零知识证明递归与复合技术研究综述

张宗洋[1)] 周子博[1)] 邓 燚[2),3)]

[1)](北京航空航天大学网络空间安全学院 北京 100191)

[2)](中国科学院信息工程研究所网络空间安全防御重点实验室 北京 100093)

[3)](中国科学院大学网络空间安全学院 北京 100049)

摘 要 零知识证明作为一种重要的密码学协议,是实现数据安全流通的关键技术之一.其允许证明者向验证者证明某个断言的正确性,而又不泄露任何额外信息.零知识证明所描述的断言可划分成代数断言、非代数断言和复合断言,而递归与复合技术可以极大地提高零知识证明协议的性能并深入拓展其功能,是当前的研究热点.本文系统且全面地研究了零知识证明的递归与复合技术.首先,在针对代数断言的递归零知识证明方面,全面研究了关于内积关系的递归零知识证明协议,并从证明复杂度、通信复杂度、验证复杂度等角度对比分析了基于 Pedersen 承诺方案的内积论证协议.其次,在针对非代数断言的递归零知识证明方面,全面梳理了增量可验证计算方案与基于电路的证明系统组合这两种主流应用的研究现状,并对比分析了增量可验证计算方案的复杂度、关键技术及实现方案等.然后,在针对复合断言的复合零知识证明方面,从复杂度、启动阶段、关键模块等角度对比分析了承诺并证明的零知识证明协议.最后,给出了零知识证明递归与复合技术的未来研究方向.

关键词 零知识证明;递归零知识证明;内积论证;增量可验证计算方案;复合零知识证明;承诺并证明的零知识证明

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2024.02466

A Survey on Recursive and Composite Techniques of Zero-Knowledge Proofs

ZHANG Zong-Yang[1)] ZHOU Zi-Bo[1)] DENG Yi[2),3)]

[1)](*School of Cyber Science and Technology*, *Beihang University*, *Beijing* 100191)

[2)](*Key Laboratory of Cyberspace Security Defense*, *Institute of Information Engineering*, *Chinese Academy of Sciences*, *Beijing* 100093)

[3)](*School of Cyber Security*, *University of Chinese Academy of Sciences*, *Beijing* 100049)

Abstract Zero-knowledge proofs, as a fundamental cryptographic primitive, allow one party in a communication to prove the validity of a certain statement to another party without revealing any additional information. They are pivotal technologies for the secure circulation of data and play an extremely important role in cryptography and security. They are not only critical components of many cryptographic protocols, such as authentication, digital signature, and public-key encryption, but also provide significant privacy protection and performance enhancement for various practical applications, including anonymous transactions, smart contracts, machine learning, and more. In general, the statements to be proven by zero-knowledge proofs are categorized into three types: algebraic statements, non-algebraic statements, and composite statements involving both algebraic and non-algebraic statements. Although zero-knowledge proof protocols for these statements have made significant progress in terms of performance, functionality, etc., there remain challenges in practical applications, such as insufficient efficiency, poor scalability, and specialized functionality. To effectively address these bottleneck issues, the techniques of recursion and composition in zero-knowledge proofs have received extensive attention in recent years. In this paper, we conduct a

收稿日期:2023-12-18;在线发布日期:2024-06-20.本课题得到国家重点研发计划(2022YFB2702702)、国家自然科学基金项目(62372020,61972017,72031001,61932019,62372447)、北京市自然科学基金(M22038,L222050,M21033)、中央高校基本科研业务费(YWF-23-L-1032)、北航博士研究生国际联合培养基金资助.**张宗洋**,博士,副教授,中国计算机学会(CCF)高级会员,主要研究领域为密码学与区块链.E-mail:zongyangzhang@buaa.edu.cn.**周子博**(通信作者),博士研究生,主要研究方向为零知识证明与密码学.E-mail:zbzhou@buaa.edu.cn.**邓 燚**,博士,研究员,主要研究领域为零知识证明及其在金融科技中的应用.

systematic and comprehensive survey on the recursive and composite techniques of zero-knowledge proofs. Firstly, regarding recursive zero-knowledge proofs for algebraic statements, inner product arguments, a kind of Σ-protocols proving that the inner product of two committed vectors equals a public scalar, firstly rely on recursive techniques to achieve a breakthrough in communication complexity, reducing it from linear to logarithmic. We extensively survey inner product arguments, comparing and analyzing them in detail, particularly those based on the Pedersen commitment scheme. This analysis is conducted from the perspectives of prover complexity, communication complexity, verifier complexity, and more. Secondly, regarding recursive zero-knowledge proofs for non-algebraic statements, the key idea is to express the verifier's algorithm as an arithmetic circuit and generate a new proof for this new circuit, which in turn proves the validity of the original proof. This recursive technique is primarily used to aggregate proofs, construct incrementally verifiable computation schemes and combine proof systems. We thoroughly examine the research landscape of these predominant applications, specifically focusing on incrementally verifiable computation schemes and the circuit-based composition of proof systems. We conduct an in-depth comparative analysis of the complexity, key technologies, and implementation skills associated with incrementally verifiable computation schemes. Thirdly, regarding composite zero-knowledge proofs for composite statements, the key idea is to prove the algebraic part using Σ-protocols, prove the non-algebraic part using general-purpose zero-knowledge proofs, and optionally prove the consistency of variables that appear in both the algebraic and non-algebraic parts using some linking protocols. Among composite statements, a basic and common form is that "the openings of some algebraic commitments satisfy an arithmetic/Boolean circuit." Zero-knowledge proofs for this type of statement are also known as commit-and-prove zero-knowledge proofs (CP-ZKPs). We provide a granular comparative analysis of CP-ZKPs from various perspectives, including complexity, the setup phase, key modules, and more. Finally, we outline future research directions for recursive and composite techniques of zero-knowledge proofs, from the aspects of algebraic statements, non-algebraic statements, and composite statements.

Keywords　zero-knowledge proofs; recursive zero-knowledge proofs; inner product arguments; incrementally verifiable computation schemes; composite zero-knowledge proofs; commit-and-prove zero-knowledge proofs

1　引　言

随着数字经济持续高速增长，数据流通已成为数据价值化的重要途径. 在开发利用数据的同时，如何加强数据要素的安全防护是数据流通面临的关键问题. 零知识证明(Zero-Knowledge Proofs，ZKPs)是实现数据安全流通的关键技术之一. 它作为一种重要的密码学协议，允许证明者向验收者证明某个断言的正确性，同时又不泄露任何额外信息.

1985 年，Goldwasser 等人[1]首次提出零知识证明的概念. 随后 Goldreich 等人[2]证明任意的 NP 语言都存在相应的零知识证明协议. 零知识证明作为关键模块被广泛应用于其他众多密码学协议中，包括身份认证[3]、数字签名[4]、公钥加密[5-6]、安全多方计算[7]等. 并且近些年来零知识证明尤其是简洁非交互零知识证明协议的快速发展也使其广泛应用到众多实际场景中，例如匿名交易[8]、智能合约[9]、程序分析[10]、网络中间盒[11]、机器学习[12]、数据库查询[13]等.

零知识证明所针对的断言包含三种类型：代数断言、非代数断言及结合了代数和非代数断言的复合断言. 针对代数断言，如证明代数群中的离散对数问题等，通常使用 Σ 协议完成高效证明[14]. 针对非代数断言，如证明哈希原象等，通常把非代数断言表示成电路，然后调用针对电路的零知识证明协议完成高效证明[15]. 针对复合断言，通常使用 Σ 协议证明代数断言部分，使用针对电路的零知识证明协议证明非代数断言部分，并选择性地使用链接协议证明两部分断言中的相同变量满足一致性[16]. 经过几

十年的发展,零知识证明协议在性能、功能等方面已取得了巨大的进展,然而在实际应用中仍存在性能不足、可扩展性差、功能单一等若干问题.为了有效处理这些问题,递归技术[17-18]在近年来被逐渐应用到零知识证明领域中①.

递归指用一个证明来证明另一个证明[17-18].在证明代数断言的 Σ 协议里,递归技术可以有效地降低通信复杂度[17],并遵循某种范式使其降至对数级别[25].在证明非代数断言的零知识证明协议里,递归技术可以用来构造高可扩展性的增量可验证计算方案[18]并实现证明系统的组合[26].这些优良特性可以反馈到针对复合断言的复合零知识证明协议里以得到更卓越的性能及更强大的功能.

1.1 相关工作

目前,学术界已有许多关于零知识证明的研究综述.例如,在密码学理论和复杂度理论层面,Goldreich[27]系统梳理了零知识证明自提出后约二十年的发展情况;Li 等人[28]对零知识证明的相关定义、针对若干 NP 问题的应用及若干组合操作等做了综述,其中组合特指密码学理论层面的顺序组合、并行组合和并发组合.在协议构造层面,若干综述[15,29-31]介绍了零知识简洁非交互知识论证(zero-knowledge Succinct Non-interactive ARgument of Knowledge,zkSNARK)这一特殊零知识证明协议的模块化构造方法及分类等.特别的,Baum 等人[32]对基于向量不经意线性求值(Vector Oblivious Linear Evaluation)构造的零知识证明协议做了综述.同时,零知识证明国际标准化组织 ZKProof② 也一直在持续更新零知识证明社区参考文档③.然而,这些综述几乎不涉及零知识证明的递归与复合技术.Thaler[15]介绍了针对非代数断言的递归零知识证明的应用,即聚合证明、组合证明系统和构造增量可验证计算方案,但该综述只介绍了基于电路组合证明系统的思想,没有介绍相关研究工作,且没有对增量可验证计算方案的构造详细分类和详细对比性能等.此外,该综述没有分类介绍针对代数断言的递归零知识证明和针对复合断言的复合零知识证明.ZKProof 组织在 0.3 版本的参考文档里仅简略提到了针对非代数断言的零知识证明递归技术.

1.2 本文贡献及结构

本文从针对代数断言的递归零知识证明、针对非代数断言的递归零知识证明和针对复合断言的复合零知识证明三个方面,详细且全面地梳理了递归与复合技术的研究现状.在代数断言方面,重点对比分析了基于 Pedersen 承诺方案的内积论证协议;在非代数断言方面,系统分类与剖析了增量可验证计算方案;在复合断言方面,深入对比分析了承诺并证明的零知识证明协议.最后,在此基础上为后续研究者提供了有发展前景的研究方向.

本文的组织结构如图 1 所示.第 2 节介绍相关

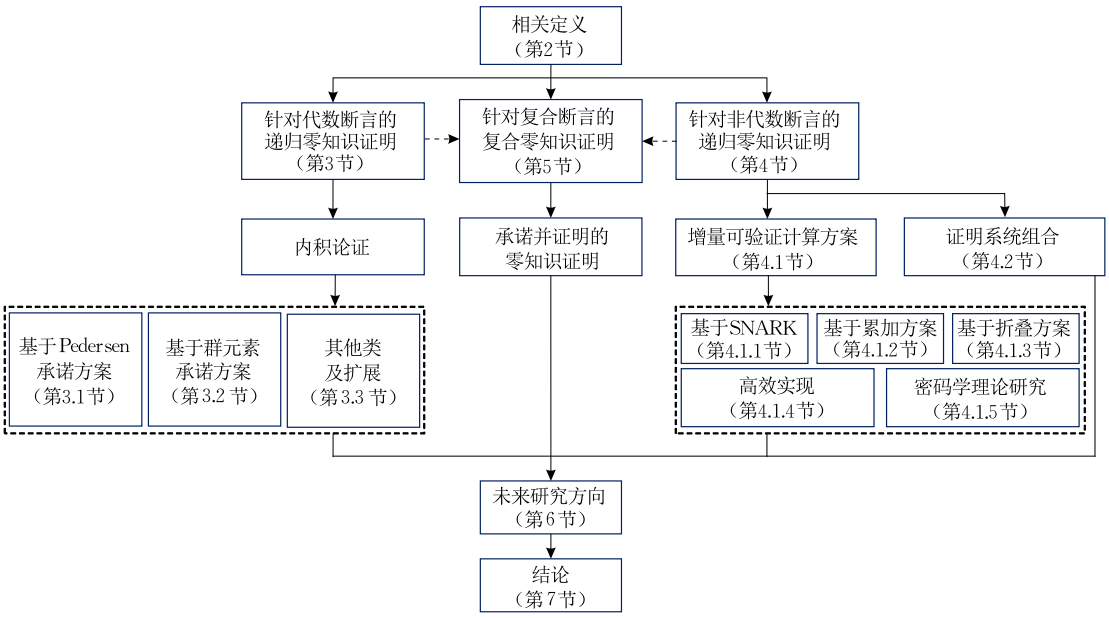


图 1 本文结构

① 在零知识证明领域,递归技术还被用于构造并发/重置零知识的模拟器[19-24],本文不涵盖这类递归技术.
② ZKProof. https://zkproof.org/
③ https://docs.zkproof.org/reference.pdf

定义；第 3 节聚焦针对代数断言的递归零知识证明，重点考虑首次利用递归技术高效证明内积关系这一特殊代数断言的内积论证协议的研究现状；第 4 节聚焦针对非代数断言的递归零知识证明，重点考虑增量可验证计算方案和基于电路的证明系统组合这两个主流应用的研究现状；第 5 节聚焦针对复合断言的复合零知识证明，重点考虑承诺并证明的零知识证明协议的研究现状；第 6 节讨论零知识证明递归与复合技术的未来研究方向；最后，在第 7 节总结全文.

2　相关定义

本文常用的缩略词及含义对照表如表 1 所示.

表 1　缩略词及其含义对照表

缩略词	含义
ZKP	Zero-Knowledge Proof 零知识证明
zkSNARK	zero-knowledge Succinct Non-interactive ARgument of Knowledge 零知识简洁非交互知识论证
NARK	Non-interactive ARgument of Knowledge 非交互知识论证
IPA	Inner Product Argument 内积论证
WIPA	Weighted Inner Product Argument 加权内积论证
GIPA	Generalized Inner Product Argument 广义内积论证
(N)IVC	(Non-uniform) Incrementally Verifiable Computation (非均匀)增量可验证计算
(N)PCD	(Non-uniform) Proof-Carrying Data (非均匀)携带证明数据
FFT	Fast Fourier Transform 快速傅里叶变换
QAP	Quadratic Arithmetic Program 二次算术程序
R1CS	Rank-1 Constraint System 一阶约束系统
CCS	Customizable Constraint System 可定制约束系统
NP	Non-deterministic Polynomial 非确定性多项式
FRI	Fast Reed-Solomon Interactive Oracle Proofs of Proximity 快速里德-所罗门编码接近性交互谕示证明
IOP	Interactive Oracle Proof 交互式谕示证明
LPCP	Linear Probabilistic Checkable Proof 线性概率可验证明
PIOP	Polynomial Interactive Oracle Proof 多项式交互式谕示证明
MPCitH	MPC-in-the-head 头脑中的安全多方计算
CP-ZKP	Commit-and-Prove Zero-Knowledge Proof 承诺并证明的零知识证明
CP-SNARK	Commit-and-Prove SNARK 承诺并证明的 SNARK

定义 1.　零知识证明. 零知识证明(Zero-Knowledge Proof，ZKP)是运行在证明者和验证者之间的一种密码协议，允许证明者向验证者证明某个断言的正确性，而又不泄露任何额外信息. 零知识证明一般满足以下三个安全属性：

(1) 完备性(Completeness). 如果断言是正确的，且证明者和验证者均诚实地执行协议，那么验证者最终会接受.

(2) 可靠性(Soundness). 如果断言是错误的，且验证者诚实地执行协议，那么他最终会拒绝.

(3) 零知识性(Zero-knowledge). 如果断言是正确的，且证明者诚实地执行协议，那么验证者只会知道断言的正确性，而不会获得任何额外信息.

在不同的场景下，这些属性还会有诸多变种.

零知识证明针对的断言一般由包含实例和见证的关系定义，根据关系中操作的类型，可以把断言划分成代数断言、非代数断言和结合了代数与非代数断言的复合断言.

定义 2.　代数断言. 代数断言(Algebraic Statement)由素数阶群、RSA 群等代数群上的关系定义，如离散对数或模根的知识证明等.

定义 3.　非代数断言. 非代数断言(Non-algebraic Statement)不涉及代数群，一般由算术/布尔电路的可满足性关系定义，如哈希函数原象或 AES 加密明文的知识证明等.

定义 4.　复合断言. 复合断言(Composite Statement)同时含代数与非代数断言，如 DSA/RSA 签名或比特币地址对应私钥的知识证明等.

