基于配对的非交互论证的大小

Jens Groth

University College London, UK
j.groth@ucl.ac.uk

摘要

非交互式参数使证明者能够让验证者相信一个声明（statement）是真实的。近来，在构建高效的小尺寸和低验证复杂度的非交互论证，即简洁非交互论证（SNARGs）和简洁非交互知识论证（SNARKs）方面，理论和实践都取得了重要进展。

SNARG 的许多结构都依赖于密码学中的配对（pairing）。在这些结构中，证明由一些群元素组成，验证时检查配对乘积方程是否成立。在本文中解决的问题是基于配对的 SNARG 效率。

本文的第一个贡献是用于算术电路可满足性基于配对（预处理）的 SNARK，它是一种 NP 完全语言。在我们的 SNARK 中，我们使用非对称配对来提高效率，证明只有 3 个群元素，验证使用 3 个配对检查单个配对乘积方程。同时该 SNARK 是零知识，不会透露证明者证明中任何证据（witness）的信息。

第二个贡献回答了 Bitansky、Chiesa、Ishai、Ostrovsky 和 Paneth (TCC 2013) 的一个开放问题，表明 2-move 线性交互式证明不能具有线性决策过程。由此可见，如果证明者和验证者使用通用非对称双线性群运算的 SNARG，其证明不能由单个群元素组成。这给出了基于配对的 SNARG 的第一个下界。这个下界是否可以扩展，排除只有 2 个群元素的 SNARG，仍然是一个开放性问题，这将证明 3 个群元素构造的最优性。

关键词: SNARKs, 非交互零知识论证, 线性交互证明, 二次算术程序, 双线性群

1 引言

Goldwasser, Micali and Rackoff [GMR89] 介绍了零知识证明 ZKP，允许证明者在不暴露任何额外信息的情况下让验证者相信一个声明。有如下 3 个性质：

完备性（Completeness）：给定一个 statement 和 witness，诚实证明者能正确证明使验证者相信。

合理性（Soundness）：不诚实证明者不能让验证者相信错误的 statement。

零知识性（Zero-knowledge）：提供的证明除了揭示 statement 是正确的，而不泄露证明者的 witness。

Blum, Feldman and Micali [BFM88] 在公共参考字符串（CRS）中模型中拓展了非交互式零知识证明（NIZK）。NIZK 证明在构建非交互式密码学方案非常有用，如数字签名和 CCA 安全的公钥加密方案。

通信的开销是零知识证明（ZKP）中的重要表现参数。Kilian [Kil92] 中证明者发送比要证明的 statement 更少的比特数，从而给出了第一个亚线性通信的零知识论证。Micali [Mic00] 中让通信有效论证中的证明者用密码函数计算验证者的挑战，提出了亚线性大小论证，正如 Kilian [Kil95] 所述，当交互式论证是公平掷币和零知识时，这会产生亚线性大小的 NIZK 证明。

Groth, Ostrovsky 和 Sahai[GOS12, GOS06, Gro06, GS12] 中介绍了基于配对的 NIZK 证明，实现了第一个标准模型下线性大小的证明。Groth[Gro10] 将这些技术与交互式零知识论证 [Gro09] 的想法组合起来，给出了第一个常数证明大小的 NIZK 论证系统。Lipmaa [Lip12] 用基于 progression-free sets 的替代结构来减少公共参考字符串的大小。

Groth 的常数证明大小 NIZK 论证系统基于一系列多项式的构造，并用配对的方式有效地验证这些等式。Gennaro, Gentry, Parno 和 Raykova[GGPR13] 用 Lagrange 插值构造多项式方程，从而提出了一个基于配对的 NIZK 论证系统，其公共参考字符串大小与 statement 和 witness 的大小成正比。他们给出了两种多项式等式：用来证明布尔电路可满足性的 QSP（quadratic span programs）和证明算数电路可满足性的 QAP（quadratic arithmetic programs）。Lipmaa [Lip13] 提出了使用纠错码的更有效的 QSP，Danezis, Fournet, Groth 和 Kohlweiss[DFGK14] 将 QSP 改进为 SSP（square span programs），给出了由 4 个群元素组成的 NIZK 论证系统，用于证明布尔电路可满足性。

在实现工作方面遵循了上述理论进展 [PHGR13, BSCG+13, BFR+13, BSCTV14b, KPP+14, BBFR15, CTV15, WSR+15, CFH+15, SVV16]。最有效的实现改进了 Gennaro[GGPR13] 等人的 QAP 方法，把它与编译器相结合，生成与要证明的 statement 等效且适合的 QAP；libsnark[BSCTV14b, BSCG+14] 还包括一个基于 [DFGK14] 的 NIZK 论证系统。

构建有效非交互式论证系统的一个动机是可验证计算。客户端将复杂的计算任务外包给云服务器并接收计算结果。为了让客户端相信计算是正确的，服务器在结果中包含一个非交互、关于正确性的论证。验证者没有多余的计算资源，在论证紧凑且计算量少的情况下才能够进行有效验证，即论证是一个简洁的非交互式论证（SNARG）或一个简洁的非交互式知识论证（SNARK）。虽然基于配对的 SNARG 对验证者来说是有效的，但证明者的计算开销仍然太高，无法保证能够在外包计算中使用 [WB15, Wal15]，因此需要进一步提高效率。目前 zkSNARK 在证明有关私人数据的 statement 中已经有了应用，如 zkSNARK 是虚拟货币方案 Pinocchio coin[DFKP13] 和 Zerocash[SCG+14] 的关键部分。

在基于配对的 NIZK 论证系统发展的同时，在 SNARK 方面也有一些有趣的工作。Gentry 和 Wichs[GW11] 表明 SNARGs 必须依赖于不可证伪的假设，并且 Bitansky 等人 [BCCT12] 证明了当且仅当存在可提取的抗碰撞哈希函数时，指定验证者的 SNARK 才存在。在效率方面则是一系列预处理 SNARK 如何组成的工作 [Val08, BCCT13, BSCTV14a]，用带有长 CRS 的预处理 SNARK

构建一个具有短 CRS、完全简洁的 SNARK。

Bitansky 等人 [BCI+13] 给出了依赖于字段元素线性编码的 SNARKs 抽象模型。模型的信息论框架称为线性交互证明 (linear interactive proofs, LIP) 的证明系统，证明者只能使用线性运算来计算他的消息。他们使用配对的技术，给出了 2-move 的 LIP 到可公开验证的 SNARK 的通用转换，或使用加法同态加密技术转换到指定验证者。

1.1 我们的贡献

简洁的 NIZK 我们为算术电路可满足性构造了一个 NIZK 论证系统，其中证明仅包含 3 个群元素。除了证明大小，证明也很容易验证。验证者只需要计算与 statement 大小成正比的指数，并检查一个只有 3 个配对的配对乘积方程。我们的构造可以用任何类型的配对实例化，包括最有效的配对类型 III 配对。

该论证具有完美的完备性和完美的零知识性。为了合理性，我们采取通用双线性群模型 [Sho97, Nec94] 中的安全证明来获得最佳性能。Gentry 和 Wichs[GW11] 部分证明了这种方法是合理的，并且基于标准的可证伪假设排除了 SNARG。然而，在 Abe、Groth、Ohkubo 和 Tibouchi [AGOT14] 之后，我们通过证明我们的构造在对称配对设置中的安全性来抵抗密码分析。为了获得最佳效率，在非对称设置中使用我们的 NIZK 参数是有意义的，但是，通过在对称设置中提供安全证明，我们可以获得额外的安全性：即使密码分析的进步在群间产生了未知的高效可计算同构，但这并不一定导致我们的方案不可用。因此，我们有一个统一的 NIZK 参数，可以用任何类型的配对实例化，从而产生最佳效率和最佳通用双线性群。

我们在表 1 和表 2 中分别给出了布尔电路可满足性和算数电路可满足性的性能比较，包括 CRS 大小，证明大小，证明者的计算，验证者的计算以及验证过程中使用的配对方程数量。我们在所有效率参数上的表现都优于最先进的技术。

表 1: ℓ 比特 statement、m 条电路和 n 个 2 输入逻辑门布尔电路可满足性，不同方案的比较。符号：$\mathbb{G}$ 表示群元素，M 表示乘法，E 表示幂运算，P 表示配对。CRS 大小包括 $\mathbb{G}_1$ 中的 $m+2n$ 个元素和 $\mathbb{G}_2$ 上的 n 个元素中获得的但我们选择在 CRS 中包含一些预先计算的值以减少证明者的计算，请参阅第 3.2 节。

	CRS size	Proof size	Prover comp.	Verifier comp.	PPE
[DFGK14]	$2m+n-2\ell\mathbb{G}_1, m+n-\ell\mathbb{G}_2$	$3\mathbb{G}_1, 1\mathbb{G}_2$	$m+n-\ell E_1$	$\ell M_1, 6P$	3
This work	$3m+n\mathbb{G}_1, m\mathbb{G}_2$	$2\mathbb{G}_1, 1\mathbb{G}_2$	nE_1	$\ell M_1, 3P$	1

在这两个比较中，线路的数量都超过了门的数量，$m \geq n$，因为每个门都有一条输出线路。对于典型情况，我们希望 statement 的大小 ℓ 比 m 和 n 小。这两个表排除了我们给出的证明大小。在布尔电路可满足性的情况下，我们考虑任意 2 输入逻辑门。在算术电路可满足性的情况下，我们使用 2 输入乘法门，其中每个输入可以是其他线的加权和。我们假设每个乘法门输入取决于恒定数

表 2: ℓ 元素的 statement、m 条电路、n 个乘法门算术电路可满足性，不同方案的比较。符号：$\mathbb{G}$ 表示群元素，E 表示幂运算，P 表示配对。我们比较前两行中的对称配对和最后两行中的非对称配对。

	CRS size	Proof size	Prover comp.	Verifier comp.	PPE
[PHGR13]	$7m+n-2\ell\mathbb{G}$	$8\mathbb{G}$	$7m+n-2\ell E$	$\ell E, 11P$	5
This work	$m+2n\mathbb{G}$	$3\mathbb{G}$	$m+3n-2\ell E$	$\ell E, 3P$	1
[BSCTV14a]	$6m+n-\ell\mathbb{G}_1, m\mathbb{G}_2$	$7\mathbb{G}_1, 1\mathbb{G}_2$	$6m+n-\ell E_1, mE_2$	$\ell E, 12P$	5
This work	$m+2n\mathbb{G}_1, n\mathbb{G}_2$	$2\mathbb{G}_1, 1\mathbb{G}_2$	$m+3n-\ell E_1, nE_2$	$\ell E, 3P$	1

量的线路，否则评估关系本身的成本可能会超过后续证明生成的成本。

我们注意到 [PHGR13] 使用的是对称双线性群 $\mathbb{G}_1 = \mathbb{G}_2$，因此能够与我们方案的一个对称双线性群实例进行比较，我们方案在 CRS 中减少了 n 个元素。然而，在他们的系统实现 Pinocchio 中，非对称配对有更好的效率。切换到非对称配对只需要稍作修改，如 [BSCTV14a] 用于此类 SNARK 的规范，已在 libsnark 库中实现。

