分离挑战

1. 挑战者随机选择比特 $b \in \{0, 1\}$
2. 如果 $b = 0$，那么挑战者运行敌手 $\mathcal{A}$，使得其与 π 进行交互：
   a. 挑战者采样参考串 $s \leftarrow \text{drawrefstring}\ (\lambda)$，并将其发送给 $\mathcal{A}$.
   b. $\mathcal{A}$ 可以重复进行以下三种操作：
      i. $\mathcal{A}$ 向挑战者发送 $(\texttt{Commit}, w)$，如果一个 `(Commit)` 信息先前发送给挑战者，则挑战者返回 `(False)`；否则，挑战者抽取随机串 r，计算 $(c, \mu) = \text{commit}\ (\lambda, w, r, s)$，记录状态 μ 并将 c 返回给 $\mathcal{A}$.

ii. $\mathcal{A}$ 向挑战者发送 $(\texttt{Prove}, \phi, P)$，如果没有先前的 $(\texttt{Commit})$ 信息，则挑战者返回 $(\texttt{False})$；否则，挑战者抽取新的随机串 r，计算 $(p, \mu') = \text{prove}\ (\lambda, \mu, \phi, P, r)$，更新状态 $\mu \leftarrow \mu'$ 并且将 p 返回给 $\mathcal{A}$.

iii. $\mathcal{A}$ 向挑战者发送 $(\texttt{Verify}, c', \phi', p')$，挑战者返回 $\text{verify}\ (\lambda, s, c', \phi', p')$ 的结果.

3. 如果 $b = 1$，那么挑战者运行敌手 $\mathcal{A}$，使得其与 $\mathcal{F}_{\text{ctp}}^{\rho}$ 和 $\mathcal{S}$ 交互.

   a. 挑战者随机均匀抽取 $r_{\mathcal{S}} \in \{0,1\}^{n_{\mathcal{S}}}$，并且运行 $\mathcal{S} = \mathcal{S}(r_{\mathcal{S}}, \lambda)$ 生成 5 个程序. 接下来挑战者依据后 4 个程序生成一个 $\mathcal{F}_{\text{ctp}}^{\rho}$ 的实例，并且运行 $\texttt{SimulateRefString}$ 生成模拟参考字符串 s_{sim} 发送给 $\mathcal{A}$.

   b. $\mathcal{A}$ 可以重复进行以下三种操作：发送 $(\texttt{Commit}, w)$, $(\texttt{Prove}, \phi, P)$ 或 $(\texttt{Verify}, c', \phi', p')$，挑战者必须依据先前实例化的 $\mathcal{F}_{\text{ctp}}^{\rho}$ 的信息，并且将其返回值发送给 $\mathcal{A}$.

4. 最终 $\mathcal{A}$ 输出 b'，如果 $b' = b$，则称 $\mathcal{A}$ 挑战胜利.

2.3 Theoretical Guarantees and Main Result

如果 $\mathcal{A}$ 最终以 $\frac{1}{2}+\varepsilon$ 获胜，则称 ε 为 $\mathcal{A}$ 的分离优势.

第二个要求便是对于任意运行时间上界 $B \in \mathbb{N}$ 以及任意分离优势 $\varepsilon > 0$，如果每个运行时间至多为 $B \cdot C_\pi(\lambda)$ 的敌手的分离优势不超过 ε，则称 π 是**安全的**，其中 $\lambda = \tau(B, \varepsilon)$，$C_\pi(\lambda)$ 是 commit, prove, verify 的最大运行时间之和.

2.3 Theoretical Guarantees and Main Result

1. 完美完备性：如果敌手 $\mathcal{A}$ 所有发送的 $(\texttt{Verify}, c, \phi, p)$ 信息中的 c 都是先前挑战者响应 $(\texttt{Commit}, w)$ 信息中返回的值，并且 p 是先前挑战者响应 $(\texttt{Prove}, \phi, P)$ 信息中返回的值，那么称 $\mathcal{A}$ 为**良性**敌手. 而提交后证明协议 π 是**完美完备的**，如果存在模拟器 $\mathcal{S}_B$，使得对于每个 λ 和每个良性敌手 $\mathcal{A}$，其分离优势为 0.