定义 5.　Σ 协议[33]. Σ 协议是一类三步公开抛币的交互式零知识证明协议. 在第一步，证明者发送初始消息 a 给验证者；在第二步，验证者发送挑战 e 给证明者；在第三步，证明者发送对挑战的响应 z 至验证者. Σ 协议一般满足以下三个安全属性：

(1) 完备性(Completeness). 如果断言是正确的，且证明者和验证者均诚实地执行协议，那么验证者最终会接受.

(2) n-特殊可靠性(n-Special Soundness). 存在一个概率多项式时间的提取算法，其输入任意的实例 x 及任意 n 个针对 x 的可接受副本 $(a,e_1,z_1),\cdots,(a,e_n,z_n)$，其中 $e_1,\cdots,e_n$ 均不相同，输出一个针对 x 的见证 w.

(3) 特殊诚实验证者零知识性(Special Honest-Verifier Zero-knowledge). 存在一个概率多项式时

间的模拟器,其输入任意的实例 x 和挑战 e,输出一个副本(a,e,z),且该副本和诚实证明者与验证者真实执行协议生成的副本具有完全一致的分布.

定义 6. 零知识简洁非交互知识论证[15]. 零知识简洁非交互知识论证(zero-knowledge Succinct Non-interactive ARgument of Knowledge,zkSNARK)是一类特殊的零知识证明协议,额外满足简洁性,非交互及知识论证. 其中简洁性指协议的证明规模较小且验证高效;非交互指协议中证明者只向验证者发送一次消息,验证者收到后自行验证;知识论证指协议不仅具备可靠性,还表明若证明者能让诚实验证者信服,则他知道相应的见证.

定义 7. 递归零知识证明. 递归零知识证明(Recursive Zero-knowledge Proof)指用一个证明来证明另一个证明的技术. 对于一个证明 π,验证者不直接对其验证,而是由证明者生成另一个证明 π',其证明 π 满足相应的验证算法.

定义 8. 复合零知识证明. 复合零知识证明(Composite Zero-knowledge Proof)指高效证明复合断言的技术. 针对其中的代数断言部分,使用 Σ 协议证明;针对其中的非代数断言部分,使用针对电路的零知识证明协议证明;针对同时存在于两部分断言中的变量,选择性地使用特定的"链接协议"证明变量在 Σ 协议和针对电路的零知识证明协议中满足一致性.

现有实用的零知识证明协议一般都包含启动阶段,生成证明和验证时所需的公共参数. 根据所需的信任程度,启动阶段可分为可信、通用可更新和透明三种类型.

定义 9. 可信的启动阶段. 可信的启动阶段(Trusted Setup)指过程中会产生秘密的陷门,且不能向任何人尤其是恶意证明者公开,否则协议便不具备可靠性. 因此,该启动阶段具有较高的可信要求,一般需要由可信第三方执行.

定义 10. 通用可更新的启动阶段. 通用可更新的启动阶段(Universally Updatable Setup)指过程中也会产生秘密的陷门,但任何人都可以对生成的公共参数实施更新操作,只要有一个诚实方正确地实施了更新,那么最终的公共参数就是可信的,且生成的公共参数可适用于一定规模下的所有断言. 因此,该启动阶段的可信要求相对较低.

定义 11. 透明的启动阶段. 透明的启动阶段(Transparent Setup)指过程中不会产生任何秘密的陷门. 因此,该启动阶段具有最低的可信要求.

3 针对代数断言的递归零知识证明

针对代数断言,最常见及高效的证明方式是使用 Σ 协议[14]. 该类协议虽然具有较高的实际性能,但第三步发送的消息规模通常和断言规模成线性关系,导致其具有线性级别的通信复杂度. 为了突破这个界限并实现对数级别的通信复杂度,若干协议依赖于递归技术[17,34]. 递归指用一个证明来证明另一个证明. 在 Σ 协议中,证明者可以不在第三步发送响应,而是证明他知道这些响应且满足验证者需验证的关系. 这个验证关系也可看作一个代数断言且通常会和初始证明的代数断言具有一样的形式,只是规模有所缩减①. 接着可以不断重复该递归过程,直到响应的规模达到较小的常数级别,然后由证明者发送给验证者,验证者验证以完成证明.

上述递归技术在 Σ 协议中的应用首次出现于内积论证(Inner Product Argument,IPA). 内积论证是一类特殊的零知识证明协议,允许证明者向验证者证明两个被承诺向量的内积等于某个公开标量,其也可以看作是一种特殊的求和验证论证[36]. 内积论证是众多密码学协议的核心组成模块,如算术电路可满足性论证[17,34,38-41]、范围证明[34,38-40]、多项式承诺方案[41-45]等. 根据所用承诺方案的不同,内积论证有多种类型的实例化,包括基于 Pedersen 承诺方案的内积论证协议、基于群元素承诺方案的内积论证协议、基于格承诺方案等的其他类内积论证协议. 同时,内积论证所使用的递归技术又被研究者们应用到了证明一般代数断言的 Σ 协议中,带来了卓越的性能提升. 下面依据这些分类介绍内积论证协议的研究现状.

3.1 基于 Pedersen 承诺方案的内积论证协议

在这类内积论证协议中,断言中被承诺的向量均由域元素组成,承诺方案使用的是 Pedersen 承诺方案[46],协议的安全性主要基于离散对数假设,典型协议总结如表 2 所示.

若两个向量被分别承诺成了两个群元素,则记此类 IPA 为 $\mathrm{IPA}_{\mathrm{tvc}}$,其所证明的关系可以表示为

① 这种对"问题"做归约的技术有时也被称为"分割再折叠(split-and-fold)"技术[35-37],即把一个大"问题"分割成若干小"问题",再通过某种方式如随机线性组合把这些小"问题"折叠成一个小"问题",实现大"问题"到小"问题"的归约.

表 2　基于 Pedersen 承诺方案的内积论证协议总结

协议		类别	证明复杂度	通信复杂度	验证复杂度	启动阶段	底层假设
Groth09[47]		$\mathrm{IPA}_{\mathrm{tvc}}$ 加权 $\mathrm{IPA}_{\mathrm{tvc}}$	$O(N)E$ $O(N)M$	$O(N)\mathbb{F}_p$	$O(N)E$	透明	DL
BCC+16[17]		$\mathrm{IPA}_{\mathrm{tvc}}$	$O(N)E$ $O(N)M$	$O(\log N)\mathbb{G}$ $O(\log N)\mathbb{F}_p$	$O(N)E$ $O(\log N)M$	透明	DL
Bulletproofs[34]		$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E$ $O(N)M$	$O(\log N)\mathbb{G}$	$O(N)E$	透明	DL
DRZ20[39]		$\mathrm{IPA}_{\mathrm{tvc}}$	$O(N)E_1$ $O(N)M$	$O(\log N)\mathbb{G}_1$ $O(\log N)\mathbb{F}_p$	$O(\log N)E_1$ $O(\log N)M$ $O(\log N)P$	通用 可更新	A-DL
ZZL+21[48]		$\mathrm{IPA}_{\mathrm{tvc}}$	$O(N)E$ $O(N)M$	$O(\log N)\mathbb{G}$	$O(N)E$	透明	DL
Bulletproofs+[40]		加权 $\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E$ $O(N)M$	$O(\log N)\mathbb{G}$	$O(N)E$	透明	DL
KLS22[41]	协议一	$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N2^{\sqrt{\log N}})E_1$	$O(\sqrt{\log N})\mathbb{G}_t$	$O(N)E_1$	透明	DL DPair
	协议二	$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E_1$	$O(\log N)\mathbb{G}_t$	$O(\sqrt{N})E_2$	透明	DL
	协议三	$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E_u$	$O(\log N)\mathbb{G}_v$	$O(\sqrt{N}\log N)E_v$	透明	DL
ZZT+23[49]		$\mathrm{IPA}_{\mathrm{tvc}}$ $\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E_1$ $O(N)M$	$O(\log N)\mathbb{G}_1$	$O(\log N)E_1$ $O(\log N)P$	通用 可更新	A-DL
KLL+23[50]		$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N)E_1$	$O(\log N)\mathbb{G}_t$	$O(\sqrt{N})E_2$	透明	DL
LS23[51]		$\mathrm{IPA}_{\mathrm{ovc}}$	$O(N2^{\sqrt{\log N}})E_1$	$O(\sqrt{\log N})\mathbb{G}_t$	$O(N/2^{\sqrt{\log N}})E_1$	透明	DL DPair

注：① 令$\mathbb{G}$表示阶为素数 p 的椭圆曲线循环群，$\mathbb{F}_p$表示对应的标量域，($\mathbb{G}_1$，$\mathbb{G}_2$，$\mathbb{G}_t$)表示阶为素数 p 的双线性群，($\mathbb{G}_u$，$\mathbb{G}_v$)表示一对阶分别为素数 u，v 的椭圆曲线群，其中$\mathbb{G}_u$由二维射影空间$\mathbb{P}^2(\mathbb{F}_v)$上的 Weierstrass 方程定义. 在通信复杂度列，令其表示相应群或域上的元素.

② N 表示向量维数；E 表示群幂运算，下标指出相应的群；M 表示域乘预算；P 表示配对运算.

③ 在底层假设列，DL 表示离散对数假设(Discrete Logarithm Assumption)；A-DL 表示非对称的离散对数假设(Asymmetric Discrete Logarithm Assumption)；DPair 表示双重配对假设(Double Pairing Assumption).

$$\{(\boldsymbol{g},\boldsymbol{h}\in\mathbb{G}^N,A,B\in\mathbb{G},z\in\mathbb{F}_p;\boldsymbol{a},\boldsymbol{b}\in\mathbb{F}_p^N):$$
$$A=\boldsymbol{g}^{\boldsymbol{a}}\wedge B=\boldsymbol{h}^{\boldsymbol{b}}\wedge z=\langle\boldsymbol{a},\boldsymbol{b}\rangle\},$$

其中 $A=\boldsymbol{g}^{\boldsymbol{a}}=\prod_{i=1}^{N}g_i^{a_i}$，$B=\boldsymbol{h}^{\boldsymbol{b}}=\prod_{i=1}^{N}h_i^{b_i}$ 分别表示对向量 $\boldsymbol{a},\boldsymbol{b}$ 的 Pedersen 承诺，$\langle\boldsymbol{a},\boldsymbol{b}\rangle=\sum_{i=1}^{N}a_ib_i$表示向量内积.

若两个向量被一起承诺成了一个群元素，则记此类 IPA 为 $\mathrm{IPA}_{\mathrm{ovc}}$，其所证明的关系可以表示为

$$\{(\boldsymbol{g},\boldsymbol{h}\in\mathbb{G}^N,P\in\mathbb{G},z\in\mathbb{F}_p;\boldsymbol{a},\boldsymbol{b}\in\mathbb{F}_p^N):$$
$$P=\boldsymbol{g}^{\boldsymbol{a}}\boldsymbol{h}^{\boldsymbol{b}}\wedge z=\langle\boldsymbol{a},\boldsymbol{b}\rangle\}.$$

特别的，若向量内积中的每一项都有一个公开的权重值，则称此类内积论证为加权内积论证(Weighted Inner Product Argument，WIPA).

2009 年，Groth[47]首次针对基于 Pedersen 承诺方案的内积关系，设计了透明的零知识 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议和零知识加权 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议，但这两个协议的通信、证明和验证复杂度均为 $O(N)$，其中具体通信开销主要为 $O(N)$个域元素，具体证明和验证开销主要为 $O(N)$个群幂运算. 基于这些内积论证协议，Groth 设计了在离散对数假设下具有 $O(\sqrt{N})$通信复杂度的算术电路可满足性论证协议，随后研究者们沿着该工作的技术路线进一步实现了 $O(\sqrt[3]{N})$的通信复杂度[52]，且进一步减少了协议轮数[53]. 在不考虑零知识属性的情况下，Bootle 等人[17]基于递归技术设计了首个具有 $O(\log N)$通信复杂度的 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议，其具体通信复杂度为 $6\log_2 N$，证明和验证复杂度均为 $O(N)$. 基于此协议，他们设计了首个在离散对数假设下具有 $O(\log N)$通信复杂度的算术电路可满足性论证协议. 基于 Bootle 等人的工作，Bünz 等人[34]设计了具有 $2\log_2 N$ 通信复杂度的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议，并用此协议设计了首个具有 $O(\log N)$通信复杂度的范围证明协议.

随后，基于递归技术构造内积论证成为主流，研究者们在此基础上进一步降低协议的复杂度. Chung 等人[40]基于递归技术改进了 Groth[47]的零知识加权 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议，设计了具有 $O(\log N)$通信复杂度的零知识加权 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议. Zhang 等人[48]指出，由于 $\mathrm{IPA}_{\mathrm{tvc}}$ 和 $\mathrm{IPA}_{\mathrm{ovc}}$ 所证明关系的不同，其所适用的应用场景也有所不同. 由于 $\mathrm{IPA}_{\mathrm{ovc}}$ 中两个向量被承诺在了一起，在应用时必须确保向量 $\boldsymbol{g}$ 和 $\boldsymbol{h}$ 是随机独立的，而在有些场景下该要求无法得到满足. 相比之下，在应用 $\mathrm{IPA}_{\mathrm{tvc}}$ 类协议时无需满足这种要求. 鉴于此，Zhang 等人以 $\mathrm{IPA}_{\mathrm{tvc}}$ 为对象，利用验证者的随机挑战把 Bootle 等人[17]的 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议的通信复杂度由 $6\log_2 N$ 降至 $4\log_2 N$，且证明和验证复杂度保持不变，该技术同样在洗牌论证协议——Curdleproofs[54]中有所体现. 借助结构化的承诺密钥，Daza 等人[39]设计了首个同时具有 $O(\log N)$通信

复杂度和验证复杂度的 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议. 然而,该协议的启动阶段是通用可更新的而不再是透明的. Zhou 等人[49]借助验证者的随机挑战进一步降低了 Daza 等人的 $\mathrm{IPA}_{\mathrm{tvc}}$ 协议的具体通信和验证复杂度,并利用结构化的承诺密钥设计了首个同时具有 $O(\log N)$ 通信和验证复杂度的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议. Kim 等人[41]考虑基于离散对数假设且启动阶段公开透明的场景,设计了三个 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,其一具有 $O(\sqrt{\log N})$ 的通信复杂度和 $O(N)$ 的验证复杂度,其二具有 $O(\log N)$ 的通信复杂度和 $O(\sqrt{N})$ 的验证复杂度,其三具有 $O(\log N)$ 的通信复杂度和 $O(\sqrt{N}\log N)$ 的验证复杂度且无需配对群. 基于该项工作,Kim 等人[50]进一步扩充了其中的第二个 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,提供了更完备的安全性证明并做了编程实现. Lee 等人[51]结合其中的第一个和第二个 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,设计了同时具有 $O(\sqrt{\log N})$ 通信复杂度和 $O(N/2^{\sqrt{\log N}})$ 验证复杂度的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,即同时具有亚对数级别的通信复杂度和亚线性级别的验证复杂度.

此外,若干研究致力于使内积论证协议具有更多良好的特性. Wahby 等人[42]改进了 Bünz 等人[34]的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,使其可以证明被承诺向量和公开向量的内积关系. 基于此改进的协议,他们构造了高效的多线性多项式承诺方案. Hoffmann 等人[38]解释了如何对 Bünz 等人[34]的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议实施随机化,使其具备零知识属性,同时保持对数级别的通信复杂度. 基于改进的 $\mathrm{IPA}_{\mathrm{ovc}}$ 协议,他们构造了针对二次关系可满足性问题的证明.

3.2 基于群元素承诺方案的内积论证协议

在这类内积论证协议中,断言中被承诺的向量至少有一个是由群元素组成的向量,一般使用 AFGHO 承诺方案[55]或其变种[56]对群向量做承诺. 若另一个被承诺的向量由域元素组成,则仍用 Pedersen 承诺方案对域向量做承诺. 若一个向量为群向量,另一个向量为域向量,则此类内积论证又称多幂论证(Multi-exponentiation Argument),其所证明的关系可以表示为

$$\{(\boldsymbol{G}\in\mathbb{G}_2^N,\boldsymbol{h}\in\mathbb{G}_1^N,T\in\mathbb{G}_t,B,Z\in\mathbb{G}_1;\boldsymbol{A}\in\mathbb{G}_1^N,\boldsymbol{b}\in\mathbb{F}_p^N):\ T=\boldsymbol{E}(\boldsymbol{A},\boldsymbol{G})\wedge B=\boldsymbol{h}^{\boldsymbol{b}}\wedge Z=\boldsymbol{A}^{\boldsymbol{b}}\},$$

其中 $T=\boldsymbol{E}(\boldsymbol{A},\boldsymbol{G})=\prod_{i=1}^{N}e(A_i,G_i)$ 表示对群向量 $\boldsymbol{A}$ 的 AFGHO 承诺,$B=\boldsymbol{h}^{\boldsymbol{b}}=\prod_{i=1}^{N}h_i^{b_i}$ 表示对域向量 $\boldsymbol{b}$ 的 Pedersen 承诺,$\boldsymbol{A}^{\boldsymbol{b}}=\prod_{i=1}^{N}A_i^{b_i}$ 可看作 $\boldsymbol{A}$ 与 $\boldsymbol{b}$ 的内积.

