证明、论证与零知识

Kurt Pan

2023 年 5 月 13 日

目录

Chapter 1

引言

本书讨论了可验证计算（VC）。可验证计算是指一种名为交互式证明（IPs）和论证的密码学协议，证明者能够向验证者提供证明者正确地执行了所请求的计算的保证。交互式证明和论证在 20 世纪 80 年代被提出，代表了一个陈述为真的“证明”包括什么概念上的重大扩展。传统上的证明是一个静态对象，可以很容易地通过逐步检查来检验正确性，因为证明的每个单独步骤都应该易于验证。相比之下，交互式证明允许证明者和验证者之间的进行交互，以及允许无效证明以微小但非零的概率通过验证。交互式证明和论证之间的区别在于，论证（而非交互式证明）允许存在对错误陈述的“证明”，只要找到这些“证明”需要巨大的计算能力就可以。[1]

20 世纪 80 年代中期和 90 年代初的著名理论结果表明，至少原则上讲，可验证计算协议可以达到惊人的成就：包括让手机可以去监控强大但不受信任（甚至恶意）的超级计算机的执行，让计算能力较弱的外设（例如，安全卡读卡器）将安全核心工作外包给强大的远程服务器，以及让数学家仅通过查看所谓证明中的几个符号就能对定理的正确性具有很高的信心。[2]

当 VC 协议具有一种称为零知识的性质时，它会在密码学环境中非常有用。零知识的意思是证明或论证除了其本身的有效性之外，不会泄露任何信息。

为了具体说明为什么零知识协议有用，考虑以下来自身份认证的典型例子。假如 Alice 选择了一个随机口令 x，公开了一个哈希值 $z = h(x)$。其中 h 是一个单向函数，给定一个对随机选择 x 的 $z = h(x)$，需要大量计算能力才能找到 z 在 h 下的原像，即一个满足 $h(x') = z$ 的 x'。假如 Alice 在之后想要说服 Bob 她是发布 z 的同一个人。她可以通过向 Bob 证明她知道一个满足 $h(x') = z$ 的 x' 来实现这一点。这将使 Bob 相信 Alice 是发布 z 的同一个人，因为这意味着 Alice 要么一开始就知道 x，要么她反转了 h （这被认为是超出了 Alice 的计算能力）。

Alice 如何让 Bob 相信她知道 h 下 z 的原像？一个简单的证明是 Alice 将 x 发送给 Bob，Bob 可

[1] 比如说，一个不是 IP 的论证系统，可能会使用密码系统，使得作弊的证明者有可能找到一个可通过验证的对假陈述的“证明”，当且仅当证明者可以攻破密码系统。

[2] 只要证明以一种特定的、略微有些冗余的形式书写。具体见第9章中对概率可检验证明（PCPs）的讨论。

以轻松检验 $h(x) = z$。但这样所揭示的信息比 Alice 知道 z 的原像要多得多。具体来说它揭示了原像本身。Bob 可以使用该知识永远冒充 Alice，因为现在他也知道 z 的原像了。

为了防止 Bob 获悉可能泄露口令 x 的信息，给出不能揭示任何超出其自身有效性的内容的证明是很重要的。这正是零知识性质所保证的。

本书的一个特定目标是描述构造所谓的零知识简洁非交互式知识论证（简称 zk-SNARKs）的各种方法。“简洁”意味着证明很短。“非交互”意味着证明是静态的，由来自证明者的单个消息组成。（第二个）“知识”大致意味着协议不仅确定一个陈述为真，而且证明者知晓一个“证据”来证明该陈述为真。[3]满足所有这些性质的论证系统在整个密码学中有无数应用。

对密码学相关的高度特定化的陈述（如证明离散对数的知识 [Sch89]）的实用零知识协议已经有了数十年了。然而，通用零知识协议直到最近才变得足够高效，可以用于密码部署中。通用的意思是，协议设计技术适用于任意计算。这一令人兴奋的进展包括了漂亮的新协议的提出，并引发了各界对零知识证明和论证的浓厚兴趣。本书旨在以统一的方式让人们能够轻松理解这些协议设计的主要思想和方法。