大小问题（SIZE MATTERS.）虽然证明大小减少到 3 个群元素且证明时间短，我们想强调当组合 SNARK 时有一点很重要。[BCCT13, BSCTV14a] 中具有长 CRS 预处理的 SNARKs 能够用来组合实现短 CRS[1]完整简洁的 SNARK[1]。这个转换将 statement 分成更小的部分，证明每个部分本身是正确的，并递归地构造其他证明的知识证明，这些证明共同表明这些部分是正确的并且可以组合在一起。在证明的递归构造中，当证明很小且易于验证时，这是特别有用的，因为" 存在满足验证方程的证明....." 的 statement 本身变得很小。我们利用证明者的较低计算量和递归组合中的语句 statement 的事实，因此对 SNARK 有更有效的验证程序。Chiesa 和 Virza[CV16] 报告显示，在 [BSCTV14a] 的实现中使用我们的 SNARK 可以提高 4-5 倍的速度。

技术（TECHNIQUE.）文献中所有基于双线性对的 SNARK 都遵循一个通用范式，其中证明者用通用群操作计算多个群元素，验证者用多个配对的乘积等式验证证明。Bitansky 等人 $[\text{BCI}^+13]$ 用线性非交互式证明（LIPs）的定义形式化这样的范式。一个线性非交互证明要求在有限域上，证明者和验证者的消息都由有限域元素向量组成。此外还要求证明者只能使用线性操作去计算她的消息。只要我们有一个合适的 2-move LIP，这个 LIP 就能够用配对的方式执行 “在指数上” 的等式，编译到一个 SNARK 中。我们提高效率的一个来源是我们为算术电路设计了一个 LIP 系统，其中证明者只发送 3 个群元素。对比 [GGPR13][PHGR13] 的二次算术程序中，证明者发送 4 个群元素的 LIP。

[1]我们注意到，针对通用敌手的合理性在组合中没有保留（这个问题也出现在 [Val08] 中），因为在编写与另一个 SNARK 验证对应的 statement 时，组合需要双线性群的具体实例。我们说的是，如果我们的 SNARK 在标准模型中是知识合理的，那么我们可以使用递归来得到完全简洁的 SNARKs。

提高效率的第二个来源是相比之前的工作，我们对 LIP 有一个更聚合的编译。Bitansky 等人 $\left[\text{BCI}^{+}13\right]$ 提出在一个对称双线性群上的转化，每个域元素被编译成两个群元素。然后他们用 KoE 假设，论证证明者知道相关的域元素。一个不太保守的选择是将每个域元素编译为单个组元素。每个域元素使用单个群元素进行编译提高了效率，但我们不能再使用 KoE 假设，所以我们只在通用组模型 [Sho97][BBG05] 证明了安全性。也可以在这两个极端之间做出选择，例如，Parno 等人 [PHGR13] 中有一个带有 4 个域元素的 LIP，它被编译成 7 个群元素。总而言之，在本文中，我们选择了最大效率并将 LIP 中的每个域元素编译为单个群元素，并在通用群模型中论证它的安全性。

我们更喜欢使用非对称双线性群，因为它们比对称双线性群效率更高。这意味着除了证明者在 LIP 中发送的域元素的数量以及如何聚合编译的选择外，还有更多的内容。使用非对称双线性群时，域元素可以在第一个群、第二个群或两者中表示为指数。我们的 LIP 是经过精心设计的，每个域元素都被编译成一个群元素，以便将证明大小总共减少到 3 个群元素。

更低的下界 研究更有效的非交互式论证，自然会考虑最小证明大小是多少。我们讲说明证明大小只有单个群元素，基于配对的 SNARGs 是不存在的。这个结果与 Bitansky 等人 $\left[\text{BCI}^{+}13\right]$ 中是否存在对验证者具有线性决策过程的 LIP 开放性问题相关。这样的线性决策过程将非常有用，例如它能够基于 ElGamal 加密构造 SNARGs。

我们消极回答了这个开放性问题，证明了具有线性决策过程的 LIPs 是不存在的。一个结论是任何基于配对的 SNARG 必须将证明中的群元素配对在一起，使得决策过程是二次的而不是线性的。因此，在不对称双线性群上时，在两个群中都有元素才能进行这样的配对。这排除了单个群元素的 SNARGs 存在，无论它是否是零知识，并表明我们的 NIZK 论证具有接近最优证明大小。通过构造一个 SNARG，其中每个群 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 中只有一个元素来完全缩小差距，这仍然是一个有趣的开放式问题，或者排除这种 SNARG 的存在。

2 预备知识

给定两个函数 $f, g : \mathbb{N} \to [0, 1]$，当 $|f(\lambda) - g(\lambda)| = \lambda^{-\omega(1)}$，记作 $f(\lambda) \approx g(\lambda)$。如果 $f(\lambda) \approx 0$，我们说 f 是可忽略的，如果 $f(\lambda) \approx 1$，我们说 f 压倒性的。我们将用 λ 表示安全参数，直觉上，随着 λ 的增长，我们期望更高的安全性。

当算法 A 的输入为 x 和随机数 r，输出为 y 时，记作 $y = A(x; r)$。随机选择一个随机数 r，令 $y = A(x; r)$ 的过程，记作 $y \leftarrow A(x)$。从集合 S 中均匀随机采样 y，记作 $y \leftarrow S$。

我们将假设从一个集合，如在 $\mathbb{Z}_p$ 中，均匀随机采样是可能的。

根据 Abe 和 Fehr [AF07]，当 $\mathcal{A}$ 输入为 x 输出为 y，且 $\mathcal{X}_{\mathcal{A}}$ 有相同输入（包括随机掷币）时输出 z，记作 $(y; z) \leftarrow (\mathcal{A} \| \mathcal{X}_{\mathcal{A}})(x)$。

2.1 双线性群

我们将使用双线性群 $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h)$，具有以下性质：

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ 是阶为 p 的群，p 为素数
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ 是一个双线性映射
- g 是 $\mathbb{G}_1$ 的生成元, h 是 $\mathbb{G}_2$ 的生成元，且 $e(g, h)$ 是 $\mathbb{G}_T$ 的生成元
- 存在一个有效的算法计算群操作，评估双线性映射，判断群成员，判断群元素是否相等，以及在群中采样生成元。我们指定这些操作为通用群操作。

有许多方法可以将双线性群设置为 $\mathbb{G}_1 = \mathbb{G}_2$ 的对称双线性群和 $\mathbb{G}_1 \neq \mathbb{G}_2$ 的非对称双线性群。Galbraith, Paterson 和 Smart[GPS08] 将双线性群进行分类，第 I 类为 $\mathbb{G}_1 = \mathbb{G}_2$，第 II 类中存在一个有效可计算非平凡的同态映射 $\Psi : \mathbb{G}_2 \to \mathbb{G}_1$，以及第 III 类 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 间没有这样的有效可计算非平凡的同态映射。第 III 类双线性群是最高效的双线性群，因此与实际应用最为相关。我们给出第 III 类双线性群基于配对的 SNARGs 的最低下界。另一方面，我们的构造能够被 3 类双线性群实例化。

用通过离散对数表示群元素的符号会很有用。我们强调离散对数很难计算，这些符号只是为了便于表示。用 $[a]_1$ 记作 g^a，用 $[b]_2$ 记作 h^b，用 $[c]_T$ 记作 $e(g, h)^c$。令单位元 $g = [1]_1, h = [1]_2$ 和 $e(g, h) = [1]_T$，零元为 $[0]_1$ $[0]_2$ 和 $[0]_T$。使用群的离散对数符号表示，很自然地在所有群中使用加法符号，例如 $[a]_T + [b]_T = [a + b]_T$。群元素的向量将表示为 $[a]_i$。我们的符号允许使用标准线性代数符号定义自然运算，因此有 $[\boldsymbol{a}]_i + [\boldsymbol{b}]_i = [\boldsymbol{a} + \boldsymbol{b}]_i$。假设 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 具有相同的维度，并且假设适当的维度，我们定义 $A[\boldsymbol{b}]_i = [A\boldsymbol{b}]_i$。给定两个 n 个群元素向量 $[\boldsymbol{a}]_1$ 和 $[\boldsymbol{b}]_2$，我们将它们的点积定义为 $[\boldsymbol{a}]_1 \cdot [\boldsymbol{b}]_2 = [\boldsymbol{a} \cdot \boldsymbol{b}]_T$，可以使用配对 e 有效计算。

如果算法只使用通用群操作来创建和操作群元素，我们称算法是通用的。Shoup[Sho97] 考虑用随机单射编码 $[\cdot]_i$ 代替真正的群元素，从而形式化了通用群模型（GGM）。然后通过算法可以访问的 oracle 处理通用群操作，例如，它可以在 (add, $[a]_i, [b]_i$) 上返回 $[a + b]_i$。由于编码的随机性，通用算法只能通过通用群预言机进行有意义的操作。这其中的一个含义是，如果它有输入 $[\boldsymbol{a}]_1$ 并返回元素 $[\boldsymbol{b}]$，我们可以通过检查它在 $\mathbb{G}_1$ 中所做的加法查询，有效地推导出矩阵 M 使得 $\boldsymbol{b} = M\boldsymbol{a}$。在 $\mathbb{G}_2$ 中也是如此，而在 $\mathbb{G}_T$ 中也可能有从配对操作计算的元素，但我们仍然可以将任何输出元素写为输入的显式二次多项式。

2.2 知识的非交互零知识论证

给定安全参数 λ，令 $\mathcal{R}$ 是一个关系生成器，返回一个多项式时间可判定的二元关系 R。对于 $(\phi, w) \in R$，我们称 ϕ 是 statement，w 是 witness。我们定义 $\mathcal{R}_\lambda$ 是给定 1^λ 的关系生成器，生成的所有可能关系 R 的集合。为了符号简单，我们下面将假设 λ 可以从 R 的描述中推导出来。关系生成器还可以输出一些辅助信息，如辅助输入 z，将提供给对手。$\mathcal{R}$ 的有效证明者的可公开验证的非交互式论证是概率多项式算法 (Setup, Prove, Vfy, Sim) 的四元组满足：

$(\sigma, \tau) \leftarrow \text{Setup}(R)$：生成一个公开参考字符串 σ，以及为 R 生成一个模拟陷门 τ。

$\pi \leftarrow \text{Prove}(R, \sigma, \phi, w)$：证明算法输入是一个公开参考字符串 σ 和 $(\phi, w) \in R$，返回论证 π。

$0/1 \leftarrow \text{Vfy}(R, \sigma, \phi, \pi)$：验证算法输入是一个公开参考字符串 σ，一个 statementϕ 和论证 π，返回 0(拒绝) 或者 1 (接受)。

$\pi \leftarrow \text{Sim}(R, \tau, \phi)$：模拟算法输入是一个模拟陷门和 statement ϕ，返回一个论证 π。

定义 1 如果 *(Setup, Prove, Vfy)* 满足以下定义的完美完备性和计算合理性，我们说它是 $\mathcal{R}$ 的一个非交互论证。

定义 2 如果 *(Setup, Prove, Vfy, Sim)* 满足以下定义的完美完备性，完美零知识和计算知识合理性，我们说它是 $\mathcal{R}$ 的一个完美非交互式零知识论证。