2. 均匀参考字符串：如果 π 的参考字符串生成程序 `"drawrefstring"` 是一个均匀抽取的分布，那么称 π 是**均匀参考字符串的**.

3. 可行计算：如果
   a. drawrefstring 的运行时间是 λ 的多项式函数.
   b. commit 的运行时间是 λ, n_w 的多项式函数.
   c. prove, verify 的运行时间是 $n_w, n_\phi, n_p, \lambda$ 以及 ρ 的运行时间的多项式函数.

则称 π 是**可行计算的**.

Theorem 2.3.3.1.1：在广泛认可的计算不可行假设下，对任何多项式时间可计算的关系 ρ，存在一个安全的可行计算的提交后证明协议 π，并且其具有完美完备性和均匀参考字符串性质.

2.3 Theoretical Guarantees and Main Result

2.3.4 Main Result

Theorem 2.3.4.1: 在广泛认可的计算不可行假设下，存在一个满足实现性、承诺性，隐匿性和可行计算性的直接披露提交后运行协议目录.

2.3.5 Succinct Communication and Verification

但问题仍然存在，传统情况下，如果机制本身没有先验结构，那么机制描述的长度可能比机制已知时运行所需传递的信息的指数级更长. 即使在零知识协议中只需要传输有关协议的性质，但是其证明可能需要依赖于整个机制的输入和输出过程，使得传输的信息量仍然可能与传统协议相近.

但接下来可以证明, 可以构造一个满足先前所有的要求的零知识协议, 并且在通信效率上实现指数级的提升, 使得通信需求降低至与传统情况下机制已知时的同等水平.

另一个差异在于验证方式, 因为参与者无法知晓机制, 所以需要全新的验证方式, 这是否能带来参与者计算量的减少? 答案依然是肯定的, 参与者可以在低于自行计算的成本下, 无信任地验证结果正确性.

Definition 2.3.5.1: 一个直接披露提交后运行协议目录 $\left((n_c^\sigma, L_c^\sigma, \psi_c^\sigma, n_e^\sigma, L_e^\sigma, \psi_e^\sigma), \left(\hat{S}_c^\sigma, \hat{S}_e^\sigma\right)\right)_{\sigma\in\Sigma}$ 被称为**简明的**, 如果所有的信息长度和验证器运行时间都是本质常数的, 即:

1. 提交信息长度 L_c^σ 和评估信息长度 L_e^σ 都是本质常数的.
2. 提交验证器 ψ_c^σ 和评估验证器 ψ_e^σ 的最大运行时间都是本质常数的.

简明协议目录的参与者会认为计算速度相较于传统协议实现了指数级加速，但是机制设计者的角度看来所需的计算复杂度仍然与传统协议相当.

Theorem 2.3.5.1：在随机预言机模型下，存在一个满足实现性、承诺性、隐匿性和可行计算性的简明直接披露提交后运行协议目录.

2.3 Theoretical Guarantees and Main Result

提交后证明协议也有其对应的简明性.

Definition 2.3.5.2：一个提交后证明协议 π 被称为**简明的**，如果其是可行计算的，并且 verify 的运行时间是关于 n_w, n_P 以及 ρ 的运行时间的多对数函数.

Theorem 2.3.5.2：随机预言机模型下，对于任意多项式时间可计算的关系 ρ，存在一个简明的提交后证明协议 π，使得其满足完美完备性、均匀参考字符串性质.

Extensions

3.1 Private Actions and Moral Hazard: Zero-Knowledge Contracts

本框架亦可扩展至含私有行为的机制，例如合约.