背景和上下文 在 20 世纪 80 年代中期和 90 年代，理论计算机科学家理论计算机科学家发现 IP 和论证系统可以比传统的静态和信息论安全 [4]的 **NP** 证明（至少在渐近意义上）更高效。刻画这些协议能力的基础定理（例如 **IP** = **PSPACE** [LFK+92; Sha92]、**MIP** = **NEXP** [BFL91] 和 PCP 定理 [ALM+98; AS98]）都是计算复杂性理论中最具影响力和最著名的定理。[5]

零知识协议设计方法及本书的理念 论证系统通常分两步开发。开发一个信息论安全协议（例如 IP、多证明者交互式证明 (MIP) 或概率可检查证明 (PCP)），以对单个或多个证明者假定以某种受限方式行事建模。(例如，在 MIP 中，会假定证明者不会相互发送有关他们从验证者那里收到的挑战的信息)。第二步，信息论安全的协议与密码学相结合，以“强制”（单个）证明者以受限方式行事，从而产生一个论证系统。这第二步通常还会赋予所生成的论证系统重要的性质，例如零知识、简洁性和非交互性。如果生成的论证系统满足所有这些性质，那么它实际上是一个 zk-SNARK。

本书中信息论安全模型的顺序 我们首先介绍 IP，然后是 MIP，然后是 PCP 和 IOP，再然后是线性 PCP。该顺序大致遵循研究文献中模型出现的时间顺序。有点讽刺的是，这些模型是以类似于倒时间序的方式应用于实际的 SNARK 设计中的。比如说第一个实用的 SNARK 就是基于线性 PCP 的。

[3]例如，上述身份验证场景中确实需要对“存在一个口令 x 使得 $h(x) = z$”这一陈述进行零知识知识证明。这是因为应用要求 Bob 不仅要确信在 h 下存在z 的原像 x（如果 h 是满射函数，这将始终为真），而且还要确信 Alice 知道 x。

[4]信息论安全

[5]The results IP=PSPACE and MIP=NEXP are both covered in this survey (see Sections 4.5.5 and 8.5 respectively).

事实上这并非巧合：引入线性 PCP 的主要动机就是为了得到更简单和更实用的简洁论证，特别是考虑到从 PCP 导出的论证的实际不可用性。

逐章大纲

阅读本书的建议

首次阅读可跳过的材料 Sections 4.2-4.5 are devoted to detailed example applications of the sum-check protocol and explaining how to efficiently implement the prover within it. While these sections contain interesting results and are useful for familiarizing oneself with the sum-check proto- col, subsequent chapters do not depend on them. Similarly, Chapter 5 on the Fiat-Shamir transformation and Chapter 6 on front-ends are optional on a first reading. Sections 9.3 and 9.4 provide PCPs that are mainly of historical interest and can be skipped. Chapters 11 and 13 offer treatments of zero-knowledge that largely stand on their own. Similarly, Chapter 18 discusses SNARK composition and stands on its own.

1.1 数学证明

本书涵盖了数学证明的不同概念及其在计算机科学和密码学中的应用。非正式地，我们所说的证明是指任何能使某人相信某项陈述为真的东西，而“证明系统”是指任何判定什么是令人信服的证明的程序。也就是说，证明系统由验证程序决定，该验证程序将任何陈述和声称该陈述为真的“证明”作为输入，判定该证明是否有效。

我们希望证明系统具有哪些性质呢？这有四个比较明显要有的。

- 任何真陈述都应该有对其有效性的令人信服的证明。此性质通常称为完备性。
- 任何假陈述都不应有令人信服的证明。此性质称为可靠性。
- 理想情况下，验证程序将是“高效的”。粗略地说，这意味着对简单的陈述应该有简短的（令人信服的）证明，且可以快速检验。
- 理想情况下，证明生成也应该是高效的。粗略地说，这意味着对简单的陈述应该有可以快速找到的（令人信服的）短证明。

传统上，数学证明是可以逐行撰写并逐行检验其正确性的东西。这种传统的证明概念正是复杂性

类 **NP** [6] 所刻画的概念。但是在过去的 30 多年里，计算机科学家研究了更通用和更奇异的证明概念。这改变了计算机科学家对证明某事意味着什么的观念，并导致了复杂性理论和密码学的重大进步。