若两个向量均为群向量,则此类内积论证又称内配对积论证(Inner Pairing Product Argument),其所证明的关系可以表示为

$$\{(\boldsymbol{G}\in\mathbb{G}_2^N,\boldsymbol{H}\in\mathbb{G}_1^N,T,U,Z\in\mathbb{G}_t;\boldsymbol{A}\in\mathbb{G}_1^N,\boldsymbol{B}\in\mathbb{G}_2^N):\ T=\boldsymbol{E}(\boldsymbol{A},\boldsymbol{G})\wedge U=\boldsymbol{E}(\boldsymbol{H},\boldsymbol{B})\wedge Z=\boldsymbol{E}(\boldsymbol{A},\boldsymbol{B})\},$$

其中 $T=\boldsymbol{E}(\boldsymbol{A},\boldsymbol{G})=\prod_{i=1}^{N}e(A_i,G_i)$,$U=\boldsymbol{E}(\boldsymbol{H},\boldsymbol{B})=\prod_{i=1}^{N}e(H_i,B_i)$ 分别表示对群向量 $\boldsymbol{A},\boldsymbol{B}$ 的 AFGHO 承诺,$\boldsymbol{E}(\boldsymbol{A},\boldsymbol{B})$ 可看作 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 的内积.

2019 年,Lai 等人[57]首次引入了针对配对语言的内积论证,即内配对积论证. 针对不同的内积关系,他们设计了两个透明的内配对积论证协议,都具有对数通信复杂度、线性证明和验证复杂度. 基于这两个协议,他们构造了针对一般双线性群算术关系的高效论证协议. Bünz 等人[44]借助结构化的承诺密钥改进了上述协议,使其具备了对数级别的验证复杂度. 然而,协议的启动阶段变成通用可更新而不再透明. 同时,他们还定义了广义内积论证(Generalized Inner Product Argument,GIPA)并基于递归技术给出了具有对数通信复杂度的一般构造,统一了 Lai 等人的内配对积论证及基于 Pedersen 承诺方案的内积论证,简化了内积论证的构造及安全性证明. 此外,广义内积论证还蕴涵了多幂论证的定义,借助结构化的承诺密钥,可实例化具有对数通信和验证复杂度的多幂论证协议. 2021 年,Lee[45]利用 AFGHO 承诺方案中消息空间和承诺密钥空间的对称性,设计了首个透明且具有对数通信和验证复杂度的内配对积论证协议. 2022 年,Gailly 等人[56]改进了 Lai 等人使用的群元素承诺方案,使其适用于结构化的承诺密钥,由此介绍了两个新的针对群元素的双重同态承诺方案,并套用广义内积论证的框架设计了一个新的具有对数通信和验证复杂度的内积论证协议,该协议证明的关系同时结合了多幂论证关系和内配对积论证关系.

在应用方面,该类内积论证协议可用来构造多变量多项式承诺方案、聚合 SNARK 证明,甚至可以用来降低基于 Pedersen 承诺方案的内积论证协议的通信复杂度. Bünz 等人[44]利用设计的内积论证协议构造了高效的双变量多项式承诺方案,并实例化了首个具有 $O(\sqrt{d})$ 证明复杂度、$O(\log d)$ 通信和验证复杂度及 $O(\sqrt{d})$ 规模公共参数的单变量多项式承诺方案,其中 d 指多项式的阶. 此外,他们还利用内积论证协议聚合 Groth16 协议[58]的证明. 对于 n 个 Groth16 证明,聚合后的证明规模仅为 $O(\log n)$. Lee[45]利用设计的内配对积论证协议构造了首个具有对数通信和验证复杂度的透明多变量多项式承诺方案——Dory,其被用于设计具有对数通信和验证复杂度的透明 zkSNARK——Xiphos[59]. Gailly 等人[56]利用设计的内积论证构造了针对 Groth16 协

议的聚合方案——SnarkPack. 对于 n 个 Groth16 证明，聚合后的证明规模和验证复杂度均为 $O(\log n)$，相比于 Bünz 等人[44]的聚合方案，该方案要求 n 个 Groth16 证明使用相同的结构化承诺密钥，这样在聚合时可以重复使用这些密钥，而无需额外的可信启动步骤. Ambrona 等人[60]利用广义内积论证对 Plonk 协议[61]的证明实施聚合. 对于 n 个 Plonk 证明，聚合后的证明规模和验证复杂度均为 $O(\log n)$. Kim 等人[41]利用聚合的多幂论证协议，设计了首个具有亚对数通信复杂度的基于 Pedersen 承诺方案的内积论证协议.

3.3　其他类内积论证协议及扩展

内积论证协议还可用其他的承诺方案做实例化. 2022 年，Kuchta 等人[62]首次设计了基于格的内积论证协议，其中被承诺的向量均由环上的元素组成，使用基于格的承诺方案[63]对向量做承诺. 基于 Bootle 等人[17]的递归技术，他们的内积论证协议也实现了对数通信复杂度. 同时，内积论证协议还可基于里德-所罗门码构造[64-66]. 在这类协议中，一般先用里德-所罗门码对向量做编码，然后使用默克尔哈希树对编码后的结果做承诺. 此类内积论证协议可被用于构造抗量子的可验证多项式委托方案及算术电路可满足性论证等.

此外，内积论证的递归构造技术还启发了其他密码学方案的高效构造. 2019 年，Bowe 等人[43]发现 Bünz 等人[34]的内积论证协议的验证算法是高度结构化的，当验证 n 个证明时，可以让第三方对证明实施聚合，使得验证者只需执行 n 次对数级别的操作和一次线性级别的操作，而非平凡地执行 n 次线性级别的操作. 他们把这个技术应用到了递归零知识证明中，使得编码验证算法的验证电路只需执行对数级别的操作，进而设计了首个透明的高效增量可验证计算方案——Halo. 2020 年，Bünz 等人[67]利用递归技术构造了具有对数通信复杂度的基于未知阶群的多项式求值协议，并基于此进一步构造了首个实际证明高效且具有对数通信和验证复杂度的透明 zkSNARK——Supersonic. 2023 年，Das 等人[68]基于内积论证提出了新的门限签名方案，验证密钥仅含 7 个群元素，签名仅含 8 个群元素，验证签名仅需 8 个群幂运算和 8 个配对运算，方案支持任意的门限且签名者可有任意的权重.

同时，内积论证协议所证明的内积关系可视为一种特殊的代数关系，进而与证明代数关系的 Σ 协议存在若干潜在关联. 2020 年，Attema 等人[25]提出了压缩的 Σ 协议理论，协调了内积论证协议与 Σ 协议，使得应用内积论证时可以遵循已有的安全协议理论，同时实现相同的对数通信复杂度. 该理论继承了 Σ 协议的灵活性和通用性，同时将其通信复杂度从线性级别压缩至对数级别. 随后，压缩的 Σ 协议理论被应用及扩展至众多场景，包括部分知识证明[69]、双线性群算术电路证明[70]、基于格的证明[71]、环上的算术电路证明[72]等.

4　针对非代数断言的递归零知识证明

非代数断言通常都表示成算术电路或布尔电路，然后调用针对电路的零知识证明协议完成证明. 经过多年的发展，针对电路的零知识证明协议相继被提出，这些协议在关键技术、性能、安全性等诸多方面大相径庭，其中一类引起学术界和工业界极大兴趣的协议被统称为(zk)SNARK. SNARK 已有诸多构造方法，而且在性能、功能、安全性等层面不尽相同[15]. 总体而言，传统 SNARK 难以兼顾证明复杂度、通信复杂度和验证复杂度，随着其逐渐应用至更复杂工程项目中，其高空间复杂度和弱扩展性等弊端也逐渐凸显. 为了处理这些问题，递归零知识证明应运而生.

递归指用一个证明来证明另一个证明. 对于一个 SNARK 证明 π，可以不让验证者直接验证，而是把验证算法表示成验证电路，然后调用针对电路的 SNARK 生成证明 π'，其证明 π 满足验证电路. 若 π' 通过验证，则 π 也是有效证明. 这种递归的基本思想进一步拓宽了 SNARK 的应用领域，并可解决传统 SNARK 在工程应用中的诸多问题. 一个直观的应用场景即是对多个证明实施聚合. 当需要验证多个证明时，可以把多个验证算法表示成一个电路，接着由证明者证明该电路是可满足的，这样验证者只需验证一个证明，极大地提高了效率. 这种聚合的思想主要应用到了诸多 zk-Rollup 项目中，如 Aztec①、Scroll② 等. 在学术研究方面，目前递归零知识证明主要用于构造增量可验证计算方案及组合证明系统. 其中，增量可验证计算方案可以降低 SNARK 的内存占用、提高其扩展性并实现增量验证特性；通过组合证明系统，可以结合不同 SNARK 的性能优势③. 针对非代数断言的递归零知识证明，第 4.1 节介绍增量可验证计算方案的研究现状，第 4.2 节介绍基于电路的证明系统组合的研究现状.

4.1　增量可验证计算方案

增量可验证计算(Incrementally Verifiable Computation，IVC)的概念于 2008 年被 Valiant[76]首次

① Aztec. https://aztec.network/
② Scroll. https://scroll.io/
③ 在信息论证明层面，递归的思想还被用于组合概率可验证明(Probabilistic Checkable Proof，PCP)[73]和交互式谕示证明(Interactive Oracle Proof，IOP)[74-75].

提出,其允许证明者以增量的方式证明一个无限次有序计算执行的正确性.IVC 考虑一个定义在路径图上的无限次有序计算,每步计算的输出都附带一个证明,证明了该步及之前所有计算的正确性.证明者可以在若干步计算的基础上执行一步新的计算并产生输出及更新后的证明,此证明同样保证新的计算及之前所有计算的正确性.在性能方面,IVC 要求证明者完成每步计算并证明的开销、证明规模和验证开销均独立于计算已被执行的次数.随后 2010 年,Chiesa 等人[77]把 IVC 的概念拓展至携带证明数据(Proof-Carrying Data,PCD).PCD 可看作 IVC 的分布式变种,也可称为分布式的增量可验证计算,其证明的计算定义在一个有向无环图中,图中的节点可认为是互不信任的实体,图中的边承载计算的结果及证明,证明了该步及之前所有计算的正确性.同 IVC,PCD 中的节点更新计算及证明的开销独立于计算在图中已被执行的次数,同时证明规模和验证开销也和图的规模无关.然而,Chiesa 等人的 PCD 方案基于辅助证明者的传闻论证(Assisted-Prover Hearsay-Argument),该论证又基于签名方案和复杂度较高的通用论证[78],使得其 PCD 的性能较差,难以实际应用.

2013 年,Bitansky 等人[18]首次提出可以通过递归组合 SNARK 来构造 IVC/PCD,即把表达计算的电路和验证证明的电路组合到一起,要求每步生成的证明不仅证明了当前步骤计算的正确性,还证明了上一步输出证明的正确性,这样只需验证一次证明即可确保当前及之前所有步骤计算的正确性.上述递归的设计思想启发了后续的诸多工作并应用到了众多实际场景中,如确保语言语义[79]、可验证 MapReduce 计算[80]、可验证图像编辑[81]、可验证注册[82]、区块链[83-85]等.增量可验证计算方案的构造根据底层的关键技术可以划分为:基于 SNARK、基于累加方案和基于折叠方案,相关总结如表 3 所示.

表 3 增量可验证计算方案总结

方案	时间	类别	递归开销	单步 证明开销	证明规模	验证开销	底层假设	关键技术
BCC+13[18]	2013	PCD	×	×	×	×	×	SNARK
BCT+14[86]	2014	PCD	$3rP$	$O(\|C\|)$-FFT $O(\|C\|)$-MSM	$2\mathbb{G}_1$ $1\mathbb{G}_2$	$3P$	GGM	SNARK
Halo[43]	2019	IVC	$O(\log\|C\|)$ GSM	$O(\|C\|)$-FFT $O(\|C\|)$ GSM	$O(\log\|C\|)\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	原子累加方案
COS20[87]	2020	PCD	$O(r\log^2\|C\|)M$ $O(r\log^2\|C\|)H$	$O(\|C\|)$-FFT $O(\|C\|)$-MHT	$O(\log^2\|C\|)\mathbb{F}$	$O(\log^2\|C\|)M$ $O(\log^2\|C\|)H$	RO	SNARK
BCM+20[88]	2020	PCD	×	×	×	×	×	原子累加方案
BCL+21[89]	2021	PCD	$2r$-MSM $5r$ GSM	$O(\|C\|)$-MSM	$O(\|C\|)\mathbb{F}$ $15\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	分割累加方案
Nova[90]	2022	IVC	2 GSM	$O(\|C\|)$-MSM	$O(\log\|C\|)\mathbb{F}$ $O(\log\|C\|)\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	折叠方案
SuperNova[91]	2022	NIVC	2 GSM	$O(\|C\|)$-MSM	$O(\ell\|C\|)\mathbb{F}$ $O(\ell)\mathbb{G}$	$O(\|C\|)$-MSM $O(\ell\|C\|)M$	DL RO	折叠方案
HyperNova[92]	2023	IVC	1 GSM $O(\log\|C\|)M$ $O(\log\|C\|)H$	$O(\|C\|)$-MSM	$O(\log\|C\|)\mathbb{F}$ $O(\log\|C\|)\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	多重折叠方案
ProtoStar[93]	2023	NIVC	3 GSM	$O(\|C\|)$-MSM	$O(\ell\|C\|)\mathbb{F}$ $O(\ell)\mathbb{G}$	$O(\|C\|)$-MSM $O(\ell\|C\|)M$	DL RO	分割累加方案
ZZD23[94]	2023	PCD	$(2r-1)$-MSM $O(r\log\|C\|)M$ $O(\log\|C\|)H$	$O(\|C\|)$-MSM	$O(\log\|C\|)\mathbb{F}$ $O(\log\|C\|)\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	多重折叠方案
KiloNova[95]	2023	NPCD	$O(r)$-MSM $O(r\log\|C\|)M$ $O(\log\|C\|)H$	$O(\|C\|)$-MSM	$O(\|C\|)\mathbb{F}$ $O(1)\mathbb{G}$	$O(\|C\|)$-MSM	DL RO	一般折叠方案

注:① IVC 指增量可验证计算(Incrementally Verifiable Computation),NIVC 指非均匀的增量可验证计算(Non-uniform Incrementally Verifiable Computation),PCD 指携带证明数据(Proof-Carrying Data),NPCD 指非均匀的携带证明数据(Non-uniform Proof-Carrying Data).

② 令$\mathbb{G}$表示素数阶的椭圆曲线循环群,($\mathbb{G}_1$,$\mathbb{G}_2$,$\mathbb{G}_t$)表示素数阶的双线性群,$\mathbb{F}$表示有限域,在证明规模列,用这些符号表示相应群或域上的元素.r 表示 PCD 中某步节点的入边数量.$|C|$表示增广电路的规模.ℓ 表示 NIVC 支持的函数集合的大小.P 表示配对运算,M 表示域乘运算,H 表示哈希运算.GSM 表示群标量乘运算,如 aG,其中 $G\in\mathbb{G}$,$a\in\mathbb{F}$.$O(|C|)$-MSM 表示规模为 $O(|C|)$的多标量乘运算,如 $a_1G_1+\cdots+a_{|C|}G_{|C|}$.$O(|C|)$-FFT 表示规模为 $O(|C|)$的快速傅里叶变换,$O(|C|)$-MHT 表示构造对 $O(|C|)$规模向量的默克尔树.

③ 在底层假设列,GGM 表示一般群模型(Generic Group Model),DL 表示离散对数假设(Discrete Logarithm Assumption),RO 表示随机谕言模型(Random Oracle).

④ BCC+13 只阐述了原理和安全性证明,没有给出具体构造.BCT+14 的构造使用了 Ben-Sasson 等人[96]的预处理 SNARK,该 SNARK 基于配对且针对每个电路都需要可信建立,此类 SNARK 中性能最优的是 Groth16 协议[58],表中列的性能数据对应的是 BCT+14 结合 Groth16 所产生的 PCD 方案.BCM+20 只提供了 PCD 的构造框架,没有结合具体的 SNARK 做实例化.KiloNova 的递归开销理应包含 $O(|C|)$的域运算,但在构造中假设该运算由额外的 IVC 保证了正确性.