完美完备性（PERFECT COMPLETENESS. ）完备性表示，给定任何真实的 statement，诚实的证明者应该能够说服诚实的验证者。对于所有的 $\lambda \in \mathbb{N}, R \in \mathcal{R}_\lambda, (\phi, w) \in R$

$$\Pr[(\sigma, \tau) \leftarrow \text{Setup}(R); \pi \leftarrow \text{Prove}(R, \sigma, \phi, w) : \text{Vfy}(R, \sigma, \phi, \pi) = 1] = 1.$$

完美零知识性（PERFECT ZERO-KNOWLEDGE. ）如果一个论证除了 statement 是真实的以外，没有泄露任何信息，那么它是零知识的。如果对于所有的 $\lambda \in \mathbb{N}, (R, z) \leftarrow \mathcal{R}\left(1^\lambda\right), (\phi, w) \in R$ 和所有的敌手 $\mathcal{A}$，(Setup, Prove, Vfy, Sim) 满足

$$\begin{aligned}&\Pr[(\sigma, \tau) \leftarrow \text{Setup}(R); \pi \leftarrow \text{Prove}(R, \sigma, \phi, w) : \mathcal{A}(R, z, \sigma, \tau, \pi) = 1] \\ =&\Pr[(\sigma, \tau) \leftarrow \text{Setup}(R); \pi \leftarrow \text{Sim}(R, \tau, \phi) : \mathcal{A}(R, z, \sigma, \tau, \pi) = 1].\end{aligned}$$

我们说它是完美零知识的。

计算合理性（COMPUTATIONAL SOUNDNESS.）如果 (Setup, Prove, Vfy, Sim) 不可能证明一个错误的 statement，如在没有 witness 的情况下让验证者信服，我们说它是合理的。令 L_R 为一个包含了 statement 的语言，在 R 中存在 witness 与其中的 statement 对应。形式化地，我们要求对于所有非均匀多项式时间的敌手 $\mathcal{A}$，满足

$$\Pr\left[\begin{array}{c}(R, z) \leftarrow \mathcal{R}\left(1^\lambda\right); (\sigma, \tau) \leftarrow \text{Setup}(R); (\phi, \pi) \leftarrow \mathcal{A}(R, z, \sigma) : \\ \phi \notin L_R \text{ and } \text{Vfy}(R, \sigma, \phi, \pi) = 1\end{array}\right] \approx 0$$

计算知识合理性（COMPUTATIONAL KNOWLEDGE SOUNDNESS.）加强合理性的概念，如果存在一个提取器，当敌手生成一个合法的论证时，能够计算出 witness，我们说 (Setup, Prove, Vfy, Sim) 是一个知识的论证。提取器能够完全访问敌手的状态，包括随机掷币。形式化地，我们要求对于所有非均匀多项式时间的敌手 $\mathcal{A}$，存在一个非均匀多项式时间的提取器 $\mathcal{X}_\mathcal{A}$ 满足

$$\Pr\left[\begin{array}{c}(R, z) \leftarrow \mathcal{R}\left(1^\lambda\right); (\sigma, \tau) \leftarrow \text{Setup}(R); ((\phi, \pi); w) \leftarrow (\mathcal{A} \| \mathcal{X}_\mathcal{A})(R, z, \sigma) :] \approx 0. \\ (\phi, w) \notin R \text{ and } \text{Vfy}(R, \sigma, \phi, \pi) = 1\end{array}\right.$$

公开可验证和指定验证者(PUBLIC VERIFIABILITY AND DESIGNATED VERIFIER PROOFS.)
我们可以通过将 σ 拆分为证明者和验证者分别使用的两部分 σ_P 和 σ_V 来自然地概括非交互式论证

的定义。σ_V 能够从 σ_P 中推导时，我们说非交互论证是公开可验证的。否则我们把它称为指定验证者论证。对于指定验证者论证，可以放宽合理性和知识合理性，使得敌手只看到 σ_P 而看不到 σ_V。

SNARGs 和 SNARKs（SNARGs AND SNARKs.）一个非交互论证中，验证者在多项式时间 $\lambda + |\phi|$ 内运行，证明大小是 λ 的多项式，如果它是合理的，那么被称为预处理的简洁非交互证明 (SNARG)，如果它是知识合理的，那么被称为知识的预处理简洁非交互证明 (SNARK)。如果我们限制 CRS 为 λ 的多项式长度，我们说这是完整简洁的 SNARG 或 SNARK。Bitansky 等人 [BCCT13] 预处理的 SNARK 能够被组合实现完整简洁的 SNARK。本文重点为预处理的 SNARK，CRS 可以很长。

良性关系生成器（BENIGN RELATION GENERATORS.）Bitansky 等人 [BCPR14] 表明，不可区分混淆意味着对于每个可能的 SNARK 都有辅助输出分布，使对手能够创建一个不可能提取出 witness 的有效证明。考虑公共掷币不同输入混淆和其他加密假设，Boyle 和 Pass [BP15] 加强了这种不可能性，表明存在一个辅助输出分布，它破坏了所有可能的 SNARK 的 witness 提取。然而，这些反例依赖于特定的辅助输入分布。因此，我们将在下面假设关系生成器是良性的，因为关系和辅助输入的分布方式使得我们构建的 SNARK 可以是知识合理的。

2.3 二次算术程序

考虑一个算术电路，其中包含了有限域 $\mathbb{F}$ 上的加法门和乘法门。我们可以将一些输入/输出线路指定为 statement，电路中剩下的线路指定为 witness。这给了我们一个满足二元关系 R 的算术电路，包括 statement 线路和 witness 线路，即使其与指定的输入/输出线路一致。

概括算术电路,我们可能会对一组变量上的等式描述的关系感兴趣。一部分变量对应 statement，另一部分变量对应 witness。关系包括了满足所有等式的 statement 和 witness。这些等式定义在 $a_0 = 1$ 和变量 $a_1, \ldots, a_m \in \mathbb{F}$ 上，有如下形式

$$\sum a_i u_{i,q} \cdot \sum a_i v_{i,q} = \sum a_i w_{i,q}$$

其中 $u_{i,q}, v_{i,q}, w_{i,q}$ 是第 q 个等式，是 $\mathbb{F}$ 上的常数。

我们发现加法和乘法门是方程的特例，因此此类算术约束系统确实可以概括算术电路。例如，一个乘法门可以被描述成 $a_i \cdot a_j = a_k$（令 $u_i = 1, v_j = 1$ 且 $w_k = 1$，并设置门的其余常数为 0）。加法门在定义方程的总和中无代价处理，例如计算 $a_i + a_j = a_k$，a_k 乘上 a_ℓ，我们可以简单地写成 $(a_i + a_j) \cdot a_\ell$ 并跳过 a_k 的计算。

由 Gennaro, Gentry, Parno 和 Raykova[GGPR13]，我们重新形式化算术约束为二次算数程序，假设 $\mathbb{F}$ 足够大。给定 n 个等式，我们选择任意不同的 $r_1, \ldots, r_n \in \mathbb{F}$，令 $t(x) = \prod_{q=1}^{n} (x - r_q)$。此外，令 $u_i(x), v_i(x), w_i(x)$ 为阶 $n-1$ 的多项式，满足对所有 $i = 0, \ldots, m, q = 1, \ldots, n$，有 $u_i(r_q) = u_{i,q} \quad v_i(r_q) = v_{i,q} \quad w_i(r_q) = w_{i,q}$ 成立。我们现在有 $a_0 = 1$ 和变量 $a_1, \ldots, a_m \in \mathbb{F}$ 满足

n 个等式，当且仅当在每个点 $r_1, \ldots, r_q$ 上有

$$\sum_{i=0}^{m} a_i u_i\left(r_q\right) \cdot \sum_{i=0}^{m} a_i v_i\left(r_q\right) = \sum_{i=0}^{m} a_i w_i\left(r_q\right)$$

成立。因为 $t(X)$ 是最低阶的单项式，在每个点上有 $t\left(r_q\right) = 0$ 成立，我们重新形式化这个条件为

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) \equiv \sum_{i=0}^{m} a_i w_i(X) \bmod t(X)$$

形式上，我们将使用具有以下描述的二次算术程序 R

$$R = \left(\mathbb{F}, \text{aux}, \ell, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X)\right),$$

其中 $\mathbb{F}$ 是有限域,aux 是辅助信息,$1 \leq \ell \leq m, u_i(X), v_i(X), w_i(X), t(X) \in \mathbb{F}[X]$ 且 $u_i(X), v_i(X), w_i(X)$ 的阶严格小于 $t(X)$ 的阶 n。我们定义 $a_0 = 1$，具有这种描述的二次算术程序定义了以下二元关系

$$R = \left\{ (\phi, w) \;\middle|\; \begin{array}{l} \phi = (a_1, \ldots, a_\ell) \in \mathbb{F}^\ell \\ w = (a_{\ell+1}, \ldots, a_m) \in \mathbb{F}^{m-\ell} \\ \sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) \equiv \sum_{i=0}^{m} a_i w_i(X) \bmod t(X) \end{array} \right\}$$

如果 $\mathcal{R}$ 在大小超过 $2^{\lambda-1}$ 的域上生成这种形式的关系，我们称 $\mathcal{R}$ 是一个二次算数程序生成器。

在实践中，关系可以以许多不同的方式出现。关系生成器可能是确定性的，也可能是随机的。一种方法是首先生成域 $\mathbb{F}$，然后在该域之上构建关系的其余部分。或者首先指定多项式，然后再选择一个随机的域。为了获得最大的灵活性，我们让我们的定义相对于域和关系生成的确切方式不可知，不同的选择都可以通过适当的关系生成器选择来建模。

展望未来，我们将在我们基于配对的 NIZK 论证上，让辅助信息 aux 指定一个双线性群。将双线性群的选择作为关系生成器的一部分似乎有点令人惊讶，但这提供了一个更好的设置模型，其中关系建立在已经存在的双线性群之上。同样，在这种选择中没有失去一般性，可以将首先选择关系然后随机选择双线性群的传统设置视为关系生成器分两步工作的特殊情况，首先选择关系，然后选择一个随机的双线性群。当然，让关系生成器选择双线性群是我们需要假设它是良性的另一个很好的理由；一个双线性群合适的选择对于安全性来说是重要的。

2.4 线性非交互证明

Bitansky 等人 [BCI$^+$13] 给出了一个最近的 SNARK 构造的有用的信息论特征，他们称之为 2-move 代数输入不经意线性交互证明。为了阐明与我们在 2.2 节中定义的非交互式论证的联系，我们将把这个概念重命名为非交互线性证明（NILP）。NILP 是相对于关系生成器 $\mathcal{R}$ 定义的，我们假设关系指定一个有限域 $\mathbb{F}$，并按如下方式工作:

$(\sigma, \tau) \leftarrow \text{Setup}(R)$：设置阶段是一个概率多项式时间算法，返回向量 $\sigma \in \mathbb{F}^m$ 和 $\tau \in \mathbb{F}^n$。为了符号的简单性，我们假设 $\boldsymbol{\sigma}$ 总是包含 1 作为条目，这样 $\boldsymbol{\sigma}$ 的仿射函数和线性函数之间没有区别。