传统合约协议流程包含以下步骤：

1. 提交阶段：机制设计者需要证明激励属性，也就是明确参与者私有执行的最优努力水平.
2. 行为/回报阶段：替代直接披露阶段，参与者进行私有执行努力，公开实现随机回报 t.
3. 结果阶段：基于 t 以及可能的随机数 r_e 计算结果.

所以提交后运行协议需要进行一些修改：

1. 激励证明变更：需证明代理方的最优努力水平，而非类型披露的真实性；
2. 行为/回报阶段变更：t 表示行动产生的随机回报，而非类型披露.

3.2 Multipart Outcomes: To Each Their Own

在某些场景中，结果 $x \in X$ 可以自然分解为不同参与者关心的独立部分. 给定结果 x，定义 $v_i(x)$ 为参与者 i 应当获得的结果部分，称为 i 的结果视图，其应当至少包含参与者 i 的收益相关信息，即如果 $v_i(x) = v_i(x')$，则参与者 i 的效用在 x 和 x' 中相同.

提交后运行协议需要进行扩展：

1. 评估阶段：机制设计者依据 t 和 r_e 向每个挑战者私下发送 $\overline{v}_i$ 以及评估信息 $D_e^i \in \{0,1\}^{L_e}$.
2. 评估验证阶段：每个参与者 i 计算验证函数 $\psi_e^i(r_c, D_c, t, r_e, \overline{v}_i, D_e^i)$ 判断其是否为真.

如果验证通过，参与者便可以确信 $\overline{v}_i = v_i(M_A(t, r_e))$.

3.3 Hiding Both the Mechanism and the Types

设存在物品集 Y，结果 $x \in X$ 形如 $(y; s^1, ..., s^n)$ 表明结果 y 以及每个参与者的转移费用 s^i. 此外，希望参与者的效用函数关于其自身的转移费用 s^i 时线性的，并且不依赖于其他参与者的转移费用. 即 $\forall i, t_i \in T_i$ 是一个估价函数 $t_i : Y \to \mathbb{R}_{\geq 0}$ 使得 $u_i(t_i, x) = t_i(y) - s^i$.

机制设计目标希望依据权重向量 $w = (w_1, ..., w_n)$ 最大化加权社会福利，即选择 $y \in \operatorname{argmax}_{y \in Y} \sum_{i=1}^{n} w_i t_i(y)$. 为了激励真实报价，采用无主元规则的 Groves 机制，即设置 $s^i = -\frac{1}{w^i} \sum_{j \neq i} w_j t_j(y)$，将这一机制记为 M_w.

3.3 Hiding Both the Mechanism and the Types

但以上机制很容易遭受攻击，一旦攻击者掌握类型组合 t 以及机制结果 $M_w(t)$，其便可以得到权重向量 w.

Theorem 3.3.1：对任意 $t \in T$，除非 $\forall i = 1, ..., n; \forall y \in Y$ 均有 $t_i(y) = 0$，则 $\forall w \in W$ 总是存在函数 $E_t : X \to W$ 使得 $E_t(M_w(t)) = w$.

3.3 Hiding Both the Mechanism and the Types

所以，希望参与者的类型也具有**隐匿性**，首先考虑该如何增强第三方介导的提交后运行协议.

1. 提交阶段和提交验证阶段不变.
2. 类型报告阶段：每个参与者向第三方秘密提供其类型，类型组合记为 t.
3. 评估阶段：第三方抽取随机数 r_e，长度为 n_e. 评估 $x = M_A(t, r_e)$，并且公开结果 x.

3.3 Hiding Both the Mechanism and the Types

在此基础上考虑如何利用零知识协议来增强隐匿性，还需要引入安全多方计算这一工具. 提交阶段没有变化，主要变化在于评估阶段.

1. 评估阶段：每个参与者以及机制设计者使用安全多方计算来联合评估 $x = M_A(t, r_e)$. 每当机制设计者发送任何信息，其也必须附带对应的零知识证明，证明信息与其提交的机制描述是一致的.

但是安全多方计算带来的计算开销过于庞大使得实际应用中难以承受，这也是需要解决的问题.