⑤ Nova、HyperNova、ZZD23 的证明规模和验证开销是使用 SNARK 压缩后的结果.

除了高效构造，若干工作还深入研究了增量可验证计算方案的高效实现技术及相关密码学理论. 以下从方案构造、编程实现和密码学理论研究三方面介绍增量可验证计算方案的研究现状.

4.1.1　基于 SNARK 的增量可验证计算方案

在该类构造中，针对某个特定的 SNARK，一般先用电路表达出其验证算法，记为验证电路，然后和某步的函数电路组成一个新的电路，记为增广电路，其中函数电路实施计算，验证电路验证上一步的证明. 增广电路中除函数电路外的操作是为了实现增量可验证而需的额外操作，也称递归开销. 证明者在某步生成对增广电路的证明，验证者通过验证该证明即可确保该步及之前所有计算的正确性. 以 IVC 为例，该类构造的示意图如图 2 所示.

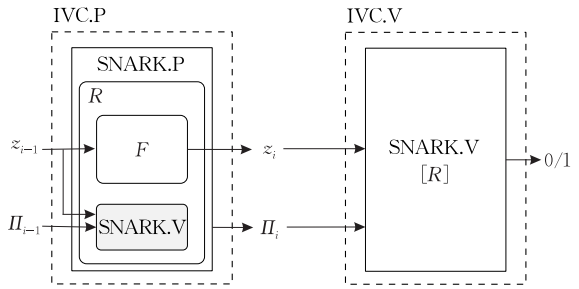


图 2　基于 SNARK 的增量可验证计算方案构造示意图

在第 i 步，证明者输入第 $i-1$ 步的输出 z_{i-1}，Π_{i-1}，其中 Π_{i-1} 证明了前 $i-1$ 步计算的正确性. 证明者计算 $z_i=F(z_{i-1})$ 并用 SNARK 生成对增广电路 R 的证明 Π_i，最终输出 z_i，Π_i. 可知若 Π_i 通过验证，则前 i 步的计算都是被正确执行的. 在性能方面，方案要求 SNARK 的验证复杂度至少亚线性于电路规模，否则增广电路的规模会以指数于计算次数的级别扩张. 在满足该要求的情况下，可知证明者在第 i 步的证明开销和计算已被执行的次数无关，且证明规模和验证开销也独立于计算次数.

2013 年，Bitansky 等人[18]首次提出这种构造思想，但他们主要研究原理和安全性，没有提供实际有效的具体构造. 2014 年，Ben-Sasson 等人[86]利用 Ben-Sasson 等人[96]的预处理 SNARK 构造了 PCD 方案并给出了 PCD 的首个编程实现. 该 SNARK 基于二次算术程序(Quadratic Arithmetic Program，QAP)，其验证算法主要含常数个配对操作，虽然满足性能要求，但在表达电路时仍需要大量的算术门. 例如，该类 SNARK 中验证复杂度最低的 Groth16 协议[58]，表达其验证算法需要约 4 万个门. 此外，在编程实现时为了避免域模拟，Ben-Sasson 等人[86]利用了 MNT4/MNT6 这个椭圆曲线循环，其性能比配对友好但不构成椭圆曲线循环的椭圆曲线要差很多，比标准的椭圆曲线则差更多. 2020 年，Chiesa 等人[87]基于全息交互式谕示证明和里德-所罗门码构造了透明且抗量子的预处理 SNARK——Fractal，并按照 Bitansky 等人的递归组合方法构造了抗量子的 PCD 方案. Fractal 具有多项式对数级别的验证复杂度，且仅基于有限域而不依赖任何椭圆曲线群，使得实现 PCD 时不需要域模拟，但验证电路需执行的域操作和哈希操作仍比较繁重，其验证电路规模约为 Groth16 验证电路规模的 40 倍.

4.1.2　基于累加方案的增量可验证计算方案

基于 SNARK 的增量可验证计算方案，由于要求 SNARK 具有亚线性的验证算法，使得许多不满足该性能条件但仍具有其他各类优势的 SNARK 难以用来构造 IVC/PCD，这反过来又限制了 IVC/PCD 的性能和安全属性. 2019 年，Bowe 等人[43]首次利用具有线性验证复杂度的 SNARK 构造了高效的 IVC 方案——Halo，并给出编程实现，其 SNARK 主要结合了 Sonic[97]的算术化和基于 Bulletproofs[34]的多项式承诺方案. 该 SNARK 的验证复杂度虽然是线性级别的，但在构造 IVC 时，其中线性级别的操作可以“推迟”到下一步并和下一步验证算法中线性级别的操作合并成一个，而在本步只需执行对数级别的验证操作，故验证电路也仅需表达对数级别的操作，使得增广电路的规模不会随着计算次数的增加而增加，进而满足 IVC 基本的性能要求. 相较于 Ben-Sasson 等人[86]的增量可验证计算方案，Halo 使用的 SNARK 满足透明启动且仅基于标准的椭圆曲线群. 在实现时为了避免验证电路的域模拟，Halo 利用了标准的椭圆曲线循环 Tweedledum/Tweedledee，其性能相较于配对友好的 MNT4/MNT6 椭圆曲线循环有极大的提升. 相较于 Chiesa 等人[87]的增量可验证计算方案，Halo 的证明规模减少了一百多 KB 且递归开销降低了一个数量级.

Halo 的构造思想于 2020 年被 Bünz 等人[88]形式化地定义为基于原子累加方案(Atomic Accumulation Scheme)的增量可验证计算方案. 原子累加方案的示意图如图 3 所示. 更通用的定义中证明和验证算法均会输入多个累加器和谓词实例，图 3 仅描绘输入为单个累加器和谓词实例的特殊情况，适用于 IVC 场景.

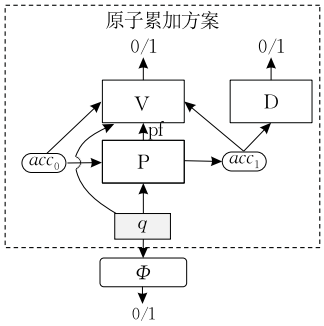


图 3 原子累加方案示意图

原子累加方案针对一个累加谓词 Φ,其输入实例 q,输出 1 或 0 表示 q 满足或不满足 Φ. 在具体实例化中,Φ 可以表示 SNARK 的验证算法或多项式承诺方案的求值验证算法,直接验证 Φ 可能需要线性级别的复杂度. 该方案由证明算法 P、验证算法 V 和决定算法 D 组成,其中证明算法 P 输入一个旧的累加器 acc_0 和谓词实例 q,输出一个新的累加器 acc_1 及证明 pf,验证算法 V 输入 acc_0,q,acc_1,pf,输出 0 或 1,决定算法 D 输入 acc_1,输出 0 或 1. 方案的安全性要求若 V 和 D 都输出 1,则 q 满足谓词 Φ 且 acc_0 满足决定算法 D. 可以发现,该方案把对 q 的验证归约为 V 的验证和 D 的验证,其中 V 一般具有亚线性如对数级别的复杂度,而 D 同 Φ 具有线性级别的复杂度. 原子累加方案的一个直接应用是摊销验证计算,如图 4 所示. 当要验证多个谓词实例 $q_1,\cdots,q_t$ 时,假设单个验证的复杂度是 $O(n)$,则直接验证 t 个的复杂度为 $O(tn)$,而借助原子累加方案,只需执行 t 次 V 算法和一次 D 算法,总复杂度仅为 $O(t\log n+n)$.

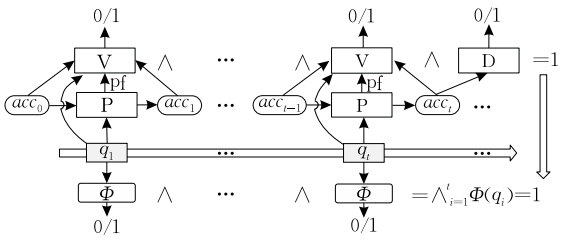
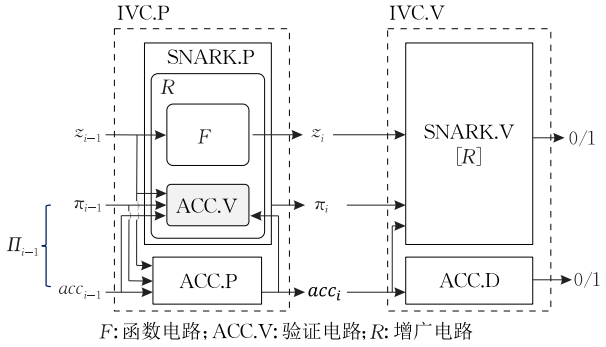


图 4 原子累加方案的摊销应用示意图

Bünz 等人[88]证明,任意具有高效原子累加方案的 SNARK 都可以编译成一个 IVC/PCD 方案,即使 SNARK 具有线性级别的验证复杂度. 若 SNARK 和其累加方案具有零知识性,那么 IVC/PCD 也具有零知识性. 若 SNARK 和其累加方案满足抗量子安全,那么 IVC/PCD 也满足抗量子安全. 以 IVC 为例,基于原子累加方案构造增量可验证计算方案的示意图如图 5 所示.



F: 函数电路; ACC.V: 验证电路; R: 增广电路

图 5 基于原子累加方案的增量可验证计算方案构造示意图

在第 i 步,证明者输入第 $i-1$ 步的输出 z_{i-1},Π_{i-1},其中 Π_{i-1} 由 SNARK 的证明 π_{i-1} 和累加器 acc_{i-1} 组成,对 π_{i-1} 的验证构成原子累加方案所针对的谓词. 证明者首先调用原子累加方案的证明算法 ACC.P 累加谓词实例 $(z_{i-1},acc_{i-1},\pi_{i-1})$ 和累加器 acc_{i-1},输出新的累加器 acc_i 和累加证明;然后构造增广电路 R,包含函数电路 F 和表达累加方案验证算法 ACC.V 的电路,计算 $z_i=F(z_{i-1})$ 并结合 ACC.P 的输出得到 R 的可满足赋值;接着调用 SNARK 的证明算法生成对 R 的证明 π_i;最后输出 IVC 的证明 $\Pi_i=(\pi_i,acc_i)$. 验证者只需调用 SNARK 的验证算法验证 π_i 并调用原子累加方案的决定算法验证 acc_i,即可确保前 i 步的计算都是正确执行的. 然而,Bünz 等人[88]主要做的是原理和安全性方面的研究,构造的 IVC/PCD 框架中没有结合具体的 SNARK 协议做实例化. 即使结合 Halo 中的 SNARK 协议做实例化,验证电路所需表达的原子累加方案的验证算法 ACC.V 仍需执行对数级别的群操作,带来比较高的递归开销.

2021 年,Bünz 等人[89]进一步改进了原子累加方案,提出了基于分割累加方案(Split Accumulation Scheme)的增量可验证计算方案,大幅降低了单步证明开销和递归开销. 分割累加方案的示意图如图 6 所示. 更通用的定义中证明和验证算法均会输

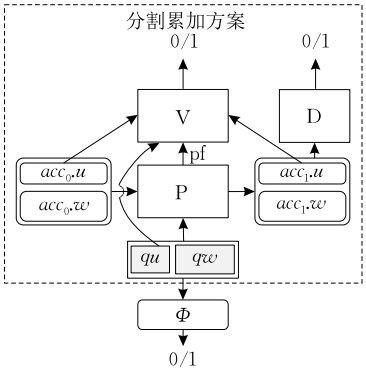


图 6 分割累加方案示意图

入多个累加器和谓词实例，图 6 仅描绘输入为单个累加器和谓词实例的特殊情况，适用于 IVC 场景.

分割累加方案的核心思想是把谓词实例 q 拆成规模较小的 qu 和规模较大的 qw 两部分，同时把累加器 acc 拆分成规模较小的 $acc.u$ 和规模较大的 $acc.w$ 两部分. 证明算法 P 同样需要输入完整的 q 和旧的累加器 acc_0，输出新的累加器 acc_1 及证明 pf，决定算法 D 需要输入完整的 acc_1，输出 0 或 1，但验证算法 V 仅需输入规模较小的 $acc_0.u$，qu，$acc_1.u$，pf，输出 0 或 1. 这样验证算法的复杂度得以大幅降低，使得相应 IVC/PCD 的递归开销得以大幅降低.

Bünz 等人[89]证明，任意具有高效分割累加方案的非交互知识论证（Non-interactive ARgument of Knowledge，NARK）都可以编译成一个 IVC/PCD 方案. 该 NARK 不仅可以具有线性级别的验证复杂度，也可以具有线性级别的通信复杂度，即非简洁的，这使得构造 IVC/PCD 时有更多具有优良特性的零知识证明协议可供选择. 若 NARK 和其累加方案具有零知识性，那么 IVC/PCD 也具有零知识性. 若 NARK 和其累加方案抗量子安全，那么 IVC/PCD 也抗量子安全. 以 IVC 为例，基于分割累加方案构造增量可验证计算方案的示意图如图 7 所示.

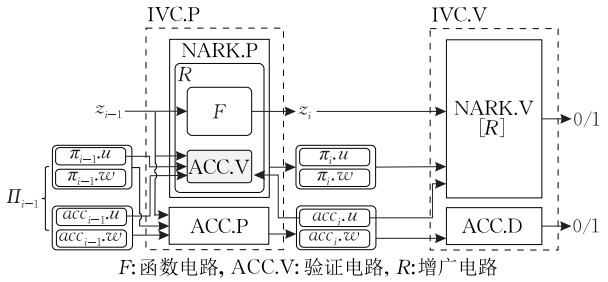


图 7　基于分割累加方案的增量可验证计算方案构造示意图

在第 i 步，证明者首先调用分割累加方案的证明算法 ACC.P 对输入的谓词实例和累加器做累加，生成新的累加器 acc_i 及证明；然后构造增广电路 R，包含函数电路 F 和表达累加方案验证算法 ACC.V 的电路，计算 $z_i=F(z_{i-1})$ 并结合 ACC.P 的输出得到 R 的可满足赋值；接着调用 NARK 的证明算法生成对 R 的证明 π_i；最后输出 IVC 的证明 $\Pi_i=(\pi_i,acc_i)$. 特别的，由于 NARK 证明的高效性且 ACC.V 仅需输入谓词实例和累加器中规模较小的部分，所构造的 IVC 具有较低的单步证明开销和递归开销. Bünz 等人[89]首先设计了针对一阶约束系统（Rank-1 Constraint System，R1CS）的 NARK 协议，然后构造了针对该协议的分割累加方案，最后构造了高效的 PCD 方案，其递归开销完全独立于电路规模.

同期，Boneh 等人[98]也研究了构造 PCD 的通用框架，提出了利用含公开或私有聚合方案的加性多项式承诺方案来构造 PCD 的思想，当多项式承诺方案具有公开聚合方案时，该思想同基于原子累加方案的构造思想；当多项式承诺方案具有私有聚合方案时，该思想同基于分割累加方案的构造思想.

2023 年，Bünz 等人[93]沿用基于分割累加方案的构造思想构造了支持 Plonkish 算术化的非均匀增量可验证计算方案——ProtoStar. Plonkish 算术化使其证明的电路可高效表达高阶约束及查询表约束，非均匀意味着 IVC 中各步执行的函数可以是不同的. 然而，ProtoStar 使用的分割累加方案只能高效地累加一个谓词实例和一个累加器，无法用来构造 PCD 方案. Eagen 等人[99]利用拉格朗日基的性质降低了 ProtoStar 累加多个谓词实例和多个累加器的开销，但并没给出进一步的 PCD 构造. 2024 年，Bünz 等人[100]构造了分别针对内存证明和 GKR 协议[101]的累加方案，基于这些方案构造的 IVC 方案可高效证明具有大内存的机器计算的正确性. 同年，Bünz 等人[102]构造了首个不依赖同态向量承诺方案的累加方案，其仅支持有限次数的累加但仍足够用于构造 IVC/PCD 方案.

4.1.3　基于折叠方案的增量可验证计算方案

累加方案针对的谓词可定义为验证任意的关系，如 NARK 的验证算法、多项式承诺方案的求值验证算法等，这使得其更具一般性且功能更丰富，但概念略显复杂. 2022 年，Kothapalli 等人[90]在累加方案的基础上提出了功能相似但概念更简洁的折叠方案（Folding Scheme）. 折叠方案的示意图如图 8 所示.