$\pi \leftarrow \text{Prove}(R, \boldsymbol{\sigma}, \phi, w)$：证明者在两个阶段操作

- 首先运行 $\Pi \leftarrow \text{ProofMatrix}(R, \phi, w)$，其中 ProofMatrix 是概率多项式时间算法，生成矩阵 $\Pi \in \mathbb{F}^{k \times m}$。
- 然后计算证明 $\pi = \Pi\boldsymbol{\sigma}$。

$0/1 \leftarrow \text{Vfy}(R, \boldsymbol{\sigma}, \phi, \pi)$: 验证者在两个阶段操作

- 首先运行确定多项式时间算法 $t \leftarrow \text{Test}(R, \phi)$，得到一个算术电路 $\boldsymbol{t} : \mathbb{F}^{m+k} \to \mathbb{F}^{\eta}$，对应一个总阶数为 d 的多元多项式向量评估。
- 当且仅当 $\boldsymbol{t}(\sigma, \pi) = \mathbf{0}$ 时接受证明。

阶数 d 和维度 μ, m, n, k, η 可以是安全参数 λ 下的常数或多项式。

定义 3（线性非交互证明）. 如果 *(Setup, Prove, Vfy)* 对以下定义的对仿射证明者策略具有完美完备性和统计知识合理性，那么说它是关系 $\mathcal{R}$ 一个线性非交互证明。

对仿射证明者策略的统计知识合理性(STATISTICAL KNOWLEDGE SOUNDNESS AGAINST AFFINE PROVER STRATEGIES.）如果能够在验证通过的证明矩阵 Π 中提取一个证据，那么 NILP 具有对仿射证明者策略的统计知识合理性。更具体的，对所有敌手 $\mathcal{A}$，存在一个多项式时间的提取器 $\mathcal{X}$

$$\Pr\left[\begin{array}{c}(R, z) \leftarrow \mathcal{R}\left(1^{\lambda}\right); (\boldsymbol{\sigma}, \boldsymbol{\tau}) \leftarrow \text{Setup}(R); (\phi, \Pi) \leftarrow \mathcal{A}(R, z); w \leftarrow \mathcal{X}(R, \phi, \Pi) :] \\ \Pi \in \mathbb{F}^{m \times k} \wedge \text{Vfy}(R, \boldsymbol{\sigma}, \phi, \Pi\boldsymbol{\sigma}) = \mathbf{0} \wedge (\phi, w) \notin R\end{array}\right] \approx 0.$$

同时将 2.2 节零知识的概念应用在 NILP 且对应一个 2-move 诚实验证者零知识的 LIP。其它到指定验证者 NILP 的潜在扩展中，CRS $\boldsymbol{\sigma}$ 被分成两部分，证明者使用 σ_P，验证者使用 σ_V。

2.5 从线性非交互证明到非交互论证

用配对的方式能够将 NILP 编译到公开可验证非交互证明，用 Paillier 加密系统 [BCI+13] 的变体能够 NILP 将编译到指定验证者非交互论证，因此 NILP 非常有用。如果我们使用配对设置，直觉是验证者阶数 $d = 2$ 的 NILP 能够" 在离散对数算法" 中执行。CRS 包括 σ 中域元素的编码。证明者利用 CRS 中群元素的线性组合计算证明。验证者通过验证一些配对乘积等式（将配对结果相乘形成的等式）检查论证，这些等式对应检查在编码后域元素上的二次等式。我们现在形式化这个方法。

当使用第 III 种类型的配对时，执行" 在离散对数算法" 中的 NILP 要求我们为每个元素指定应该在哪个群上进行操作。因此，我们将定义一个 split NILP，它的 CRS 分成两部分 $\sigma = (\sigma_1, \sigma_2)$，证明者的证明也分成两部分 $\pi = (\pi_1, \pi_2)$。证明的每一部分由对应部分的 CRS 计算得到。最后，当验证证明时，我们想让验证者的测试为一个二次等式，等式中每个变量阶数为 1。一个 split NILP 具有如下形式:

$(\sigma,\tau) \leftarrow \mathrm{Setup}(R)$：设置算法生成向量 $\sigma = (\sigma_1, \sigma_2) \in \mathbb{F}^{m_1} \times \mathbb{F}^{m_2}$ 且 $\tau \in \mathbb{F}^n$。为了符号的简单性，我们假设 σ_1 和 σ_2 都包含 1 作为条目，这样 $\boldsymbol{\sigma}$ 的仿射函数和线性函数之间没有区别。

$\pi \leftarrow \mathrm{Prove}(R, \sigma, \phi, w)$：证明者在两个阶段操作

- 首先运行 $\Pi \leftarrow \mathrm{ProofMatrix}(R, \phi, w)$，其中 ProofMatrix 生成一个 $\Pi = \begin{pmatrix} \Pi_1 & 0 \\ 0 & \Pi_2 \end{pmatrix}$ 形式的矩阵，$\Pi_1 \in \mathbb{F}^{k_1 \times m_1}$ and $\Pi_2 \in \mathbb{F}^{k_2 \times m_2}$。
- 然后计算 $\pi_1 = \Pi_1 \sigma_1$ 和 $\pi_2 = \Pi_2 \sigma_2$，返回证明 $\pi = (\pi_1, \pi_2)$。

$0/1 \leftarrow \mathrm{Vfy}(R, \sigma, \phi, \pi)$：验证者在两个阶段操作

- 首先运行 $\boldsymbol{t} \leftarrow \mathrm{Test}(R, \phi)$，得到一个算数电路 $\boldsymbol{t} : \mathbb{F}^{m_1+k_1+m_2+k_2} \to \mathbb{F}^{\eta}$，对应矩阵 $T_1, \ldots, T_\eta \in \mathbb{F}^{(m_1+k_1)\times(m_2+k_2)}$。
- 当且仅当对所有矩阵 $T_1, \ldots, T_\eta$ 满足

$$\begin{pmatrix} \sigma_1 \\ \pi_2 \end{pmatrix} \cdot T_i \begin{pmatrix} \sigma_2 \\ \pi_2 \end{pmatrix} = 0$$

时接受证明。

直觉上，在编译 split NILP 后我们想通过说明不诚实的证明者利用通用群操作不能偏离 NILP 来讨论合理性。然而，当证明者看到 CRS 时，她可以从中知道有用的信息，并且基于这些信息用某种方式去选择她的矩阵 Π。为了对抗这种类型的敌手，我们将定义 disclosure-free 的 CRS，证明者无法从中获取有用的信息，帮助她选择一个特殊的矩阵 Π。

定义 4 (*Disclosure-free NILP*) 如果对于所有敌手 $\mathcal{A}$，*split NILP* 满足

$$\Pr\left[\begin{array}{c} (R,z) \leftarrow \mathcal{R}\left(1^\lambda\right); T \leftarrow \mathcal{A}(R,z); \left(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\tau}\right), \left(\boldsymbol{\sigma}_1', \boldsymbol{\sigma}_2', \boldsymbol{\tau}'\right) \leftarrow \mathrm{Setup}(R): \\ \boldsymbol{\sigma}_1 \cdot T \boldsymbol{\sigma}_2 = 0 \textit{ and only if } \boldsymbol{\sigma}_1' \cdot T \boldsymbol{\sigma}_2' = 0 \end{array}\right] \approx 1$$

那么我们说它是 *disclosure-free* 的。

解释 disclosure-free 的 CRS，敌手在 σ_1, σ_2 上运行的任何测试输出，能够在独立生成的 σ_1', σ_2' 上运行同样的测试进行预测。

我们现在准备好用具有 disclosure-free CRS 的 split NILP(Setup, Prove, Vfy, Sim) 描述一个编译器，从而得到一个基于配对的非交互论证 $(\mathrm{Setup}', \mathrm{Prove}', \mathrm{Vfy}', \mathrm{Sim}')$:

$(\sigma,\tau) \leftarrow \mathrm{Setup}'(R)$：运行 $\left(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\tau}\right) \leftarrow \mathrm{Setup}(R)$，返回 $\sigma = \left(\left[\boldsymbol{\sigma}_1\right]_1, \left[\boldsymbol{\sigma}_2\right]_2\right)$ 及 $\tau = \boldsymbol{\tau}$。

$\pi \leftarrow \mathrm{Prove}'(R, \sigma, \phi, w)$:生成 $\left(\Pi_1, \Pi_2\right) \leftarrow \mathrm{ProofMatrix}(R, x, w)$,返回如下计算的 $\pi = \left(\left[\pi_1\right]_1, \left[\pi_2\right]_2\right)$

$$\left[\pi_1\right]_1 = \Pi_1 \left[\boldsymbol{\sigma}_1\right]_1 \quad \left[\boldsymbol{\pi}_2\right]_2 = \Pi_2 \left[\boldsymbol{\sigma}_2\right]_2 .$$

$0/1 \leftarrow \mathrm{Vfy}'(R, \sigma, \phi, \pi)$：生成 $(T_1, \ldots, T_\eta) \leftarrow \mathrm{Test}(R, \phi)$。解析 $\pi = \left(\left[\boldsymbol{\pi}_1\right]_1, \left[\boldsymbol{\pi}_2\right]\right)_2 \in \mathbb{G}_1^{k_1} \times \mathbb{G}_2^{k_2}$。当且仅当对于所有的 $T_1, \ldots, T_\eta$ 满足

$$\begin{bmatrix} \sigma_1 \\ \pi_1 \end{bmatrix}_1 \cdot T_i \begin{bmatrix} \sigma_2 \\ \pi_2 \end{bmatrix}_2 = [0]_T.$$

接受证明。

$\pi \leftarrow \mathrm{Sim}'(R, \tau, \phi)$：模拟 $(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \leftarrow \mathrm{Sim}(R, \boldsymbol{\tau}, \phi)$，返回 $\pi = \left(\left[\boldsymbol{\pi}_1\right]_1, \left[\boldsymbol{\pi}_2\right]_2\right)$。

引理 1 *上面给出的协议具有完美完备性和对只能进行多项式次通用群操作敌手的统计知识合理性。如果潜在的 split NILP 是完美零知识的，那么该协议是完美零知识的。*

证明 完美完备性来自 NILP 的完美完备性，以及它是 split NILP 的事实，允许敌手在两个相关的群 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 上使用通用群操作，计算 $\left[\pi_1\right]_1, \left[\pi_2\right]_2$ 两部分证明。

完美零知识性来自 NILP 的完美零知识性质。

保留论证对通用敌手的统计合理性。通用的敌手能够使用通用群操作，对 $\mathbb{G}_1, \mathbb{G}_2$ 和 $\mathbb{G}_T$ 上的元素进行相乘，测试群成员，评估配对，以及测试元素是否相等。