1.2 我们将研究哪些非传统证明？

本书中研究的所有证明概念本质上都是概率性的。即验证程序将做出随机选择，且可靠性将以（非常）高的概率在这些随机选择上成立。也就是说，验证程序会以（非常）小的概率宣布假陈述为真。

1.2.1 交互式证明（IP）

要理解什么是交互式证明，考虑以下应用场景可能会有帮助。想象一家企业（验证者）使用某云计算提供商来存储和处理其数据。企业将其所有数据发送并存储到云（证明者），企业仅存储非常小的数据的“秘密”摘要（云不知道用户的秘密摘要）。随后，企业向云询问有关其数据的问题，通常以计算机程序 f 的形式提出，即企业希望云使用庞大的计算基础设施在其数据上运行该程序。云这样做了之后向用户发送程序的输出 f(数据)。企业可以使用交互式证明系统 (IP) 来获得该输出正确性的一个正式保证，而不是盲目地去相信云在数据上正确执行了该程序。

在 IP 中，企业询问云，发送一系列挑战并接收一系列响应。在询问结束时，企业必须决定是接受有效答案还是拒绝无效答案。有关此交互的图，请参见图 1.1。

IP 的完备性保证了如果云在数据上正确运行了程序并遵循所规定的协议，那么用户将被说服接受有效的回答。IP 的可靠性保证了如果云返回错误的输出，那么无论云如何努力诱使用户接受回答为有效，用户都将以很高概率拒绝回答为无效。直观上，IP 的交互本质使得企业可以利用意外元素（即云无法预测企业的下一个挑战这一事实）来捕捉到云的谎言。

值得一提的是 IP 和传统静态证明之间的一个有趣区别。静态证明是可传递的，如果 Peggy（证明者）向 Victor（验证者）提供一个陈述为真证明，Victor 可以继续去说服 Tammy（第三方）相同的陈述为真，只需复制证明即可。反之，一个交互式证明是不可传递的。Victor 可以通过向 Tammy 发送他与 Peggy 的交互脚本来尝试说服 Tammy 该陈述为真，但 Tammy 并不会被说服，除非 Tammy 相信 Victor 正确地代表了交互过程。这是因为 IP 的可靠性仅在每次 Peggy 向 Victor 发送响应时，Peggy 不知道 Victor 接下来将响应什么挑战时才成立。仅凭脚本并不能向 Tammy 保证这一点。

[6] 粗略地说，复杂性类 **NP** 包含所有这样的问题：任意输入上的正确答案为“是”或“否”，且对于所有“是”实例，存在一个有效可检验的（传统）证明，该证明的正确答案为“是”。详细信息可参阅第 3.1 节。

1.2.2 论证系统

论证系统就是 IP，只是其中的可靠性性要求只需要防止在多项式时间内运行的作弊证明者。[7] 论证系统会使用密码学。粗略地说，在论证系统中，作弊证明者无法欺骗验证者接受假陈述，除非它攻破了某个密码系统，且破解该密码系统被认为需要超多项式时间。

1.2.3 多证明者交互式证明（MIP）/概率可检验证明（PCP）等

MIP 类似于 IP，但有多个证明者，并且假设这些证明者不会相互共享他们从验证者那里收到的挑战的信息。MIPs 的一个常见类比是在审讯之前将两名或更多犯罪嫌疑人放在不同的房间里，看他们是否能讲出一致的故事。执法人员可能不会对 MIPs 的研究为这种做法提供了理论依据感到太意外。具体来说，MIPs 的研究表明，将证明者锁在不同的房间里，然后分别审问他们，这可以说服审问者相信比在一起审问时更复杂的陈述。

在 PCP 中，证明与传统数学证明一样是静态的，但验证者只能从证明中读取少量（可能是随机选择的）字符。[8] 这类似于数学期刊的懒惰审稿人，他并不想费力地检查提交论文中的证明是否正确。PCP 定理 [ALM+98; AS98] 本质上指出，任何传统的数学证明都可以写成特定格式，使这个懒惰的审阅者能够仅通过检查其中的几个词而对证明的有效性获得高度信心。