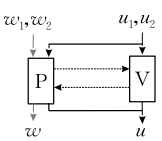
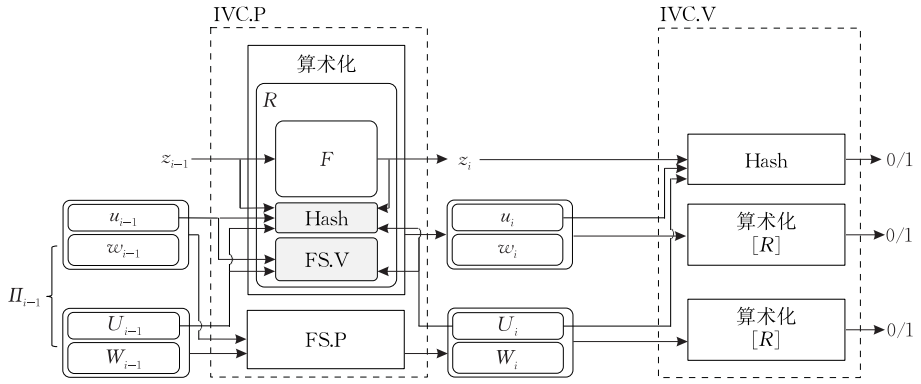


图 8　折叠方案示意图

该方案针对某个 NP 关系 $\mathcal{R}$，包含一个证明者和验证者. 证明者输入两个实例见证对 (u_1,w_1)，(u_2,w_2)，验证者仅输入两个实例 u_1,u_2，经过若干交互，证明者输出折叠后的实例见证对 (u,w)，验证者输出折叠后的实例 u. 方案的安全性要求若 $(u,w)\in\mathcal{R}$，则 $(u_1,w_1)\in\mathcal{R}$，$(u_2,w_2)\in\mathcal{R}$. 在性能方面，方案要求 V 参与折叠并验证 (u,w) 的开销小于直接验证 (u_1,w_1)，(u_2,w_2) 的开销. 易知折叠方案把对两个实例的验证归约成了对一个实例的验证.

Kothapalli 等人[90]利用其引入的折叠方案构造了高效的 IVC 方案——Nova，方案示意图如图 9 所示.



F: 函数电路; FS.V: 验证电路; Hash: 哈希电路; R: 增广电路

图 9 基于折叠方案的增量可验证计算方案构造示意图

每步输出的证明都由两个实例见证对组成. 在第 i 步,证明者首先对输入的两个实例见证对 (u_{i-1},w_{i-1}),(U_{i-1},W_{i-1}) 做折叠,生成新的实例见证对 (U_i,W_i);然后构造增广电路 R,包含函数电路 F、表达折叠方案验证算法 FS.V 的电路及为了压缩输出而需要表达哈希的电路,计算 $z_i=F(z_{i-1})$ 并结合 FS.P 的输出得到 R 的可满足赋值;接着按照某种电路的算术化生成新的实例见证对 (u_i,w_i);最后输出 IVC 的证明 $\Pi_i=((u_i,w_i),(U_i,W_i))$. 验证者通过验证证明,可确保前 i 步计算的正确性. 由于折叠方案的优异性能,Nova 的单步证明开销主要含两个多标量乘,递归开销主要含两个群标量乘,且通过额外地调用 SNARK,还可以把证明规模压缩至对数级别.

2022 年,Kothapalli 等人[91]借鉴 Nova 的构造思想,通过增加选择函数构造了高效的非均匀增量可验证计算方案——SuperNova. SuperNova 和 Nova 具有相同的单步证明开销和递归开销,但支持 IVC 中各步函数不一样的场景. 此外,Nova 在对电路做算术化时使用了 R1CS,该约束系统只能表达二次约束,在工程应用中具有一定的性能局限性. 2023 年,Mohnblatt[103]把 Nova 针对 R1CS 的折叠方案应用至可表达高阶约束的 Plonkish 约束系统中,但是证明者和验证者的群操作数量随着阶的增大而线性增长,使得高阶约束相比于二次约束的性能优势荡然无存. 同年,Kothapalli 等人[92]介绍了针对可定制约束系统(Customizable Constraint System,CCS)的多重折叠方案(Multi-folding Scheme),并基于多重折叠方案构造了高效的 IVC 方案——HyperNova,其证明开销主要含一个多标量乘,递归开销主要含一个群标量乘和对数级别的域运算及哈希运算. HyperNova 针对的 CCS 约束系统可无性能损失地转换为 R1CS 或 Plonkish,故 HyperNova 也支持高阶约束. 多重折叠方案是折叠方案的延伸,其可把超过两个的实例见证对折叠成一个. 虽然 Kothapalli 等人介绍了多重折叠方案的定义,但构造的具体方案仍是折叠了两个实例,使得 HyperNova 是一个 IVC 方案而非 PCD. 随后 Zhou 等人[94]在其基础上构造了折叠任意数目实例的多重折叠方案,并基于此构造了高效的 PCD 方案. Zheng 等人[95]介绍了一般折叠方案(Generic Folding Scheme),其类似多重折叠方案但可用于折叠来自不同电路的实例见证对. 基于该方案,Zheng 等人进一步对 PCD 方案增加了零知识性并支持函数不同即非均匀的情况,但未给出 NPCD 的形式化定义和构造.

2023 年,Liu 等人[104]利用 IVC 构造了 SNARK 证明聚合方案. 2024 年,Boneh 等人[105]构造了首个基于格的折叠方案——LatticeFold,其可用于构造抗量子的 IVC/PCD 方案. 同年,Nguyen 等人[106]提出了基于折叠方案的高可扩展性的 SNARK 构造框架——Mangrove,其在生成证明时具有较小的内存开销且高度并行化.

4.1.4 增量可验证计算方案高效实现

增量可验证计算方案通常需要在某步同时证明函数电路和验证电路,为了简便易懂,现有方案均把函数电路和验证电路组合成增广电路并作为整体来证明. 然而,除了 Chiesa 等人[87]的 PCD 方案,迄今所有的高效 IVC/PCD 方案都依赖椭圆曲线群,这使得函数电路和验证电路的计算处于不同的域中. 具体而言,给定一个定义在域 $\mathbb{F}_2$ 上的素数阶椭圆曲线循环群 $\mathbb{G}$,由循环群的性质,$\mathbb{G}$ 的阶可诱导出另一个域 $\mathbb{F}_1$,且该域的阶和 $\mathbb{F}_2$ 的阶不同,称 $\mathbb{F}_1$ 为群 $\mathbb{G}$ 的标量域(scalar field),$\mathbb{F}_2$ 为群 $\mathbb{G}$ 的基域(base field). 假设函数电路定义在域 $\mathbb{F}_1$ 上(在域 $\mathbb{F}_2$ 上同理),证明者在证明时通常使用多项式承诺方案对电路中的值做承诺,验证者验证时大多执行群操作,故表达验证算法的验证电路主要定义在域 $\mathbb{F}_2$ 上. 如果在编程实现

时仍把函数电路和验证电路组合成一个电路来证明，必然需要在电路中执行域模拟操作，导致电路规模极大扩张，效率极其低下. 为了避免域模拟，现有依赖椭圆曲线群的实现均采用椭圆曲线循环，其是一对椭圆曲线(E_1,E_2)，可构造出一对椭圆曲线群$(\mathbb{G}_1,\mathbb{G}_2)$，其中$\mathbb{G}_1$的标量域和基域分别为$\mathbb{F}_1$、$\mathbb{F}_2$、$\mathbb{G}_2$的标量域和基域分别为$\mathbb{F}_2$、$\mathbb{F}_1$，可知两个群的标量域互为对方的基域. 高效的椭圆曲线循环需要精心设计，若干工作[86,107-111]在该方面开展了深入的理论研究. 不同的增量可验证计算方案在实现时采用不同的方式来使用椭圆曲线循环，以下以 IVC 为例，介绍三种典型的实现方案.

(1) 基于 SNARK 的 IVC 高效实现

2014 年，Ben-Sasson 等人[86]首次编程实现了基于 SNARK 的增量可验证计算方案，以 IVC 为例，其电路构造示意图如图 10 所示.

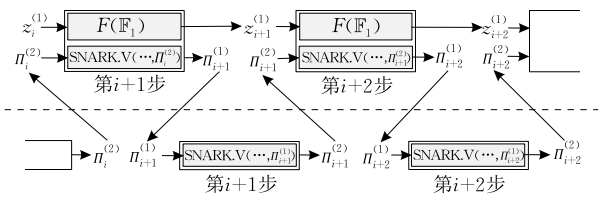


图 10　基于 SNARK 的 IVC 电路构造示意图

令Γ_1，Γ_2分别表示定义在群$\mathbb{G}_1$和$\mathbb{G}_2$上的SNARK，Γ_1证明的电路定义在$\mathbb{G}_1$的标量域$\mathbb{F}_1$上，Γ_2证明的电路定义在$\mathbb{G}_2$的标量域$\mathbb{F}_2$上. 在第 $i+1$ 步，证明者输入第 i 步的输出 $z_i^{(1)}$ 及证明 $\Pi_i^{(2)}$，首先调用 Γ_1 生成证明 $\Pi_{i+1}^{(1)}$，其证明 $z_{i+1}^{(1)}=F(z_i^{(1)})$且来自第 i 步的证明 $\Pi_i^{(2)}$ 满足 Γ_2 的验证算法；接着调用 Γ_2 生成证明 $\Pi_{i+1}^{(2)}$，其证明 $\Pi_{i+1}^{(1)}$ 满足 Γ_1 的验证算法；最后输出 $z_{i+1}^{(1)}$，$\Pi_{i+1}^{(2)}$. 在第 $i+2$ 步则重复第 $i+1$ 步的过程，以此类推后续步骤.

Ben-Sasson 等人[86]在实现时利用了 MNT4/MNT6 这个椭圆曲线循环，其中 MNT4 曲线的标量域是 MNT6 曲线的基域，而 MNT4 曲线的基域又是 MNT6 曲线的标量域. 然而，由于其增量可验证计算方案使用的 SNARK 依赖配对，所以 MNT4 和 MNT6 都是配对友好的椭圆曲线，其性能较差. 例如，为了实现 128 bits 的安全性，MNT4/MNT6 必须定义在约 800 bits 的素数域上. 同时，由于使用的 SNARK 的验证算法需要执行配对操作，验证电路需使用大量的算术门来表达，导致方案的递归开销仍比较大.

(2) 基于折叠方案的 IVC 高效实现

2022 年，Kothapalli 等人[90]基于折叠方案构造了高效的 IVC 方案——Nova，并做了编程实现，其电路构造示意图如图 11 所示.

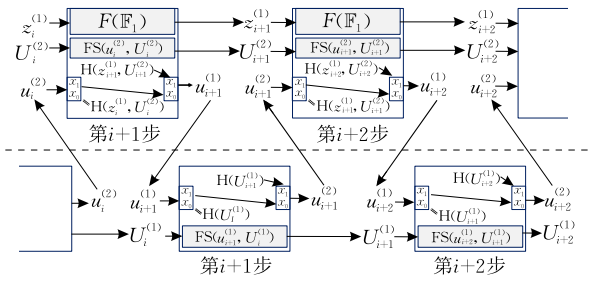


图 11　基于折叠方案的 IVC 电路构造示意图①

该实现方案改编至 Ben-Sasson 等人[86]的实现方案，主要是用折叠方案替换 SNARK. 在第 $i+1$ 步，证明者输入第 i 步的输出$z_i^{(1)}$及相关证明$(U_i^{(2)},W_i^{(2)})$，$(u_i^{(2)},w_i^{(2)})$，$(U_i^{(1)},W_i^{(1)})$，首先调用折叠方案把$(U_i^{(2)},W_i^{(2)})$，$(u_i^{(2)},w_i^{(2)})$折叠成$(U_{i+1}^{(2)},W_{i+1}^{(2)})$；然后构造实例见证对$(u_{i+1}^{(1)},w_{i+1}^{(1)})$，其证明 $z_{i+1}^{(1)}=F(z_i^{(1)})$且 $U_{i+1}^{(2)}$ 和 $u_i^{(2)}$ 被正确折叠成了 $U_{i+1}^{(2)}$；接着调用折叠方案把$(u_{i+1}^{(1)},w_{i+1}^{(1)})$，$(U_i^{(1)},W_i^{(1)})$折叠成$(U_{i+1}^{(1)},W_{i+1}^{(1)})$，并构造实例见证对$(u_{i+1}^{(2)},w_{i+1}^{(2)})$证明折叠过程的正确性；最后输出 $z_{i+1}^{(1)}$ 及证明$(U_{i+1}^{(2)},W_{i+1}^{(2)})$，$(u_{i+1}^{(2)},w_{i+1}^{(2)})$，$(U_{i+1}^{(1)},W_{i+1}^{(1)})$. 在上述过程中，为了保证折叠方案能正常调用且同一变量在两个曲线上使用时保持一致，在电路里使用哈希函数压缩输出并链接压缩结果. 在第 $i+2$ 步则重复第 $i+1$ 步的过程，以此类推后续步骤.

Nova 的启动阶段透明且依赖标准的椭圆曲线群，在实现时使用了 Pallas/Vesta(Pasta)椭圆曲线循环，其性能相比于 MNT4/MNT6 具有极大提升. 然而，Nova 的实现方案仍需要在每一步都表示出两个验证电路，分别定义在不同椭圆曲线的标量域上，使得递归开销存在一定的冗余.

(3) IVC 的 CycleFold 实现

2023 年，Kothapalli 等人[112]提出一个基于折叠方案的 IVC 的高效实现技术——CycleFold，其电路构造示意图如图 12 所示.

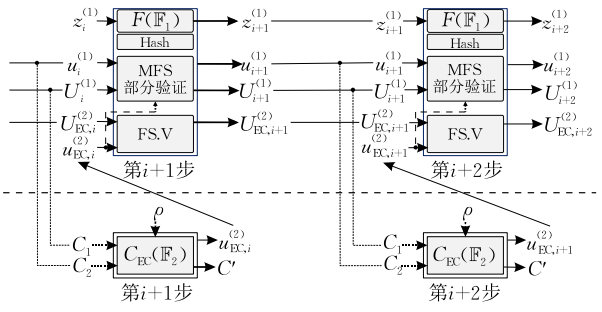


图 12　CycleFold 的电路构造示意图

① 证明者输入/输出的是实例见证对，但电路只需输入/输出实例.

该实现方案结合了 HyperNova[92] 和 Nova[90]. HyperNova 的验证算法需要执行定义在标量域上的对数级别的域操作和定义在基域上的群操作 $C'=C_1+\rho C_2$，其中 C_1, C_2 为基域上的群元素，ρ 为标量域元素. 若在实现时不使用椭圆曲线循环，则需要在标量域上模拟群操作，这带来极大的电路规模扩张；而若和 Nova 一样，使用椭圆曲线循环，则需要在基域上模拟对数级别的域操作，这同样带来极大的电路规模扩张.

为了解决上述问题，CycleFold 首先在 E_1 曲线上做证明，在第 $i+1$ 步，根据 HyperNova 的构造，电路需要表达出多重折叠方案的验证算法，即把 $u_i^{(1)}$ 和 $U_i^{(1)}$ 折叠成 $U_{i+1}^{(1)}$，为了避免模拟其中的群操作 $C'=C_1+\rho C_2$，把其单独表示为一个电路 C_{EC} 并定义在 E_2 曲线的标量域上，由椭圆曲线循环的性质，电路 C_{EC} 无需域模拟；其次，为了证明电路 C_{EC} 计算的正确性，CycleFold 用 Nova 的折叠方案把表达该电路正确计算的实例 $u_{\mathrm{EC},i}^{(2)}$ 和来自上一步证明中的实例 $U_{\mathrm{EC},i}^{(2)}$ 折叠成 $U_{\mathrm{EC},i+1}^{(2)}$，并把该折叠的验证电路和 HyperNova 除去群操作的验证电路组合起来，由椭圆曲线循环的性质，组合后的电路中几乎不存在域模拟操作；然后，由组合的电路构造实例 $u_{i+1}^{(1)}$；最后输出 $u_{i+1}^{(1)}, U_{i+1}^{(1)}, U_{\mathrm{EC},i+1}^{(2)}$. 在整个过程中，验证电路几乎没有做域模拟操作，且与 Nova 的实现不同，CycleFold 在 E_1 的标量域上可以认为表示出了完整的验证电路，而在 E_2 的标量域上仅需表示出部分验证电路，即验证算法中的群操作，有效减少了冗余的递归开销.