我们首先论证 disclosure-free 意味着敌手以可忽略的概率从 CRS 中知道非平凡的信息。每当敌手测试通用群计算的元素是否为 0 时，它可以写成 $\left[\boldsymbol{\sigma}_1\right]_1 \cdot T \left[\boldsymbol{\sigma}_2\right]_2 = [0]_T$ 配对乘积相等的测试，其中 T 可以从敌手的通用群请求中推断出来。我们运行一个修改过的敌手来代替进行这些查询，这个敌手选择替代的公共参考字符串 $(\sigma_1', \sigma_2', \tau')$ 并通过测试 $\sigma_1' \cdot T\sigma_2' = 0$ 自己回答查询。由于 disclosure-free，以这种方式得出的回答与敌手在真正的 CRS 上看到的回答相等具有压倒性的概率，因此我们从现在开始假设通用对手不会对涉及 CRS 的元素进行任何零测试。

敌手不会在 CRS 上进行任何的零测试并且只使用通用群操作，等价于敌手选择两个独立于 $\left[\sigma_1\right]_1, \left[\sigma_2\right]_2$ 的矩阵 Π_1, Π_2，然后计算证明 $\left[\pi_1\right]_1 = \Pi_1 \left[\boldsymbol{\sigma}_1\right]$ 和 $\left[\pi_2\right]_2 = \Pi_2 \left[\sigma_2\right]_2$。由离散对数，这对应运行 split NILP 知识合理性的敌手，得到矩阵 Π_1, Π_2 和证明 $\pi_1 = \Pi_1\sigma_1, \pi_2 = \Pi_1\sigma_2$。

运用验证等式的离散对数，我们看到如果敌手成功找到 ϕ 和合法的证明 π_1, π_2，这对应找到 ϕ 和 Π_1, Π_2 满足测试矩阵 $T_1, \ldots, T_\eta \leftarrow \mathrm{Test}(R, \phi)$

$$\begin{pmatrix} \sigma_1 \\ \Pi_1\sigma_1 \end{pmatrix} \cdot T_i \begin{pmatrix} \boldsymbol{\sigma}_2 \\ \Pi_2\sigma_2 \end{pmatrix} = 0$$

由 split NILP 的统计合理性，这样的概率是可忽略的，除非 Π_1, Π_2 的知识能够提取一个 witness w 满足 $(\phi, w) \in R$。 □

如果我们使用 split NILP，它只对仅限于输出矩阵 $\Pi = \begin{pmatrix} \Pi_1 & 0 \\ 0 & \Pi_2 \end{pmatrix}$ 的 split 仿射对手具有合理性，则引理 1 的证明也成立。然而，我们后面构造的 split NILP 实际上对于 Π 的任何选择都是安全的。这样做的好处是对抗密码分析，即使敌手在 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 之间找到了一个有效可计算的同构，

即使它形成了 g 到 h 的映射，我们在通用群模型中仍然具有安全性。另一个优点是该构造也适用于具有微小变化的对称双线性群。

3 非交互论证的构造

我们将构造一个二次算数程序基于配对的 NIZK 论证，其中证明进包含 3 个群元素。我们分两步进行构造，首先我们构造一个二次算数程序的 NILP，然后我们发现它也是一个 split NILP，用我们之前展示的编译技术将它转化成基于配对的论证。

3.1 二次算数程序的非交互线性证明

我们现在将构造一个二次算数程序的 NILP 生成器，输出如下形式的关系

$$R = (\mathbb{F}, \text{aux}, \ell, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X))$$

这个关系定义了一个 statement 为 $(a_1, \ldots, a_\ell) \in \mathbb{F}^\ell$ 且证据为 $(a_{\ell+1}, \ldots, a_m) \in \mathbb{F}^{m-\ell}$ 的语言，$a_0 = 1$，对某些阶为 $n-2$ 的商多项式 $h(X)$，阶为 n 的多项式 $t(X)$，满足

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) = \sum_{i=0}^{m} a_i w_i(X) + h(X)t(X),$$

$(\boldsymbol{\sigma}, \boldsymbol{\tau}) \leftarrow \text{Setup}(R)$：选择 $\alpha, \beta, \gamma, \delta, x \leftarrow \mathbb{F}^*$。设置 $\boldsymbol{\tau} = (\alpha, \beta, \gamma, \delta, x)$，

$$\boldsymbol{\sigma} = \left(\alpha, \beta, \gamma, \delta, \{x^i\}_{i=0}^{n-1}, \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma} \right\}_{i=0}^{\ell}, \left\{ \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta} \right\}_{i=\ell+1}^{m}, \left\{ \frac{x^i t(x)}{\delta} \right\}_{i=0}^{n-2} \right)$$

$\pi \leftarrow \text{Prove}(R, \sigma, a_1, \ldots, a_m)$：选择 $r, s \leftarrow \mathbb{F}$，计算一个 $3 \times (m+2n+4)$ 的矩阵 Π 满足 $\boldsymbol{\pi} = \Pi\boldsymbol{\sigma} = (A, B, C)$，其中 A, B, C 为

$$A = \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta \quad B = \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta$$
$$C = \frac{\sum_{i=\ell+1}^{m} a_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right) + h(x)t(x)}{\delta} + As + rB - rs\delta.$$

$0/1 \leftarrow \text{Vfy}(R, \sigma, a_1, \ldots, a_\ell)$：计算一个二次多变量多项式 $\boldsymbol{t}$，对应如下测试 $\boldsymbol{t}(\boldsymbol{\sigma}, \boldsymbol{\pi}) = 0$

$$A \cdot B = \alpha \cdot \beta + \frac{\sum_{i=0}^{\ell} a_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right)}{\gamma} \cdot \gamma + C \cdot \delta.$$

当且仅当等式成立时接受证明。

$\pi \leftarrow \text{Sim}(R, \tau, a_1, \ldots, a_\ell)$：选择 $A, B \leftarrow \mathbb{F}$，计算 $C = \frac{AB - \alpha\beta - \sum_{i=0}^{\ell} a_i(\beta u_i(x) + \alpha v_i(x) + w_i(x))}{\delta}$，返回 $\pi = (A, B, C)$。

在形式化证明这是一个 NILP 之前，我们给出在不同组件后的一些直觉。α 和 β 的作用是确保在每个 $a_0, \ldots, a_m$ 选择下，A, B 和 C 的一致性。在验证等式中乘积 $\alpha \cdot \beta$ 保证 A 和 B 中的 α 和 β 是非平凡的。这意味着乘积 $A \cdot B$ 对 $alpha$ 和 β 的线性依赖，我们随后证明这个线性依赖只能由 C 来平衡，其中 A, B 和 C 中 $a_0, \ldots, a_m$ 的选择一致。γ 和 δ 的作用是使验证方程的后两个乘积独立于第一个乘积，方法是将左侧因子分别除以 γ 和 δ。这防止了验证方程中对不同乘积的元素混合匹配。最后，我们用 r 和 s 随机化证明获得零知识。

定理 1 *以上结构实现了一个完美完备性，完美零知识性，以及对仿射证明者策略的统计知识合理性的 NILP。*

证明 完美完备性很容易验证。完美零知识性来自真实的证明和具有均匀随机域元素 A, B 的模拟证明。有验证等式，这些元素唯一确定 C，所以真实的证明和模拟的证明有相同的概率分布。

接下来证明对于任意仿射证明者策略，我们有不可忽略的概率提取一个 witness。当使用一个仿射证明者策略时，已知域元素 $A_\alpha, A_\beta, A_\gamma, A_\delta, A_i, n-1$ 阶多项式 $A(x), n-2$ 阶多项式 $A_h(x)$，分别对应矩阵 Π 的第一行，我们有

$$\begin{aligned} A =& A_\alpha \alpha + A_\beta \beta + A_\gamma \gamma + A_\delta \delta + A(x) + \sum_{i=0}^{\ell} A_i \frac{\beta u_i(x)}{\delta} + A_h(x)\frac{t(x)}{\delta} \\ &+ \sum_{i=\ell+1}^{m} A_i \frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta} + A_h(x)\frac{t(x)}{\delta} \end{aligned}$$

我们将 B 和 C 写成相似的形式，作为 Π 的第二行和第三行。

我们现在将验证等式看做一个多变量的 Laurent 多项式。根据 Schwartz-Zippel 引理，除非在 $\alpha, \beta, \gamma, \delta, x$ 不确定的情况下，A, B, C 对应的形式多项式使得验证等式仍然成立，否则证明者的成功概率可以忽略不计。

α^2 的不定项是 $A_\alpha B_\alpha \alpha^2 = 0$, 意味着 $A_\alpha = 0$ 或者 $B_\alpha = 0$。因为 $AB = BA$, 我们可以不失一般性假设 $B_\alpha = 0$。$\alpha\beta$ 的不定项给了我们 $A_\alpha B_\beta + A_\beta B_\alpha = A_\alpha B_\beta = 1$。这意味着 $AB = (AB_\beta)(A_\alpha B)$, 所以我们可以再重新调整后不失一般性假设 $A_\alpha = B_\beta = 1$。β^2 的不定项给了我们 $A_\beta B_\beta = A_\beta = 0$。我们现在简化敌手构造的 A 和 B 为以下形式

$$A = \alpha + A_\gamma \gamma + A_\delta \delta + A(x) + \cdots \quad B = \beta + B_\gamma \gamma + B_\delta \delta + B(x) + \cdots$$

接下来，考虑有关 $\frac{1}{\delta^2}$ 的项。我们有

$$\begin{aligned} &\left(\sum_{i=\ell+1}^{m} A_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right) + A_h(x)t(x) \right) \\ &\left(\sum_{i=\ell+1}^{m} B_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right) + B_h(x)t(x) \right) = 0 \end{aligned}$$

左侧因子为 0 或者右侧因子为 0。由对称性，我们不失一般性地假设 $\sum_{i=\ell+1}^{m} A_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)+t(x)A_t(x)=0$。在 $\alpha\frac{\sum_{i=\ell+1}^{m} B_i(\beta u_i(x)+\alpha v_i(x)+w_i(x))+B_h(x)t(x)}{\delta}=0$ 中的项也显示有 $\sum_{i=\ell+1}^{m} B_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)+B_h(x)t(x)=0$ 成立。

有关 $\frac{1}{\gamma^2}$ 的项给我们

$$\sum_{i=0}^{\ell} A_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)\cdot\sum_{i=0}^{\ell} B_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)=0,$$

左侧因子为 0 或者右侧因子为 0。由对称性，我们不失一般性地假设 $\sum_{i=0}^{\ell} A_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)=0$。在 $\alpha\frac{\sum_{i=0}^{m} B_i(\beta u_i(x)+\alpha v_i(x)+w_i(x))}{\gamma}=0$ 的项也显示有 $\sum_{i=0}^{\ell} B_i\left(\beta u_i(x)+\alpha v_i(x)+w_i(x)\right)=0$ 成立。

项 $A_\gamma\beta\gamma=0$ 和 $B_\gamma\alpha\gamma=0$ 意味着 $A_\gamma=0, B_\gamma=0$。我们有

$$A=\alpha+A(x)+A_\delta\delta\quad B=\beta+B(x)+B_\delta\delta.$$

验证等式的关于 α 的剩余项 $\alpha B(x)=\sum_{i=0}^{\ell} a_i\alpha v_i(x)+\sum_{i=\ell+1}^{m} C_i\alpha v_i(x)$。关于 β 的项 $\beta A(x)=\sum_{i=0}^{\ell} a_i\beta u_i(x)+\sum_{i=\ell+1}^{m} C_i\beta u_i(x)$。对 $i=\ell+1,\ldots,m$，定义 $a_i=C_i$，我们有

$$A(x)=\sum_{i=0}^{m} a_i u_i(x)\quad B(x)=\sum_{i=0}^{m} a_i v_i(x).$$

最后，我们观察 x 相关的指数

$$\sum_{i=0}^{m} a_i u_i(x)\cdot\sum_{i=0}^{m} a_i v_i(x)=\sum_{i=0}^{m} a_i w_i(x)+C_h(x)t(x).$$