哲学上讲，MIP 和 PCP 都是非常有趣的研究对象，但它们并不能直接应用于大多数密码学场景中，因为它们对证明者（们）做出了不切实际或繁琐的假设。例如，任何 MIP 的可靠性仅在证明者不相互共享有关他们从验证者那里收到挑战的信息时才成立。这在大多数密码学场景中并不直接有用，因为通常在这些场景中只有一个证明者，即使有多个证明者，也没有办法强制证明者不进行通信。类似地，虽然验证者只需读取 PCP 的几个字符，但一个 PCP 的直接实现却需要证明者将整个证明传输给验证者，这将是大多数现实场景中的主要开销所在（尽管例子中有一个懒惰的期刊审稿人）。也就是说，一旦证明者将整个证明传输给验证者了，再让验证者避免阅读整个证明几乎没有什么现实世界的好处。

然而，通过将 MIP 和 PCP 与密码学相结合，我们将看到如何将它们变成论证系统，这些论证系统可以直接应用于密码学场景。例如，我们将在 9.1 节中看到如何将 PCP 变成一个论证系统，在该系统中，证明者不必将整个 PCP 发送给验证者。

本书的第 10.1 节实际上提供了一个统一的抽象，称为多项式 *IOP*，在其中所有 IP、MIP 和 PCP 都是一个特例。实际上，通过称为多项式承诺方案的密码学原语，任何多项式 IOP 都可以转换为具有

[7] 这大致意味着如果输入的大小为 n，那么证明者的运行时间（对于足够大的 n 值）应该以 n 的某个常数幂为界，例如 n^{10}。

[8] 更准确地说，PCP 验证者去读证明中尽可能多的内容是可以的。然而，仅在验证者只需要阅读证明的一小部分即可高置信度地确定证明是否有效的情况下，才认为 PCP 是高效的。

短证明的论证系统。

Chapter 2

随机性的力量：指纹和 Freivalds 算法

Chapter 3

定义和技术预备知识

3.1 定义的健壮性及交互式的威力

Chapter 4

交互式证明系统

Chapter 5

使用 Fiat-Shamir 得到公开可验证非交互式论证

Chapter 6

前端：将计算机程序转化为电路

Chapter 7

基于交互式证明的首个电路可满足性简洁论证

Chapter 8

MIP 和简洁论证系统

Chapter 9

PCP 和简洁论证系统

9.1 将 PCP 编译为简洁论证系统

Chapter 10

交互式预言证明

10.1 多项式 IOP 和相应的简洁论证系统

Chapter 11

零知识证明和论证系统

Chapter 12

基于离散对数困难性的 Σ-协议和承诺

Chapter 13

通过承诺并证明和遮盖多项式实现零知识

Chapter 14

基于离散对数困难性的多项式承诺

Chapter 15

基于配对的多项式承诺

Chapter 16

多项式承诺总结

Chapter 17

线性 PCP 和简洁论证

Chapter 18

SNARK 组合和递归

Chapter 19

实用论证系统鸟瞰

Bibliography

[ALM+98] Sanjeev Arora et al. *Proof verification and the hardness of approximation problems*. Journal of the ACM (JACM) 45.3 (1998), pp. 501–555 (Cited on pages 6, 9).

[AS98] Sanjeev Arora and Shmuel Safra. *Probabilistic checking of proofs: A new characterization of NP*. Journal of the ACM (JACM) 45.1 (1998), pp. 70–122 (Cited on pages 6, 9).

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. *Non-deterministic exponential time has two-prover interactive protocols*. Computational complexity 1 (1991), pp. 3–40 (Cited on page 6).

[LFK+92] Carsten Lund et al. *Algebraic methods for interactive proof systems*. Journal of the ACM (JACM) 39.4 (1992), pp. 859–868 (Cited on page 6).

[Sch89] Claus-Peter Schnorr. *Efficient Identification and Signatures for Smart Cards*. Annual International Cryptology Conference. 1989 (Cited on page 6).

[Sha92] Adi Shamir. *IP = PSPACE*. Journal of the ACM (JACM) 39.4 (1992), pp. 869–877 (Cited on page 6).

BIBLIOGRAPHY