4.1.5 增量可验证计算方案密码学理论研究

除了研究增量可验证计算方案的高效构造及高效实现，以使其更广泛地应用到实际场景中，若干工作还从密码学理论方面对其做了深入研究. 2008 年，Valiant[76] 首次介绍了 IVC 的概念并提供了两个构造，其中一个构造假设存在一个非标准且难以置信的密码学原语，而另一个构造的安全性没有任何归约证明. 2010 年，Chiesa 等人[77] 首次介绍了 PCD 的概念并提供了在标准密码学假设下可证明安全性的 PCD 构造. 2013 年，Bitansky 等人[18] 证明假设存在抗碰撞的哈希函数，那么任意公开可验证的 SNARK 都可以被高效地转换成一个针对常数深度分布式计算或多项式深度链式计算的公开可验证 PCD 方案，且任意这样公开可验证的 PCD 方案又可被高效地转换成复杂度保留的公开可验证的 SNARK 或 PCD 方案，其中复杂度保留指无昂贵的预处理且证明者的时间/空间复杂度与经典 NP 验证所需的基本相同.

2019 年，Naor 等人[113] 提出了一个新的 IVC 构造框架，即使用同态加密和增量可更新的概率可验证明来构造 IVC，相比于基于 SNARK 的构造，该构造的安全性基于可证伪的假设. 然而，该类构造中的函数必须是确定性的，且完备性仅在中间证明都是诚实生成的情况下才成立，如果敌手在某步生成了通过验证的证明，那么诚实方对于后续的计算无法再生成新的证明. 2022 年，Paneth 等人[114] 利用批量论证构造了针对任意多项式步数机器执行的 IVC，并且其安全性基于可证伪假设.

2022 年，Chen 等人[115] 在低度随机谕言模型(Low-degree Random Oracle Model)下构造了透明的 SNARKs，使得利用 SNARKs 构造透明 PCD 时不用启发式地实例化谕言并非黑盒地使用它，但在现实世界中实例化低度随机谕言模型比较困难. 2023 年，Chen 等人[116] 介绍且实例化了算术化随机谕言模型(Arithmetized Random Oracle Model)，并在给定抗碰撞哈希函数的条件下，构造了针对任意深度合规谓词的 PCD 方案. 同年，Chiesa 等人[117] 证明由具有直线知识可靠性(Straightline Knowledge Soundness) 的 SNARKs 构造的 PCD 和底层的 SNARKs 具有相同的安全性，即对 SNARKs 的递归组合不会带来安全性损失. Hall-Andersen 等人[118] 证明如果证明系统可以增量地接收一个长的见证作为输入且具有零知识性，那么 Valiant[76] 的猜想是正确的，即在标准随机谕言模型下无法构造 IVC.

4.2 基于电路的证明系统组合

在过去的十几年中，SNARKs 的高效设计和实现取得了巨大进展并已有诸多实际应用，但在效率方面，目前的 SNARKs 要么具有较小的证明开销，但证明规模较大且验证较慢，如 Spartan[119]、Orion[120]、Brakedown[121] 等，要么具有较小的证明规模和较小的验证开销，但证明开销较大，如 Pinocchio[122]、Groth16[58] 等. 为了结合两者的性能优势，若干工作基于电路对证明系统做递归组合，其示意如图 13 所示.

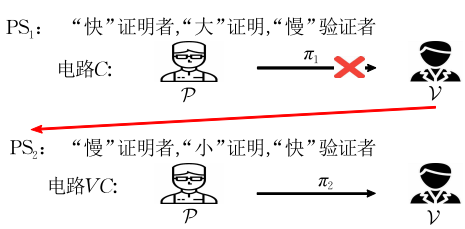


图 13 基于电路的证明系统组合示意图

假设证明系统 PS_1 具有较小的证明开销但相对较大的证明规模和较大的验证开销，证明系统 PS_2 具有较小的证明规模和较小的验证开销但相对较大的证明开销. 对于一个电路 C，使用证明系统 PS_i（其中 $i=1,2$）的证明开销、证明规模和验证开销分别表示为 $PS_i(C).\mathcal{P}$、$PS_i(C)$ 和 $PS_i(C).\mathcal{V}$. 则可知 $PS_1(C).\mathcal{P}<PS_2(C).\mathcal{P}$，$PS_1(C)>PS_2(C)$，$PS_1(C).\mathcal{V}>PS_2(C).\mathcal{V}$. 为了组合两个证明系统，首先调用 PS_1 生成对电路 C 的证明 π_1，但不直接把 π_1 发送给 PS_1 的验证者，而是把其验证算法表示成电路 VC，接着调用 PS_2 证明 π_1 是 VC 的可满足输入，并把生成的证明 π_2 发送给 PS_2 的验证者做验证，记组合的证明系统为〈PS_1，PS_2〉，可知其证明电路 C 的证明开销、证明规模和验证开销分别为 $PS_1(C).\mathcal{P}+PS_2(VC).\mathcal{P}$，$PS_2(VC)$ 和 $PS_2(VC).\mathcal{V}$. 证明系统 PS_1、PS_2 和〈PS_1，PS_2〉的性能对比如表 4 所示.

表 4　普通和递归组合的证明系统性能对比

证明系统	证明开销	证明规模	验证开销
PS_1	$PS_1(C).\mathcal{P}$	$PS_1(C)$	$PS_1(C).\mathcal{V}$
PS_2	$PS_2(C).\mathcal{P}$	$PS_2(C)$	$PS_2(C).\mathcal{V}$
〈PS_1，PS_2〉	$PS_1(C).\mathcal{P}+PS_2(VC).\mathcal{P}$	$PS_2(VC)$	$PS_2(VC).\mathcal{V}$

虽然 PS_1 具有较大的验证开销，但其复杂度一般也是亚线性甚至对数级别的，故电路 VC 的规模要远小于电路 C 的规模，因此在证明开销方面，虽然 PS_2 相较于 PS_1 具有较大的证明开销，但 $PS_2(VC).\mathcal{P}$ 要远小于 $PS_1(C).\mathcal{P}$，故组合后的证明系统〈PS_1，PS_2〉具有接近于 PS_1 的证明开销. 在证明规模方面，可知 $PS_1(C)>PS_2(C)>PS_2(VC)$，故〈PS_1，PS_2〉具有最小的证明规模. 在验证开销方面，可知 $PS_1(C).\mathcal{V}>PS_2(C).\mathcal{V}>PS_2(VC).\mathcal{V}$，故〈$PS_1$，$PS_2$〉具有最小的验证开销. 综上，组合后的证明系统〈$PS_1$，$PS_2$〉在性能方面结合了两个证明系统的优势，整体性能更为优越.

利用电路对证明系统做递归组合的思想较为平凡，但如何选择合适的证明系统来安全高效地组合需要深入透彻的分析研究. 截至目前，已有若干学术研究和工程项目对不同的证明系统做了组合，产生了卓越的性能，且该方向的研究正呈现出蒸蒸日上的趋势.

2021 年，Golovnev 等人[121]设计了具有线性证明复杂度且兼容任意域（满足一定大小）的 SNARK 协议——Brakedown，并指出可以通过递归组合 Spartan[119]、Xiphos[59]、Groth16[58] 等 zkSNARKs 来减少证明规模并实现零知识性. 2022 年，Polygon Zero 团队结合 Plonk 协议[61]的信息论证明和基于快速里德-所罗门编码接近性交互谕示证明（Fast Reed-Solomon Interactive Oracle Proofs of Proximity，FRI）[35]的多项式承诺方案设计了一个 SNARK 协议——Plonky2①. 该协议的性能和 FRI 的码率紧密相关，当码率较大时，协议具有较小的证明开销但证明规模较大，当码率较小时，协议具有较小的证明规模但证明开销较大，这种性能的灵活性使其非常适合做证明系统的组合. 如在区块链里对交易做聚合证明时，可以先使用证明高效的 Plonky2 分别生成对若干交易的证明 π_i，再用证明规模小的 Plonky2 生成对这些 π_i 的证明 π，最后把小规模的 π 传至链上以供验证. 2022 年，Belling 等人[123]通过递归组合 GKR 协议[101]和 Groth16 协议[58]高效地证明了 MiMC 哈希函数[124]的正确执行. Xie 等人[26]考虑区块链中跨链通信的场景，首先设计了 Virgo 协议[64]的分布式变种——deVirgo 以提高证明效率，然后递归组合了 deVirgo 和 Groth16 协议以减少证明规模和验证开销，最后基于这些技术设计了首个实用且无需额外可信假设的跨链桥协议——zkBridge. 2023 年，Campanelli 等人[125]通过递归组合 Spartan 协议[119]和 Groth16 协议，设计了无需 FFT 且具有几乎线性证明复杂度、常数级别通信和验证复杂度的 SNARK 协议——Testudo. Xie 等人[120]考虑基于编码的具有线性证明复杂度的零知识证明协议[121,126-127]，观察到这些协议的证明由 $O(N^{1/c})$ 个码字组成，且验证算法可以表示成码字和一些公开向量的内积，其中 N 表示电路规模. 为了降低通信复杂度，他们把验证算法表示成电路，并调用 Virgo 协议[64]做递归组合，最后构造了具有 $O(N)$ 证明复杂度、$O(\log^2 N)$ 通信复杂度和 $O(N)$ 验证复杂度的零知识证明协议——Orion.

5　针对复合断言的复合零知识证明

在现实场景中应用零知识证明时，实际问题转成的断言除了只包含代数群上的代数操作或只包含需要用电路表达的非代数操作，还有可能同时包含两种类型的操作. 例如，在比特币交易所偿付能力证明场景中，交易所拟向客户证明他拥有足够的资金来偿还债务，即他拥有足够数量的比特币地址. 由比特币地址的生成原理，该场景的证明可抽象成以下断言：对于公开的 h，证明者证明他知道秘密 x 满足

① Plonky2. https://github.com/0xPolygonZero/plonky2

$h=\mathrm{H}(g^x)$，其中 H 是一个哈希函数如 SHA-256，g 是椭圆曲线群的生成元. 该断言既包含代数操作 g^x 又包含非代数操作 H.

一种平凡的证明方式是把断言转换成一种形式，即代数断言或非代数断言，然后调用针对代数断言的 Σ 协议或针对非代数断言的零知识证明协议完成证明. 然而，一方面，把由电路表达的非代数断言转换成代数断言需要把电路的每个门都视为其输入输出的代数关系，如乘法或加法，在用 Σ 协议证明时，每个门对应的代数关系都会给证明开销增加若干公钥操作并给证明规模增加若干群元素，当电路规模较大时，这个开销难以接受. 例如，Agrawal 等人[128]指出，把表达哈希函数或分组密码的电路视为代数断言证明时，需要数以万计的群幂操作和数以万计的群元素. 在另一方面，若把代数断言表示成电路，则会使得电路的规模极其庞大. 表达一个群幂操作可能就需要成千上万个门，而针对电路的零知识证明协议的性能又和电路规模紧密相关，故这种方式也使得证明效率极其低下. 为了高效证明既含代数操作又含非代数操作的复合断言，可以使用复合零知识证明技术，即使用 Σ 协议证明代数部分，使用针对电路的零知识证明协议证明非代数部分，并选择性地使用链接协议证明同时存在于两部分中的变量满足一致性.

实际应用中出现的复合断言形式多样，但可归纳为："若干被代数承诺的值使得电路可满足"，该关系可形式化地表示为[129]

$$\{(ck,(x,(c_j)_{j\in[\ell]})\in\mathcal{D}_x\times\mathcal{C}^\ell;((m_j)_{j\in[\ell]},(o_j)_{j\in[\ell]},w)\in\mathcal{D}_1\times\cdots\times\mathcal{D}_\ell\times\mathcal{O}^\ell\times\mathcal{D}_w):\bigwedge_{j\in[\ell]}c_j=\mathrm{Com}(ck,m_j,o_j)\wedge C(x;(m_j)_{j\in[\ell]},w)=1\},$$

其中 $\mathcal{D}_x,\mathcal{C},\mathcal{D}_1,\cdots,\mathcal{D}_\ell,\mathcal{O},\mathcal{D}_w$ 表示不同元素所处的空间，c_j 为对 m_j 的代数承诺如 Pedersen 承诺，C 为电路. 证明该关系的零知识证明又被称为承诺并证明的零知识证明(Commit-and-Prove Zero-Knowledge Proof，CP-ZKP)[130-131]. 特别的，若满足简洁性、非交互及知识论证，则又称承诺并证明的 SNARK(Commit-and-Prove SNARK，CP-SNARK)[132]，相关总结如表 5 所示. 该类协议已被应用至众多实际场景中，如匿名凭证[16,128,133]、可验证加密[134]、电子投票[134]、证明组合[42,119,135-136]、比特币交易所偿付能力证明[128]、集合成员关系证明[137]等，且已被零知识证明标准化组织 ZKProof 标准化[129].

表 5 针对复合断言的承诺并证明的零知识证明协议总结

协议		时间	证明开销	证明规模	验证开销	公开抛币	启动阶段	关键模块
CGM16[16]	构造一	2016	$O(\lvert w\rvert)$ pub $O(\lvert C\rvert)$ sym	$O(\lvert C\rvert+\lvert w\rvert)$	$O(\lvert w\rvert)$ pub $O(\lvert C\rvert)$ sym	×	透明	混淆电路
	构造二	2016	$O(\lambda)$ pub $O(\lvert C\rvert+\lvert w\rvert\lambda)$ sym	$O(\lvert C\rvert+\lvert w\rvert\lambda)$	$O(\lambda)$ pub $O(\lvert C\rvert+\lvert w\rvert\lambda)$ sym	×	透明	混淆电路
AGM18[128]		2018	$O(\lvert C\rvert+\lambda)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(1)$	$O(\lvert w\rvert+\lambda)$ pub	√	可信	LPCP
Bulletproofs[34]		2018	$O(\lvert C\rvert)$ pub	$O(\log\lvert C\rvert)$	$O(\lvert C\rvert+\lvert w\rvert)$ pub	√	透明	内积论证
BHH+19[138]		2019	$O(\lvert w\rvert+\lambda)$ pub $O(\lvert C\rvert\lambda)$ sym	$O(\lvert C\rvert\lambda+\lvert w\rvert)$	$O(\lvert w\rvert+\lambda)$ pub $O(\lvert C\rvert\lambda)$ sym	√	透明	MPCitH
LegoSNARK[132]	LegoAC1	2019	$O(\lvert C\rvert)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(1)$	$O(\lvert w\rvert)$ pub	√	可信	$\mathrm{CP}_{\mathrm{had}}$ $\mathrm{CP}_{\mathrm{lin}}$
	LegoAC2	2019	$O(\lvert C\rvert)$ pub	$O(\log\lvert C\rvert)$	$O(\log\lvert C\rvert+\lvert w\rvert)$ pub	√	可信	$\mathrm{CP}_{\mathrm{had}}$ $\mathrm{CP}_{\mathrm{lin}}$
	LegoUAC	2019	$O(\lvert C\rvert)$ pub	$O(\log^2\lvert C\rvert)$	$O(\log^2\lvert C\rvert+\lvert w\rvert)$ pub	√	通用可更新	$\mathrm{CP}_{\mathrm{had}}$ $\mathrm{CP}_{\mathrm{sfprm}}$
Lunar[139]		2021	$O(\lvert C\rvert+\lvert w\rvert)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(\lvert w\rvert)$	$O(\lvert w\rvert)$ pub	√	通用可更新	PIOP
ECLIPSE[140]		2022	$O(\lvert C\rvert+\lvert w\rvert)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(\log\lvert w\rvert)$	$O(\lvert w\rvert)$ pub	√	通用可更新	PIOP
FT22[141]		2022	$O(\lvert C\rvert)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(1)$	$O(1)$ pub	√	通用可更新	PIOP
ZCY+23[33]		2023	$O(\lambda)$ pub $O(\lvert C\rvert\log\lvert C\rvert)$ sym	$O(\log^{O(1)}\lvert C\rvert+\lambda)$	$O((\lvert w\rvert+\lambda)^2)$ pub $O(\lvert C\rvert)$ sym	√	透明	MPCitH

注：① pub 指公钥操作，一般是群幂运算；sym 指对称密钥操作；$|C|$ 表示电路规模，CGM16、BHH+19 针对布尔电路，其余协议针对算术电路；$|w|$ 表示被承诺输入的长度；λ 表示安全参数；证明规模由群或域元素的数量衡量.

② 公开抛币表示验证者生成的随机数来自公开的随机源，满足该条件的协议可以使用 Fiat-Shamir 转换实现非交互；非公开抛币即私密抛币表示验证者生成的随机数来自私密的随机源，此类协议无法实现非交互.

③ $\mathrm{CP}_{\mathrm{had}}$，$\mathrm{CP}_{\mathrm{lin}}$，$\mathrm{CP}_{\mathrm{sfprm}}$ 是 LegoSNARK 里定义的 CP-SNARK 基本组件；LPCP 指线性概率可验证明(Linear Probabilistic Checkable Proof)；MPCitH 指头脑中的安全多方计算(MPC-in-the-head)；PIOP 指多项式交互式谕示证明(Polynomial Interactive Oracle Proof).