因此 $(a_{\ell+1},\ldots,a_m)=(C_{\ell+1},\ldots,C_m)$ 是对 statement$(a_1,\ldots,a_\ell)$ 的 witness。 □

2 个域元素的 NILP（2 FIELD ELEMENT NILPs）很自然的一个问题是，证明者在 NILP 中发送的域元素的数量是否可以进一步减少。Danezis 等人 [DFGK14] 的平方扩展程序为布尔可满足性电路生成了 2 个域元素的 NILP。对于算术可满足性电路，将电路重写为只用平方门，可能得到一个 2 元素的 NILP，因为每个乘法门 $a\cdot b=c$ 能够被重写为 $(a+b)^2-(a-b)^2=4c$。当算术电路只有平方门时，对所有的 i 有 $u_i(x)=v_i(x)$。NILP 中选择 $r=s$，我们有 $B=A+\beta-\alpha$，所以证明者只需要发送两个元素 A 和 C 就可以做出令人信服的证明。将算术电路重写为仅使用平方门可能会使门的数量增加一倍，并且还需要一些额外的电路来减去平方，因此减少 NILP 的大小会带来巨大的计算成本。

3.2 对二次算数程序的 NIZK 论证

我们现在将给出一个基于配对的二次算数程序。我们考虑关系生成器 $\mathcal{R}$，它将返回这样形式的关系

$$R=\left(p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,g,h,\ell,\{u_i(X),v_i(X),w_i(X)\}_{i=0}^{m},t(X)\right),$$

其中 $|p| = \lambda$。这样的关系定义了一个域 $\mathbb{Z}_p$ 和一个 statement 的语言 $(a_1, \ldots, a_\ell) \in \mathbb{Z}_p^\ell$，$a_0 = 1$，对于某个 $n-2$ 次的商多项式 $h(X)$ 满足

$$\sum_{i=0}^{m} a_i u_i(X) \cdot \sum_{i=0}^{m} a_i v_i(X) = \sum_{i=0}^{m} a_i w_i(X) + h(X)t(X)$$

我们上面给出的 NILP 一个重要设计特点是可以很容易地使其成为一个 split NILP。证明元素 A B 和 C 在验证方程中只使用一次，因此很容易将它们分配到双线性测试的不同侧。将 CRS 分成两部分，以便计算证明的每一侧，我们就得到了一个 split NILP。由此得到的 split NILP 是 disclosure-free 的，因此可以像我们在第 2.5 节中所做的那样在通用群模型中编译为 NIZK 论证。由于配对友好的椭圆曲线通常有 $\mathbb{G}_1$ 中的群元素表示小于 $\mathbb{G}_2$[GPS08] 中的群元素表示，因此我们选择将 A 和 C 分配给 $\mathbb{G}_1$，将 B 分配给 $\mathbb{G}_2$，以获得最大效率。这给出我们以下 NIZK 论证

$(\sigma, \tau) \leftarrow \mathrm{Setup}(R)$：选择 $\alpha, \beta, \gamma, \delta, x \leftarrow \mathbb{Z}_p^*$。定义 $\tau = (\alpha, \beta, \gamma, \delta, x)$ 并计算 $\sigma = ([\boldsymbol{\sigma}_1]_1, [\boldsymbol{\sigma}_2]_2)$：

$$\boldsymbol{\sigma}_1 = \begin{pmatrix} \alpha, \beta, \delta, \left\{x^i\right\}_{i=0}^{n-1}, \left\{\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma}\right\}_{i=0}^{\ell} \\ \left\{\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\delta}\right\}_{i=\ell+1}^{m}, \left\{\frac{x^i t(x)}{\delta}\right\}_{i=0}^{n-2} \end{pmatrix} \quad \boldsymbol{\sigma}_2 = \left(\beta, \gamma, \delta, \left\{x^i\right\}_{i=0}^{n-1}\right).$$

$\pi \leftarrow \mathrm{Prove}(R, \sigma, a_1, \ldots, a_m)$：选择 $r, s \leftarrow \mathbb{Z}_p$ 并计算 $\pi = ([A]_1, [C]_1, [B]_2)$：only if

$$A = \alpha + \sum_{i=0}^{m} a_i u_i(x) + r\delta \quad B = \beta + \sum_{i=0}^{m} a_i v_i(x) + s\delta$$

$$C = \frac{\sum_{i=\ell+1}^{m} a_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right) + h(x)t(x)}{\delta} + As + Br - rs\delta$$

$0/1 \leftarrow \mathrm{Vfy}(R, \sigma, a_1, \ldots, a_\ell, \pi)$：解析 $\pi = ([A]_1, [C]_1, [B]_2) \in \mathbb{G}_1^2 \times \mathbb{G}_2$. 当且仅当

$$[A]_1 \cdot [B]_2 = [\alpha]_1 \cdot [\beta]_2 + \sum_{i=0}^{\ell} a_i \left[\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma}\right]_1 \cdot [\gamma]_2 + [C]_1 \cdot [\delta]_2$$

时接受证明。

$\pi \leftarrow \mathrm{Sim}(R, \tau, a_1, \ldots, a_\ell)$：选择 $A, B \leftarrow \mathbb{Z}_p$ 并计算一个模拟的证明 $\pi = ([A]_1, [C]_1, [B]_2)$，其中 C 为

$$C = \frac{AB - \alpha\beta - \sum_{i=0}^{\ell} a_i \left(\beta u_i(x) + \alpha v_i(x) + w_i(x)\right)}{\delta}$$

定理 2 *以上协议是一个非交互零知识论证，具有完美完备性和完美零知识性。同时它对只能进行多项式次数通用双线性群操作的敌手具有统计知识合理性。*

证明 很容易看到这个非交互论证编码了一个 split NILP。为了应用引理 1，唯一剩下的就是证明 CRS 是 disclosure-free 的。我们观察到 CRS σ_1 和 σ_2 包含在 $\mathbb{Z}_p^*$ 上评估的多变量 Laurent 多项式。$\sigma_1 \cdot T\sigma_2$ 形式的测试可以评估为零，因为对应形式的多变量 Laurent 多项式为零，或者因为它

是一个非零的 Laurent 多项式恰好可以评估在输入变量的具体选择中为零。根据 Schwartz-Zippel 引理的直接扩展，后一种情况仅以可忽略的概率发生，因为负和正总阶数在 λ 中是多项式有界的。剩下的可能性是测试对应于形式上的零多项式，但在这种情况下，任何其他 CRS σ_1', σ_2' 也会有 $\sigma_1' \cdot T\sigma_2' = 0$。 □

对称双线性群（SYMMETRIC BILINEAR GROUPS.）非交互论证系统也能在对称双线性群 $\mathbb{G}_1 = \mathbb{G}_2$ 且 $g = h$ 上工作。在这种情况中，CRS 包括了在 $[\sigma_1]_1$ 和 $[\sigma_2]_2$ 上的组合，证明和验证等式的计算与上面所描述的一致。

高效性证明大小是 $\mathbb{G}_1$ 上的 2 个元素和 $\mathbb{G}_2$ 上的 1 个元素。CRS 包括了关系 R 的描述，$\mathbb{Z}_p$ 上的 n 个元素，$\mathbb{G}_1$ 上的 $m + 2n + 3$ 个元素，$\mathbb{G}_2$ 上的 $n + 3$ 个元素。

验证者没必要知道完整的 CRS，他只需知道

$$\sigma_V = \left(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, [1]_1, \left\{ \left[\frac{\beta u_i(x) + \alpha v_i(x) + w_i(x)}{\gamma} \right]_1 \right\}_{i=0}^{\ell}, [1]_2, [\gamma]_2, [\delta]_2, [\alpha\beta]_T \right).$$

就足够。验证者的 CRS 只包括 $\mathbb{G}_1$ 上的 $\ell + 2$ 个元素，$\mathbb{G}_T$ 上的 1 个元素。

验证过程包括了检查证明是否包含了 3 个合适的群元素，以及检查一个配对乘积方程。验证者计算 $\mathbb{G}_1$ 上的 ℓ 指数，少量群元素的乘法，以及 3 个配对计算 (假设 $[\alpha\beta]_T = [\alpha]_1 \cdot [\beta]_2$ 在验证者的参考字符串中已经被计算)。

证明者必须计算多项式 $h(X)$，对 $q = 1, \ldots, n$，计算多项式的评估值上

$$\sum_{i=0}^{m} a_i u_i(r_q) = \sum_{i=0}^{m} a_i u_{i,q} \quad \sum_{i=0}^{m} a_i v_i(r_q) = \sum_{i=0}^{m} a_i v_{i,q} \quad \sum_{i=0}^{m} a_i w_i(r_q) = \sum_{i=0}^{m} a_i w_{i,q}$$

这取决于这个关系式计算所需的时间。如果它产生于一个算术电路，其中每个乘法门连接到常数根电路，那么这种关系将是稀疏的，计算将是关于 n 的线性复杂度。由于多项式的阶为 $n-1$，它们完全由这些评估的点决定。如果 $r_1 \ \ldots, r_n$ 是一个合适素数 p 的单位根，她可以在 $\mathbb{Z}_p$ 的 $O(n \log n)$ 操作中使用标准的快速傅里叶变换技术计算 $h(X)$。证明者还可以使用 FFT 技术计算 $\sum_{i=0}^{m} a_i u_i(X)$ 和 $\sum_{i=0}^{m} a_i v_i(X)$ 的系数。有了所有的系数，证明者在 $\mathbb{G}_1$ 中做 $m + 3n - \ell + 3$ 的指数运算，在 $\mathbb{G}_2$ 中做 $n + 1$ 的指数运算。

渐近地，随着安全参数的增加，指数运算是主要的代价。然而，在实践中，FFT 计算中的乘法对于一般大小的安全参数和大的 statement 可能代价更高。在这种情况下，它可能值得使用一个更大的 CRS，对 $i = 0 \ \ldots, m$ 包含预先计算的 $[u_i(x)]_1$ $[v_i(x)]_1$ $[v_i(x)]_2$ 元素，让 a 和 B 可以直接构造，而不是验证者必须计算 $\sum_{i=0}^{m} u_i(x)$ 和 $\sum_i v_i(x)$ 的系数，然后做指数运算。在布尔电路的情况下，我们有 $a_i \in \{0, 1\}$，当计算 A 和 B 时，证明者可以用这些预先计算的元素为每个元素只做 m 次群上的乘法。出于这个原因，我们在表 1 中让 CRS 更长，以获得较低的计算成本。[2]

[2]由于修改后能加快证明者计算的 CRS 可以从原始 CRS 中计算，因此安全证明仍然适用，我们获得了针对通用对手的知识合理性。我们注意到，如果非交互论证在标准模型中具有知识合理性，那么修改了 CRS 后在标准模型中也具有知识合理性，假设我们仍然将原始的 *CRS* 提供给提取器。)

3.3 非交互论证的下界

一个有趣的问题是，非交互论证能有多高效。我们现在将给出一个下界，表明基于配对的非交互论证在证明中必须至少有两个群元素。更准确地说，我们研究基于配对的论证，其中 CRS 中包含双线性群和一些群元素的描述，证明由证明者使用通用群操作计算的多个群元素组成，验证者使用通用双线性群操作检查证明。我们将说明，对于这种基于配对的论证系统，如果语言包含如下定义的困难决策问题，那么证明需要包含来自 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 的元素。

考虑关系 R 上的采样问题，有两个采样的算法 Yes 和 No。Yes 采样关系中的 statement 和 witnesse。No 采样关系所定义的语言 L_R 外的 statement。我们感兴趣的关系是，很难判断一个 statement ϕ 是否被 Yes 或 No 采样。

定义 5 *如果有两个多项式算法满足对于 $(R,z) \leftarrow \mathcal{R}\left(1^\lambda\right)$，我们有压倒性的概率使得 $\mathrm{Yes}(R) \rightarrow (\phi,w) \in R$ 且 $\mathrm{No}(R) \rightarrow \phi \notin L_R$，对所有非均匀多项式时间区分的敌手 $\mathcal{A}$ 有*

$$\Pr\left[(R,z) \leftarrow \mathcal{R}\left(1^\lambda\right); \phi_0 \leftarrow \mathrm{No}(R); (\phi_1,w_1) \leftarrow \mathrm{Yes}(R); b \leftarrow \{0,1\} : \mathcal{A}\left(R,z,\phi_b\right) = b\right] \approx \frac{1}{2}.$$

那么我们说关系生成器 $\mathcal{R}$ 是决策困难的。

如果单向函数存在，我们可以构造伪随机生成器。伪随机生成器用来生成一个伪随机字符串，作为一个 Yes 的实例，其所用的种子是 witness。为了得到一个 No 的实例，我们采样一个均匀随机字符串，该字符串具有压倒性的概率不是伪随机。如果关系 R 是 NP 完全，或者仅足够表达伪随机生成器，那么它就有一个困难的决策问题。特别是，当我们处理基于配对的论证时，我们必须至少假设离散对数问题是困难的，因而存在关系生成器与决策困难问题。

3.4 线性交互证明没有线性决策过程

我们将证明 NILP 没有 1 阶验证者。这是 Bitansky 等人 [BCI$^+$13] 提出的一个开放性问题。即使我们考虑指定验证者 NILP，给出证明者不可见的 σ_V，结果仍然成立。我们只考虑现在定义的较弱的合理性概念，而不是知识合理性。

定义 6（*对仿射证明者策略的统计合理性*）*如果一个 LIP 对所有的敌手 $\mathcal{A}$ 有*

$$\Pr\left[\begin{array}{c}(R,z) \leftarrow \mathcal{R}\left(1^\lambda\right); (\boldsymbol{\sigma}_P, \boldsymbol{\sigma}_V, \boldsymbol{\tau}) \leftarrow \mathrm{Setup}(R); (\phi, \Pi) \leftarrow \mathcal{A}(R,z) \\ \pi = \Pi \boldsymbol{\sigma}_P; \boldsymbol{t} \leftarrow \mathrm{Test}\left(R,\phi,\sigma_V\right) : \phi \notin L_R \wedge \boldsymbol{t}(\boldsymbol{\pi}) = \mathbf{0}\end{array}\right] \approx 0$$

我们说它对仿射证明者策略是合理的。

定理 3 *对于决策困难问题的关系生成器，不存在 1 阶验证者的 NILP。*

证明 根据定义，1 阶的 NILP 有一个决策过程，产生一个算术电路 $t : \mathbb{F}^k \rightarrow \mathbb{F}^\eta$，评估 1 阶多项式，测试是否有 $\boldsymbol{t}(\boldsymbol{\pi}) = \mathbf{0}$。多项式阶为 1，因此有效计算矩阵 $A \in \mathbb{F}^{\eta \times k}$ 和向量 $\boldsymbol{b} \in \mathbb{F}^\eta$，使它们满足对应的检查 $A\boldsymbol{\pi} = \boldsymbol{b}$ 是可能的。

我们现在构造一个算法 $\mathcal{A}$，给定 R 和 ϕ，很有可能确定 $\phi \in L_R$ 中还是 $\phi \notin L_R$。在我们 NILP 的定义中，证明者和合理性的敌手独立于 σ_P 和 σ_V 选择证明矩阵 Π 是关键的。其思想是重复运行一个诚实的验证者，创建许多 CRS 和验证。当 $\phi \in L_R$ 时，相同的证明矩阵 Π 将对所有诚实运行的验证者给出合法证明。另一方面，当 $\phi \notin L_R$，合理性使得同样的 Π 不太可能通过多次测试。

我们现在给出一些细节。首先 $\mathcal{A}(R,\phi)$ 运行 setup$N = mk \log |F|$ 次，得到 $(\boldsymbol{\sigma}_{1,P}, \boldsymbol{\sigma}_{1,V}), \ldots, (\boldsymbol{\sigma}_{N,P}, \boldsymbol{\sigma}_{N,V})$。然后对每个 CRS 对应的 statement ϕ 创建测试 $A_i \in \mathbb{F}^{\eta \times k}$ 和 $b_i \in \mathbb{F}^{\eta}$。由完备性，如果我们有 ϕ 对应的 witness，我们能够计算一个证明矩阵 Π，满足对所有 N 个测试有 $A_i \Pi \sigma_{i,P} = b_i$。算法 $\mathcal{A}$ 不知道 $\phi \in L_R$ 的 witness，但它可以解决一系列线性方程组，看看是否存在这样的 Π。如果存在，它输出 1 表示 $\phi \in L_R$，否则输出 0 表示 $\phi \notin L_R$。 □

我们现在分析决策算法 $\mathcal{A}$ 成功的概率。输入 $\phi \in L_R$，NILP 的完备性意味着存在这样的 Π，因为方程组是线性的，所以能够有效求解。因此，这个决策算法当 $\phi \in L_R$ 时输出 1。输入 $\phi \notin L_R$，NILP 的合理性意味着 Π 的任意选择只有低概率通过验证。它通过所有 $N = mk \log |F|$ 验证的概率上界为 $negl(\lambda)^{mk \log |F|}$。有 $|F|^{mk}$ 种可能的 Π 的选择，因此对任意 Π 通过所有测试概率是可忽略的。因此，当 $\phi \notin L_R$ 时，决策算法以压倒性的概率输出 0。

定理 3 的证明对于 split NILP 也成立，如果我们限制合理性对手产生 split 的矩阵 Π_1, Π_2，这只是证明者和敌手可以产生的证明矩阵的一个额外限制。在第 2.5 中，我们从 disclosure-free split NILP 构建了一个基于配对的 SNARK。一般的合理性敌手无法知道配对设置中 CRS 中有用的信息，这是 disclosure-free 的。因此，它所能做的就是选择一个 statement $\phi \notin L_R$ 和独立于 σ 的证明矩阵 Π，尝试通过验证。由于单元素证明对应于第三类配对设置中的线性决策过程，我们得到如下推论:

推论 1 *在第 2.5 节中描述的由 disclosure-free split NILP 构造的第 III 类型设置中具有基于配对的 SNARK 的关系生成器必须在证明中至少有两个元素，以使语言是非平凡的。*

3.5 基于配对的通用非交互论证的大小下界

现在，我们概括这样一个 statement：不要求 CRS 是 disclosure-free 的，第 III 类型的群上基于配对的非交互论证必须在 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 中都有元素。然而，论证背后的直觉仍然是一样的：如果我们有一个单方面论证，其中只有 $\mathbb{G}_1$ 中的元素，或者只有 $\mathbb{G}_2$ 中的元素，那么验证方程就变成线性的，就有可能违背合理性。对于一般情况，我们表明即使 CRS 和证明包含 $\mathbb{G}_T$ 中的元素，这也成立。这一结果意味着在基于配对的非交互论证中至少有两个群元素的下界。

我们将考虑基于配对的论证系统 (Setup, Prove, Vfy)，其中 CRS 和证明由使用通用群操作计算的群元素组成，验证者使用通用双线性群操作来测试证明。我们通过创建一些配对乘积方程并当它们全部成立时接受，来限制验证者来检验证明的有效性。所有已知的基于配对的 SNARK 都满足这个限制。该限制排除了违反说该论点是" 基于配对" 背后的意图。例如，如果 CRS 和证明的群元素可以表示位串，$\left(G^0\ G^1\ \ldots, G^1\ G^0\right)$，那么我们可以想象验证者将读取 CRS 中的位串，使用它来

创建一个基于非配对的 SNARK，将其编码为位串发送给验证者，验证者将解码证明中的位串并检查它。显然，这只是对不同类型的 SNARK 进行编码的一种非常麻烦的方式，并且不能被认为是基于配对的。

让我们明确说明上面所述基于配对的非交互论证的含义，以及使用通用群操作的结果。
$(\sigma, \tau) \leftarrow \text{Setup}(R)$：这个关系包括双线性配对 $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ 并且 CRS 中包含 $\mathbb{G}_1, \mathbb{G}_2$ 和 $\mathbb{G}_T$ 上的群元素，如 $\sigma = ([\boldsymbol{\sigma}_1]_1, [\boldsymbol{\sigma}_2]_2, [\boldsymbol{\sigma}_T]_T)$ 。
$\pi \leftarrow \text{Prove}(R, \sigma, \phi, w)$：证明者使用通用群操作去创建证明。这意味着他选择矩阵 Π_1, Π_2 和 Π_T，并通过设置计算证明。[3]

$$\pi = (\Pi_1 [\boldsymbol{\sigma}_1]_1, \Pi_2 [\boldsymbol{\sigma}_2]_2, \Pi_T [\boldsymbol{\sigma}_T]_T)$$

注意到我们没有假设 CRS 是 disclosure-free 的，所以 Π_1, Π_2, Π_T 与 $\sigma_1, \sigma_2, \sigma_T$ 相关是可能的。
$0/1 \leftarrow \text{Vfy}(R, \sigma, \phi, \pi)$：验证者分两步验证。首先，它生成矩阵和向量 $\{T_i, \boldsymbol{t}_i\}_{i=1}^{\eta}$。它独立于证明选择这些矩阵和向量，但会使用到 statement 和 CRS。当且仅当所有的配对乘积方程成立时

$$\left[\boldsymbol{\sigma}_1^\top, \boldsymbol{\pi}_1^\top\right]_1 \cdot T_i \begin{bmatrix} \boldsymbol{\sigma}_2 \\ \pi_2 \end{bmatrix} = \boldsymbol{t} \cdot \begin{bmatrix} \boldsymbol{\sigma}_T \\ \boldsymbol{\pi}_T \end{bmatrix}_T$$

它接受证明。

定理 4 *对于具有决策困难问题的关系生成器，除非证明在 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 中都有元素，否则上面描述的基于配对的非交互论证是不存在的。*