在设计 CP-ZKP 协议时，主要先设计针对电路的零知识证明协议，然后使用链接协议证明 c_j 所承诺的值 m_j 和电路里使用的 m_j 是一致的. CP-ZKP 协议的性能等指标通常依赖于其使用的针对电路的零知识证明协议，且有时并不需要链接协议. 以下简要介绍 CP-ZKP 协议的研究现状.

若干工作利用安全多方计算技术构造 CP-ZKP 协议. 2016 年，Chase 等人[16]考虑到基于 RSA 或 DSA 签名的匿名凭证中会产生复合断言，为了高效地证明，他们利用 Jawurek 等人[142]的基于混淆电路的零知识证明协议和额外的链接协议设计了两个 CP-ZKP 协议来证明断言："给定一个代数承诺 C，证明者知道承诺的打开 x 且满足 $f(x)=1$"，其中函数 f 被表达成布尔电路. 该协议的公钥操作数量独立于电路规模. 然而，由于其使用的零知识证明协议是私密抛币的，该 CP-ZKP 协议天然无法实现非交互. 2019 年，Backes 等人[138]利用基于 MPC-in-the-Head(MPCitH)的 ZKBoo[143]/ZKB++协议[144]和链接协议设计了透明且非交互的高效 CP-ZKP 协议. 2023 年，Zhang 等人[33]结合基于可验证秘密分享的 Σ 协议和基于 MPCitH 的 Ligero[145]/Ligero++协议[65]，设计了无需链接协议的透明非交互 CP-ZKP 协议. 基于 MPCitH 的 CP-ZKP 协议具有较小的证明开销和验证开销，但其证明规模相对较大.

2018 年，Bünz 等人[34]提出零知识证明协议 Bulletproofs，其证明算术电路可满足时在关系的实例中额外增加了若干 Pedersen 承诺，并证明这些承诺的打开和电路里的值满足某些特定的代数关系，也可视为一种 CP-ZKP 协议. 由于关系实例中的每个 Pedersen 承诺针对的都是单个域元素，且复合断言具有一定的特殊性，在证明电路的协议里可以直接使用这些 Pedersen 承诺，而无需额外的链接协议证明其打开值和电路证明协议里使用的值的一致性. 然而，该 CP-ZKP 协议证明的复合断言不具备一般性，不能模块化地应用到其他复合断言中.

2018 年，Agrawal 等人[128]提出了一个框架来证明由逻辑与、逻辑或、表达成代数操作的函数和表达成电路的函数所构造的断言，基于该框架可处理任意形式的复合断言. 针对电路部分，他们使用了基于线性概率可验证明(Linear Probabilistic Checkable Proof，LPCP)的 Pinocchio 协议[122]，其具有常数级别的证明规模且验证的公钥操作数量独立于电路规模，但协议需要可信的启动阶段.

2019 年，Campanelli 等人[132]提出了一个证明复合断言的框架——LegoSNARK，该框架可以模块化地组合不同的证明组件. 他们设计了若干证明代数关系的 CP-SNARK 基本组件，包括 $\mathrm{CP}_{\mathrm{link}}$ 证明两个类 Pedersen 承诺的打开一致、$\mathrm{CP}_{\mathrm{lin}}$ 证明某个被承诺的向量满足某个线性关系、$\mathrm{CP}_{\mathrm{had}}$ 证明三个被承诺的向量满足哈达玛积关系、$\mathrm{CP}_{\mathrm{sfprm}}$ 证明某个被承诺的向量满足某个置换关系和 $\mathrm{CP}_{\mathrm{mm}}$ 证明两个被承诺矩阵的乘积等于某个承诺或公开的矩阵，并基于这些组件构造了若干 CP-SNARK 协议，包括 Groth16 协议[58]的承诺并证明变种——LegoGro16、LegoAC1、LegoAC2、LegoUAC 等. 然而，除了 LegoUAC 协议，其他协议都需要可信的启动阶段，而 LegoUAC 协议的启动阶段虽然是通用可更新的，其证明规模仍比较大、验证开销仍比较高.

若干工作利用基于多项式交互式谕示证明(Polynomial Interactive Oracle Proof，PIOP)的 SNARK 构造通用可更新的 CP-SNARK 协议. 2021 年，Campanelli 等人[139]设计了一个通用的编译器来把信息论安全的证明如 PIOP 编译成 CP-SNARK，并利用设计的通用可更新的 SNARK 协议和链接协议实例化 CP-SNARK. 然而，其证明规模和验证开销与复合断言中代数承诺的数量成线性关系. 2022 年，Aranha 等人[140]也设计了一个通用的编译器——ECLIPSE，把 PIOP 编译成 CP-SNARK，他们用链接协议和三种主流的通用可更新 SNARK 协议：Sonic[97]、Plonk[61]、Marlin[146]做了实例化. 为了使 CP-SNARK 协议的证明规模简洁于复合断言中代数承诺的数量，他们还利用压缩的 Σ 协议理论[25]对链接协议的证明规模做了压缩，使其降至对数级别. 同年，Fiore 等人[141]构造了 Marlin 协议的承诺并证明变种，相比于 ECLIPSE，该协议的证明规模和验证开销独立于复合断言中被承诺向量的维数. 然而，该协议针对的复合断言中仅含一个代数承诺，且该承诺使用的是 KZG 多项式承诺，而非传统的 Pedersen 承诺.

此外，若干工作研究 CP-ZKP 协议但并不针对复合断言这个应用场景. 2015 年，Costello 等人[135]把二次算术程序(Quadratic Arithmetic Program，QAP)一般化至 MultiQAP，允许证明者对数据做承诺并用在多个证明中，他们进而构造了证明者可以跨证明共享状态的可验证计算协议——Geppetto. 2016 年，Lipmaa[136]提出了一个证明友好的可提取承诺方案，并用其构造了针对不同语言的证明高效的自适应 CP-SNARK 协议，其证明开销主要含线性级别的密码学操作.

6 未来研究方向

零知识证明的递归与复合技术发展尚不成熟，仍然存在很多问题亟待解决，有许多值得进一步研究的方向.

(1)针对代数断言的递归零知识证明

递归零知识证明早期主要用在内积论证这一证明内积关系的 Σ 协议，针对的代数关系较为局限. 后续工作提出压缩的 Σ 协议理论，继承 Σ 协议灵活性和通用性的同时能将其通信复杂度从线性级别压缩至对数级别. 因此，在未来可以考虑把递归零知识证明和压缩的 Σ 协议理论应用到更多不同的密码学场景中. 同时，不同的特定场景可能会呈现出错综复杂的代数断言，难以直接且高效地应用压缩的 Σ 协议理论. 为此，针对特定场景，可以进一步研究压缩的 Σ 协议理论的高效应用方法.

(2)针对非代数断言的递归零知识证明

目前该方向主流的应用是构造增量可验证计算方案和组合证明系统. 对于增量可验证计算方案，目前主流的构造基于分割累加方案或(多重)折叠方案. 前者相比于后者能更高效地支持非均匀的函数集合及查询关系，但后者相比于前者概念更简洁且工程应用较成熟，如何通过技术手段综合两者的优势是一个值得研究的方向. 此外，分布式的增量可验证计算方案虽然已经有了高效的理论构造，但在实现和应用方面仍有所欠缺. 因此，未来可以探索研究该类方案的高效实现架构及在实际场景中的应用手段.

对于证明系统组合，核心思想是用电路表达出内部证明系统的验证算法. 现有组合的证明系统利用了 FRI 的码率或结合了基于 sumcheck 的 SNARK 协议与 Groth16 协议. 在未来可以更深入地利用组合的核心思想，尝试对更多种类的 SNARK 协议做组合，以获得非平凡的性能及功能.

(3)针对复合断言的复合零知识证明

目前该方向主流的研究是构造针对复合断言的 CP-SNARK 协议，主要的技术手段是改进 SNARK 协议以构造其承诺并证明的变种. 然而目前 CP-SNARK 协议均基于早期的 SNARK 协议，其性能相对较差. 因此，未来可以尝试改进性能更卓越的 SNARK 协议，构造其承诺并证明的变种，以获得更高的性能并支持更多功能，例如无需 FFT、支持电路表达高阶约束等.

7 结 论

零知识证明是数据安全流通的关键技术之一. 零知识证明的递归技术可以丰富针对代数断言的和针对非代数断言的零知识证明协议的性能及功能，零知识证明的复合技术使复合断言能被高效地证明. 本文系统梳理了零知识证明递归与复合技术的研究现状. 在针对代数断言的递归零知识证明方面，详细综述了内积论证协议，并重点对比分析了基于 Pedersen 承诺方案的内积论证协议. 在针对非代数断言的递归零知识证明方面，重点对比分析了增量可验证计算方案，并详细综述了基于电路组合证明系统的研究. 在针对复合断言的复合零知识证明方面，详细对比分析了承诺并证明的零知识证明协议. 总体而言，零知识证明的递归与复合技术还处于发展初期，未来还应考虑利用递归技术高效证明更复杂的代数断言、进一步提升增量可验证计算方案的性能及功能、对更多种类的零知识证明协议做组合以获得非凡的性能特征、提高承诺并证明的零知识证明协议高效处理高阶约束的能力等，促进零知识证明在更多实际场景中的广泛应用.

参 考 文 献

[1] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems//Proceedings of the 17th Annual ACM Symposium on Theory of Computing. Providence, USA, 1985: 291-304

[2] Goldreich O, Micali S, Wigderson A. How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design//Proceedings of the 6th Annual International Cryptology Conference. Santa Barbara, USA, 1986: 171-185

[3] Fiege U, Fiat A, Shamir A. Zero knowledge proofs of identity//Proceedings of the 19th Annual ACM Symposium on Theory of Computing. New York, USA, 1987: 210-217

[4] Guillou L C, Quisquater J J. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge//Proceedings of the 8th Annual International Cryptology Conference. Santa Barbara, USA, 1988: 216-231

[5] Naor M, Yung M. Public-key cryptosystems provably secure against chosen ciphertext attacks//Proceedings of the 22nd Annual ACM Symposium on Theory of Computing. Baltimore, USA, 1990: 427-437

[6] Sahai A. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security//Proceedings of the 40th

Annual Symposium on Foundations of Computer Science. New York, USA, 1999: 543-553

[7] Goldreich O, Micali S, Wigderson A. How to play any mental game, or a completeness theorem for protocols with honest majority//Proceedings of the 19th Annual ACM Symposium on Theory of Computing. New York, USA, 1987: 218-229

[8] Sasson E B, Chiesa A, Garman C, et al. Zerocash: Decentralized anonymous payments from bitcoin//Proceedings of the 2014 IEEE Symposium on Security and Privacy. Berkeley, USA, 2014: 459-474

[9] Kalodner H, Goldfeder S, Chen X, et al. Arbitrum: Scalable, private smart contracts//Proceedings of the 27th USENIX Security Symposium. Baltimore, USA, 2018: 1353-1370

[10] Fang Z, Darais D, Near J P, et al. Zero knowledge static program analysis//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event, Republic of Korea, 2021: 2951-2967

[11] Grubbs P, Arun A, Zhang Y, et al. Zero-knowledge middleboxes//Proceedings of the 31st USENIX Security Symposium. Boston, USA, 2022: 4255-4272

[12] Liu T, Xie X, Zhang Y. ZkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event, Republic of Korea, 2021: 2968-2985

[13] Zhang Y, Genkin D, Katz J, et al. vSQL: Verifying arbitrary SQL queries over dynamic outsourced databases//Proceedings of the 2017 IEEE Symposium on Security and Privacy. San Jose, USA, 2017: 863-880

[14] Cramer R. Modular Design of Secure Yet Practical Cryptographic Protocols [Ph.D. dissertation]. University of Amsterdam, Netherlands, 1996

[15] Thaler J. Proofs, arguments, and zero-knowledge. Foundations and Trends in Privacy and Security, 2022, 4(2-4): 117-660

[16] Chase M, Ganesh C, Mohassel P. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials//Proceedings of the 36th Annual International Cryptology Conference. Santa Barbara, USA, 2016: 499-530

[17] Bootle J, Cerulli A, Chaidos P, et al. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting//Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Vienna, Austria, 2016: 327-357

[18] Bitansky N, Canetti R, Chiesa A, et al. Recursive composition and bootstrapping for SNARKs and proof-carrying data//Proceedings of the 45th Annual ACM Symposium on Theory of Computing. Palo Alto, USA, 2013: 111-120

[19] Richardson R, Kilian J. On the concurrent composition of zero-knowledge proofs//Proceedings of the 18th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Prague, Czech Republic, 1999: 415-431

[20] Kilian J, Petrank E. Concurrent and resettable zero-knowledge in poly-logarithmic rounds//Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. Heraklion, Greece, 2001: 560-569

[21] Prabhakaran M, Rosen A, Sahai A. Concurrent zero knowledge with logarithmic round-complexity//Proceedings of the 43rd Annual Symposium on Foundations of Computer Science. Vancouver, Canada, 2002: 366-375

[22] Deng Y, Goyal V, Sahai A. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy//Proceedings of the 50th Annual Symposium on Foundations of Computer Science. Atlanta, USA, 2009: 251-260

[23] Canetti R, Lin H, Paneth O. Public-coin concurrent zero-knowledge in the global hash model//Proceedings of the 10th Theory of Cryptography Conference. Tokyo, Japan, 2013: 80-99

[24] Goyal V, Gupta D, Jain A. What information is leaked under concurrent composition?//Proceedings of the 33rd Annual International Cryptology Conference. Santa Barbara, USA, 2013: 220-238

[25] Attema T, Cramer R. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics//Proceedings of the 40th Annual International Cryptology Conference. Santa Barbara, USA, 2020: 513-543

[26] Xie T, Zhang J, Cheng Z, et al. zkBridge: Trustless cross-chain bridges made practical//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles, USA, 2022: 3003-3017

[27] Goldreich O. Zero-knowledge twenty years after its invention. https://eprint.iacr.org/2002/186

[28] Li F, McMillin B. A survey on zero-knowledge proofs. Advances in Computers, 2014, 94: 25-69

[29] Nitulescu A. zk-SNARKs: A gentle introduction. https://www.di.ens.fr/~nitulesc/files/Survey-SNARKs.pdf

[30] Li Wei-Han, Zhang Zong-Yang, Zhou Zi-Bo, et al. An overview on succinct non-interactive zero-knowledge proofs. Journal of Cryptologic Research, 2022, 9(3): 379-447 (in Chinese)
(李威翰，张宗洋，周子博等. 简洁非交互零知识证明综述. 密码学报，2022，9(3)：379-447)

[31] Partala J, Nguyen T H, Pirttikangas S. Non-interactive zero-knowledge for blockchain: A survey. IEEE Access, 2020, 8: 227945-227961

[32] Baum C, Dittmer S, Scholl P, et al. SoK: Vector OLE-based zero-knowledge protocols. https://eprint.iacr.org/2023/857

[33] Zhang M, Chen Y, Yao C, et al. Sigma protocols from verifiable secret sharing and their applications//Proceedings of the 29th International Conference on the Theory and Application of Cryptology and Information Security. Guangzhou, China, 2023: 208-242

[34] Bünz B, Bootle J, Boneh D, et al. Bulletproofs: Short proofs for confidential transactions and more//Proceedings of the 2018 IEEE Symposium on Security and Privacy. San Francisco, USA, 2018: 315-334

[35] Ben-Sasson E, Bentov I, Horesh Y, et al. Fast reed-solomon interactive oracle proofs of proximity//Proceedings of the 45th International Colloquium on Automata, Languages, and Programming. Prague, Czech Republic, 2018: 14:1-14:17

[36] Bootle J, Chiesa A, Sotiraki K. Sumcheck arguments and their applications//Proceedings of the 41st Annual International Cryptology Conference. Virtual Event, 2021: 742-773

[37] Kothapalli A, Parno B. Algebraic reductions of knowledge//Proceedings of the 43rd Annual International Cryptology Conference. Santa Barbara, USA, 2023: 669-701

[38] Hoffmann M, Klooß M, Rupp A. Efficient zero-knowledge arguments in the discrete log setting, revisited//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, UK, 2019: 2093-2110

[39] Daza V, Ràfols C, Zacharakis A. Updateable inner product argument with logarithmic verifier and applications//Proceedings of the 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography. Edinburgh, UK, 2020: 527-557

[40] Chung H, Han K, Ju C, et al. Bulletproofs+: Shorter proofs for a privacy-enhanced distributed ledger. IEEE Access, 2022, 10: 42081-42096