证明 我们假设有一个基于配对的非交互论证，如上所述，其中的证明在 $\mathbb{G}_1$ 中没有元素。同理可以证明 $\mathbb{G}_2$ 中没有元素的情况。这意味着证明 $[\Pi_2 \boldsymbol{\sigma}_2]_2, [\Pi_T \boldsymbol{\sigma}_T]_T$ 的形式，其中 Π_2, Π_T 是由一般证明者选择的矩阵。证明测试的矩阵和向量可以被重写为 $(A_1, B_1, \boldsymbol{c}_1, \boldsymbol{d}_1, \ldots, A_\eta\ B_\eta, \boldsymbol{c}_\eta, \boldsymbol{d}_\eta)$，验证者对其进行检查

$$[\boldsymbol{\sigma}_1]_1 \cdot A_i [\boldsymbol{\sigma}_2]_2 + [\boldsymbol{\sigma}_1]_1 \cdot B_i \Pi_2 [\boldsymbol{\sigma}_2]_2 = \boldsymbol{c}_i \cdot [\boldsymbol{\sigma}_T]_T + \boldsymbol{d}_i \cdot \Pi_T [\boldsymbol{\sigma}_T]_T$$

我们观察到该验证方程对应于 Π_2 和 Π_T 中的线性方程组。

我们将使用这样一个基于配对的论证系统来设计一个算法 $\mathcal{A}(R\ \phi)$，它得到一个 statement ϕ 作为输入，生成一个 Yes 实例，或者生成一个 No 实例，并决定是哪种情况。该算法分两个阶段：首先，它为自己选择的 Yes 实例生成大量诚实的证明，然后它检查验证方程的线性关系，以检查当前实例 ϕ 是否可以有类似于其他 Yes 实例的证明。如果 ϕ 是一个 Yes 实例，它可以有这样的结构，但如果 ϕ 是一个 No 实例，它就没有这样的结构。

[3]证明者也可以在证明中包含 G_1 和 $\mathbb{G}_T$ 中的元素对，但我们可以在不损失一般性的情况下假设 $[\sigma_1]_1$ 和 $[\sigma_2]_2$ 中所有可能的元素对都包含在 $[\sigma_T]_T$ 中。

对第一阶段,这个算法采样多个 Yes 实例 $(\phi_j, w_j) \leftarrow Yes(R)$。然后它生成一个 $\mathrm{CRS}[\boldsymbol{\sigma}_1]_1, [\boldsymbol{\sigma}_2]_2, [\boldsymbol{\sigma}_T]_T$，创建证明矩阵 $\Pi_{j,2}, \Pi_{j,T}$，以及对所有的 statement 的验证测试 $(A_{j,1}, B_{j,1}, \boldsymbol{c}_{j,1}, \boldsymbol{d}_{j,1}, \ldots, A_{j,\eta}, B_{j,\eta}, \boldsymbol{c}_{j,\eta}, \boldsymbol{d}_{j,\eta})$。令 V 为 $\left(A_{j,1}, B_{j,1}\Pi_{j,2}, \boldsymbol{c}_{j,1}, \boldsymbol{d}_{j,1}^{\dagger}\Pi_T, \ldots, A_{j,\eta}, B_{j,\eta}\Pi_{j,2}, \boldsymbol{c}_{j,\eta}, \boldsymbol{d}_{j,\eta}^{\top}\Pi_{j,T}\right.$ 生成的向量空间。

该算法进行多次采样，直到 Yes-instances ϕ_j 某一行 λ 给出了已经在 V 中的向量。向量空间具有多项式维数，所以这个过程终止于多项式时间。然后，Chernoff 界告诉我们，至少有 50% 的概率，Yes 实例 ϕ 会产生一个以 V 为单位的向量。当然，即使 ϕ 是一个 Yes 实例，算法也不知道对应的 witness。

算法现在进入第二阶段。给定 ϕ，它创建测试 $(A_1, B_1, \boldsymbol{c}_1, \boldsymbol{d}_1, \ldots, A_\eta, B_\eta)$，然后它尝试解出 Π_2, Π_T，来满足 $\left(A_1, B_1\Pi_2, \boldsymbol{c}_1, \boldsymbol{d}_1^{\top}\Pi_T, \ldots, A_\eta, B_\eta\Pi_2, \boldsymbol{c}_\eta, \boldsymbol{d}_\eta^{\top}\Pi_T\right)$ 属于向量空间 V。这是一个线性方程组，所以它可以有效求解。如果算法成功解出 Π_2, Π_T 它返回 1，否则返回 0。

下面我们来分析一下算法的成功概率。如果采样 Yes 实例 ϕ，算法有至少 50% 的机会找到 Π_2, Π_T 产生一个 V 中的向量。另一方面，如果采样 No 实例 ϕ，合理性意味着找到这样一个 Π_2, Π_T 的概率可以忽略不计。我们注意到合理性依然成立，因为算法在生成设置 $[\sigma_1]_1, [\sigma_2]_2, [\sigma_T]_T$ 后诚实地运行了一般证明者和验证者几次，但从来没有使用关于潜在的离散对数值 $\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \boldsymbol{\sigma}_T$ 的特殊知识，除了诚实的证明者和验证者可能通过通用算法学到的东西。 □

推论 2 具有所述通用群算法的基于配对的非交互论证在证明中必须至少有两个群元素。

致谢

我们感谢 Alessandro Chiesa 和 Madars Virza 对本文早期版本的广泛评论，以及他们 [CV16] 在 libsnark 库中对 SNARK 的实现和分析。我们还感谢 Eran Tromer 和 Michael Walfish 对 SNARK 实现的性能进行了有趣的讨论，以及匿名评论者的评论。

参考文献

[AF07] Masayuki Abe and Serge Fehr. Perfect nizk with adaptive soundness. In *Theory of Cryptography Conference*, pages 118–136. Springer, 2007.

[AGOT14] Masayuki Abe, Jens Groth, Miyako Ohkubo, and Mehdi Tibouchi. Unified, minimal and selectively randomizable structure-preserving signatures. In *Theory of Cryptography Conference*, pages 688–712. Springer, 2014.

[BBFR15] Michael Backes, Manuel Barbosa, Dario Fiore, and Raphael M Reischuk. Adsnark: nearly practical and privacy-preserving proofs on authenticated data. In *2015 IEEE Symposium on Security and Privacy*, pages 271–286. IEEE, 2015.

[BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical identity based encryption with constant size ciphertext. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 440–456. Springer, 2005.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 326–349, 2012.

[BCCT13] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive composition and bootstrapping for snarks and proof-carrying data. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 111–120, 2013.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Omer Paneth, and Rafail Ostrovsky. Succinct non-interactive arguments via linear interactive proofs. In *Theory of Cryptography Conference*, pages 315–333. Springer, 2013.

[BCPR14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the existence of extractable one-way functions. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 505–514, 2014.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 103–112, 1988.

[BFR+13] Benjamin Braun, Ariel J Feldman, Zuocheng Ren, Srinath Setty, Andrew J Blumberg, and Michael Walfish. Verifying computations with state. In *Proceedings of the twenty-fourth ACM Symposium on Operating Systems Principles*, pages 341–357, 2013.

[BP15] Elette Boyle and Rafael Pass. Limits of extractability assumptions with distributional auxiliary input. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 236–261. Springer, 2015.

[BSCG+13] Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for c: Verifying program executions succinctly and in zero knowledge. In *Annual cryptology conference*, pages 90–108. Springer, 2013.

[BSCG+14] Eli BenSasson, Alessandro Chiesa, Daniel Genkin, Shaul Kfifir, Tromer Eran, and Madars Virza. libsnark. 2014.

[BSCTV14a] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. *Cryptology ePrint Archive*, 2014.

[BSCTV14b] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Succinct {Non-Interactive} zero knowledge for a von neumann architecture. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 781–796, 2014.

[CFH+15] Craig Costello, Cédric Fournet, Jon Howell, Markulf Kohlweiss, Benjamin Kreuter, Michael Naehrig, Bryan Parno, and Samee Zahur. Geppetto: Versatile verifiable computation. In *2015 IEEE Symposium on Security and Privacy*, pages 253–270. IEEE, 2015.

[CTV15] Alessandro Chiesa, Eran Tromer, and Madars Virza. Cluster computing in zero knowledge. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 371–403. Springer, 2015.

[CV16] Alessandro Chiesa and Madars Virza. Personal communication. 2016.

[DFGK14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct nizk arguments. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 532–550. Springer, 2014.

[DFKP13] George Danezis, Cedric Fournet, Markulf Kohlweiss, and Bryan Parno. Pinocchio coin: building zerocoin from a succinct pairing-based proof system. In *Proceedings of the First ACM workshop on Language support for privacy-enhancing technologies*, pages 27–30, 2013.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 626–645. Springer, 2013.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In *Annual International Cryptology Conference*, pages 97–111. Springer, 2006.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New techniques for noninteractive zero-knowledge. *Journal of the ACM (JACM)*, 59(3):1–35, 2012.

[GPS08] Steven D Galbraith, Kenneth G Paterson, and Nigel P Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[Gro06] Jens Groth. Simulation-sound nizk proofs for a practical language and constant size group signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 444–459. Springer, 2006.

[Gro09] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *Annual International Cryptology Conference*, pages 192–208. Springer, 2009.

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 321–340. Springer, 2010.

[GS12] Jens Groth and Amit Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM Journal on Computing*, 41(5):1193–1232, 2012.

[GW11] Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 99–108, 2011.

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 723–732, 1992.

[Kil95] Joe Kilian. Improved efficient arguments. In *Annual International Cryptology Conference*, pages 311–324. Springer, 1995.

[KPP+14] Ahmed E Kosba, Dimitrios Papadopoulos, Charalampos Papamanthou, Mahmoud F Sayed, Elaine Shi, and Nikos Triandopoulos. Trueset: Faster verifiable set computations. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 765–780, 2014.

[Lip12] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In *Theory of Cryptography Conference*, pages 169–189. Springer, 2012.

[Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 41–60. Springer, 2013.

[Mic00] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2000.

[Nec94] Vassiliy Ilyich Nechaev. Complexity of a determinate algorithm for the discrete logarithm. *Mathematical Notes*, 55(2):165–172, 1994.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE, 2013.

[SCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE symposium on security and privacy*, pages 459–474. IEEE, 2014.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 256–266. Springer, 1997.

[SVV16] Berry Schoenmakers, Meilof Veeningen, and Niels de Vreede. Trinocchio: Privacy-preserving outsourcing by distributed verifiable computation. In *International Conference on Applied Cryptography and Network Security*, pages 346–366. Springer, 2016.

[Val08] Paul Valiant. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency. In *Theory of Cryptography Conference*, pages 1–18. Springer, 2008.

[Wal15] Michael Walfish. A wishlist for verifiable computation: An applied cs perspective. 2015.

[WB15] Michael Walfish and Andrew J Blumberg. Verifying computations without reexecuting them. *Communications of the ACM*, 58(2):74–84, 2015.

[WSR+15] Riad S Wahby, Srinath Setty, Zuocheng Ren, Andrew J Blumberg, and Michael Walfish. Efficient ram and control flow in verifiable outsourced computation. 2015.