[41] Kim S, Lee H, Seo J H. Efficient zero-knowledge arguments in discrete logarithm setting: Sublogarithmic proof or sublinear verifier//Proceedings of the 28th International Conference on the Theory and Application of Cryptology and Information Security. Taipei, China, 2022: 403-433

[42] Wahby R S, Tzialla I, Shelat A, et al. Doubly-efficient zkSNARKs without trusted setup//Proceedings of the 2018 IEEE Symposium on Security and Privacy. San Francisco, USA, 2018: 926-943

[43] Bowe S, Grigg J, Hopwood D. Recursive proof composition without a trusted setup. https://eprint.iacr.org/2019/1021

[44] Bünz B, Maller M, Mishra P, et al. Proofs for inner pairing products and applications//Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security. Singapore, 2021: 65-97

[45] Lee J. Dory: Efficient, transparent arguments for generalised inner products and polynomial commitments//Proceedings of the 19th Theory of Cryptography Conference. Raleigh, USA, 2021: 1-34

[46] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing//Proceedings of the 11th Annual International Cryptology Conference. Santa Barbara, USA, 1991: 129-140

[47] Groth J. Linear algebra with sub-linear zero-knowledge arguments//Proceedings of the 29th Annual International Cryptology Conference. Santa Barbara, USA, 2009: 192-208

[48] Zhang Z, Zhou Z, Li W, et al. An optimized inner product argument with more application scenarios//Proceedings of the 23rd International Conference on Information and Communication Security. Chongqing, China, 2021: 341-357

[49] Zhou Z, Zhang Z, Tao H, et al. Efficient inner product arguments and their applications in range proofs. IET Information Security, 2023, 17(3): 485-504

[50] Kim S, Lee G, Lee H, et al. Leopard: Sublinear verifier inner product argument under discrete logarithm assumption. IEEE Transactions on Information Forensics and Security, 2023, 18: 5332-5344

[51] Lee H, Seo J H. TENET: Sublogarithmic proof and sublinear verifier inner product argument without a trusted setup//Proceedings of the 18th International Workshop on Security. Yokohama, Japan, 2023: 214-234

[52] Groth J. Efficient zero-knowledge arguments from two-tiered homomorphic commitments//Proceedings of the 17th International Conference on the Theory and Application of Cryptology and Information Security. Seoul, Republic of Korea, 2011: 431-448

[53] Seo J H. Round-efficient sub-linear zero-knowledge arguments for linear algebra//Proceedings of the 14th IACR International Conference on Practice and Theory of Public-Key Cryptography. Taormina, Italy, 2011: 387-402

[54] The Ethereum Foundation Cryptography Research Team. Curdleproofs: A shuffle argument protocol. https://github.com/asn-d6/curdleproofs/blob/main/doc/curdleproofs.pdf

[55] Abe M, Fuchsbauer G, Groth J, et al. Structure-preserving signatures and commitments to group elements. Journal of Cryptology, 2016, 29: 363-421

[56] Gailly N, Maller M, Nitulescu A. Snarkpack: Practical snark aggregation//Proceedings of the 26th International Conference on Financial Cryptography and Data Security. Grenada, 2022: 203-229

[57] Lai R W F, Malavolta G, Ronge V. Succinct arguments for bilinear group arithmetic: Practical structure-preserving cryptography//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, UK, 2019: 2057-2074

[58] Groth J. On the size of pairing-based non-interactive arguments //Proceedings of the 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Vienna, Austria, 2016: 305-326

[59] Setty S, Lee J. Quarks: Quadruple-efficient transparent zkSNARKs. https://eprint.iacr.org/2020/1275

[60] Ambrona M, Beunardeau M, Schmitt A L, et al. aPlonK: Aggregated PlonK from multi-polynomial commitment schemes //Proceedings of the 18th International Workshop on Security. Yokohama, Japan, 2023: 195-213

[61] Gabizon A, Williamson Z J, Ciobotaru O. Plonk: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. https://eprint.iacr.org/2019/953

[62] Kuchta V, Sahu R A, Sharma G. Lattice-based inner product argument//Proceedings of the 13th International Conference on Cryptology in Africa. Fes, Morocco, 2022: 236-268

[63] Baum C, Damgård I, Lyubashevsky V, et al. More efficient commitments from structured lattice assumptions//Proceedings of the 11th International Conference on Security and Cryptography for Networks. Amalfi, Italy, 2018: 368-385

[64] Zhang J, Xie T, Zhang Y, et al. Transparent polynomial delegation and its applications to zero knowledge proof//Proceedings of the 2020 IEEE Symposium on Security and Privacy. San Francisco, USA, 2020: 859-876

[65] Bhadauria R, Fang Z, Hazay C, et al. Ligero++: A new optimized sublinear IOP//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. Virtual Event, USA, 2020: 2025-2038

[66] Zhang Z, Li W, Liu X, et al. Ligerolight: Optimized IOP-based Zero-Knowledge Argument for Blockchain Scalability. IEEE Transactions on Dependable and Secure Computing, 2023, Early Access. doi: 10.1109/TDSC.2023.3336717

[67] Bünz B, Fisch B, Szepieniec A. Transparent SNARKs from DARK compilers//Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 677-706

[68] Das S, Camacho P, Xiang Z, et al. Threshold signatures from inner product argument: Succinct, weighted, and multi-threshold//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. Copenhagen, Denmark, 2023: 356-370

[69] Attema T, Cramer R, Fehr S. Compressing proofs of k-out-of-n partial knowledge//Proceedings of the 41st Annual International Cryptology Conference. Virtual Event, 2021: 65-91

[70] Attema T, Cramer R, Rambaud M. Compressed Σ-protocols for bilinear group arithmetic circuits and application to logarithmic transparent threshold signatures//Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security. Singapore, 2021: 526-556

[71] Attema T, Cramer R, Kohl L. A compressed Σ-protocol theory for lattices//Proceedings of the 41st Annual International Cryptology Conference. Virtual Event, 2021: 549-579

[72] Attema T, Cascudo I, Cramer R, et al. Vector commitments over rings and compressed σ-protocols//Proceedings of the 20th Theory of Cryptography Conference. Chicago, USA, 2022: 173-202

[73] Arora S, Safra S. Probabilistic checking of proofs: A new characterization of NP. Journal of the ACM, 1998, 45(1): 70-122

[74] Ben-Sasson E, Chiesa A, Gabizon A, et al. Interactive oracle proofs with constant rate and query complexity//Proceedings of the 44th International Colloquium on Automata, Languages, and Programming. Warsaw, Poland, 2017: 40:1-40:15

[75] Bootle J, Chiesa A, Liu S. Zero-knowledge IOPs with linear-time prover and polylogarithmic-time verifier//Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Trondheim, Norway, 2022: 275-304

[76] Valiant P. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency//Proceedings of the 5th Theory of Cryptography Conference. New York, USA, 2008: 1-18

[77] Chiesa A, Tromer E. Proof-carrying data and hearsay arguments from signature cards//Proceedings of the 1st Innovations in Theoretical Computer Science. Beijing, China, 2010: 310-331

[78] Barak B, Goldreich O. Universal arguments and their applications. SIAM Journal on Computing, 2008, 38(5): 1661-1694

[79] Chong S, Tromer E, Vaughan J A. Enforcing language semantics using proof-carrying data. https://eprint.iacr.org/2013/513

[80] Chiesa A, Tromer E, Virza M. Cluster computing in zero knowledge//Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Sofia, Bulgaria, 2015: 371-403

[81] Naveh A, Tromer E. Photoproof: Cryptographic image authentication for any set of permissible transformations//Proceedings of the 2016 IEEE Symposium on Security and Privacy. San Jose, USA, 2016: 255-271

[82] Tyagi N, Fisch B, Zitek A, et al. VeRSA: Verifiable registries with efficient client audits from RSA authenticated dictionaries//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles, USA, 2022: 2793-2807

[83] Bonneau J, Meckler I, Rao V, et al. Coda: Decentralized cryptocurrency at scale. https://eprint.iacr.org/2020/352

[84] Chen W, Chiesa A, Dauterman E, et al. Reducing participation costs via incremental verification for ledger systems. https://eprint.iacr.org/2020/1522

[85] Kattis A, Bonneau J. Proof of necessary work: Succinct state verification with fairness guarantees. https://eprint.iacr.org/2020/190

[86] Ben-Sasson E, Chiesa A, Tromer E, et al. Scalable zero knowledge via cycles of elliptic curves//Proceedings of the 34th Annual International Cryptology Conference. Santa Barbara, USA, 2014: 276-294

[87] Chiesa A, Ojha D, Spooner N. Fractal: Post-quantum and transparent recursive proofs from holography//Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 769-793

[88] Bünz B, Chiesa A, Mishra P, et al. Recursive proof composition from accumulation schemes//Proceedings of the 18th Theory of Cryptography Conference. Durham, USA, 2020: 1-18

[89] Bünz B, Chiesa A, Lin W, et al. Proof-carrying data without succinct arguments//Proceedings of the 41st Annual International Cryptology Conference. Virtual Event, 2021: 681-710

[90] Kothapalli A, Setty S, Tzialla I. Nova: Recursive zero-knowledge arguments from folding schemes//Proceedings of the 42nd Annual International Cryptology Conference. Santa Barbara, USA, 2022: 359-388

[91] Kothapalli A, Setty S. SuperNova: Proving universal machine executions without universal circuits. https://eprint.iacr.org/2022/1758

[92] Kothapalli A, Setty S. HyperNova: Recursive arguments for customizable constraint systems. https://eprint.iacr.org/2023/573

[93] Bünz B, Chen B. ProtoStar: Generic efficient accumulation/folding for special sound protocols//Proceedings of the 29th International Conference on the Theory and Application of Cryptology and Information Security. Guangzhou, China, 2023: 77-110

[94] Zhou Z, Zhang Z, Dong J. Proof-carrying data from multi-folding schemes. https://eprint.iacr.org/2023/1282

[95] Zheng T, Gao S, Guo Y, et al. KiloNova: Non-uniform PCD with zero-knowledge property from generic folding schemes. https://eprint.iacr.org/2023/1579

[96] Ben-Sasson E, Chiesa A, Tromer E, et al. Succinct non-interactive zero knowledge for a von neumann architecture//Proceedings of the 23rd USENIX Security Symposium. San Diego, USA, 2014: 781-796

[97] Maller M, Bowe S, Kohlweiss M, et al. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, UK, 2019: 2111-2128

[98] Boneh D, Drake J, Fisch B, et al. Halo infinite: Proof-carrying data from additive polynomial commitments//Proceedings of the 41st Annual International Cryptology Conference. Virtual Event, 2021: 649-680

[99] Eagen L, Gabizon A. ProtoGalaxy: Efficient ProtoStar-style folding of multiple instances. https://eprint.iacr.org/2023/1106

[100] Bünz B, Chen J. Proofs for deep thought: Accumulation for large memories and deterministic computations. https://eprint.iacr.org/2024/325

[101] Goldwasser S, Kalai Y T, Rothblum G N. Delegating computation: Interactive proofs for muggles. Journal of the ACM, 2015, 62(4): 1-64

[102] Bünz B, Mishra P, Nguyen W, et al. Accumulation without homomorphism. https://eprint.iacr.org/2024/474

[103] Mohnblatt N. Sangria: A folding scheme for PLONK. https://github.com/geometryresearch/technical_notes/blob/main/sangria_folding_plonk.pdf

[104] Liu X, Gao S, Zheng T, et al. SnarkFold: Efficient SNARK proof aggregation from split incrementally verifiable computation. https://eprint.iacr.org/2023/1946

[105] Boneh D, Chen B. LatticeFold: A lattice-based folding scheme and its applications to succinct proof systems. https://eprint.iacr.org/2024/257

[106] Nguyen W, Datta T, Chen B, et al. Mangrove: A scalable framework for folding-based SNARKs. https://eprint.iacr.org/2024/416

[107] Silverman J H, Stange K E. Amicable pairs and aliquot cycles for elliptic curves. Experimental Mathematics, 2011, 20(3): 329-357

[108] Chiesa A, Chua L, Weidner M. On cycles of pairing-friendly elliptic curves. SIAM Journal on Applied Algebra and Geometry, 2019, 3(2): 175-192

[109] El Housni Y, Guillevic A. Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition//Proceedings of the 19th International Conference on Cryptology and Network Security. Vienna, Austria, 2020: 259-279

[110] El Housni Y, Guillevic A. Families of SNARK-friendly 2-chains of elliptic curves//Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Trondheim, Norway, 2022: 367-396

[111] Bellés-Muñoz M, Jiménez Urroz J, Silva J. Revisiting cycles of pairing-friendly elliptic curves//Proceedings of the 43rd Annual International Cryptology Conference. Santa Barbara, USA, 2023: 3-37

[112] Kothapalli A, Setty S. CycleFold: Folding-scheme-based recursive arguments over a cycle of elliptic curves. https://eprint.iacr.org/2023/1192

[113] Naor M, Paneth O, Rothblum G N. Incrementally verifiable computation via incremental PCPs//Proceedings of the 17th Theory of Cryptography Conference. Nuremberg, Germany, 2019: 552-576

[114] Paneth O, Pass R. Incrementally verifiable computation via rate-1 batch arguments//Proceedings of the 63rd Annual Symposium on Foundations of Computer Science. Denver, USA, 2022: 1045-1056

[115] Chen M, Chiesa A, Spooner N. On succinct non-interactive arguments in relativized worlds//Proceedings of the 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Trondheim, Norway, 2022: 336-366

[116] Chen M, Chiesa A, Gur T, et al. Proof-carrying data from arithmetized random oracles//Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques. Lyon, France, 2023: 379-404

[117] Chiesa A, Guan Z, Samocha S, et al. Security bounds for proof-carrying data from straightline extractors. https://eprint.iacr.org/2023/1646

[118] Hall-Andersen M, Nielsen J B. On Valiant's conjecture: Impossibility of incrementally verifiable computation from random oracles//Proceedings of the 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques. Lyon, France, 2023: 438-469

[119] Setty S. Spartan: Efficient and general-purpose zkSNARKs without trusted setup//Proceedings of the 40th Annual International Cryptology Conference. Santa Barbara, USA, 2020: 704-737

[120] Xie T, Zhang Y, Song D. Orion: Zero knowledge proof with linear prover time//Proceedings of the 42nd Annual International Cryptology Conference. Santa Barbara, USA, 2022: 299-328

[121] Golovnev A, Lee J, Setty S, et al. Brakedown: Linear-time and field-agnostic SNARKs for R1CS//Proceedings of the 43rd Annual International Cryptology Conference. Santa Barbara, USA, 2023: 193-226

[122] Parno B, Howell J, Gentry C, et al. Pinocchio: Nearly practical verifiable computation//Proceedings of the 2013 IEEE Symposium on Security and Privacy. Berkeley, USA, 2013: 238-252

[123] Belling A, Soleimanian A, Bégassat O. Recursion over public-coin interactive proof systems; faster hash verification//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. Copenhagen, Denmark, 2023: 1422-1436

[124] Albrecht M, Grassi L, Rechberger C, et al. MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity//Proceedings of the 22nd International Conference on the Theory and Application of Cryptology and Information Security. Hanoi, Vietnam, 2016: 191-219

[125] Campanelli M, Gailly N, Gennaro R, et al. Testudo: Linear time prover SNARKs with constant size proofs and square root size universal setup//Proceedings of the 8th International Conference on Cryptology and Information Security in Latin America. Quito, Ecuador, 2023: 331-351

[126] Bootle J, Cerulli A, Ghadafi E, et al. Linear-time zero-knowledge proofs for arithmetic circuit satisfiability//Proceedings of the 23rd International Conference on the Theory and Applications of Cryptology and Information Security. Hong Kong, China, 2017: 336-365

[127] Bootle J, Chiesa A, Groth J. Linear-time arguments with sublinear verification from tensor codes//Proceedings of the 18th Theory of Cryptography Conference. Durham, USA, 2020: 19-46

[128] Agrawal S, Ganesh C, Mohassel P. Non-interactive zero-knowledge proofs for composite statements//Proceedings of the 38th Annual International Cryptology Conference. Santa Barbara, USA, 2018: 643-673

[129] Benarroch D, Campanelli M, Fiore D, et al. Proposal: Commit-and-prove zero-knowledge proof systems and extensions. https://docs. zkproof. org/pages/standards/accepted-workshop4/proposal-commit. pdf

[130] Kilian J. Uses of Randomness in Algorithms and Protocols [Ph. D. dissertation]. Massachusetts Institute of Technology, USA, 1989

[131] Canetti R, Lindell Y, Ostrovsky R, et al. Universally composable two-party and multi-party secure computation//Proceedings of the 34th Annual ACM Symposium on Theory of Computing. Montréal, Canada, 2002: 494-503

[132] Campanelli M, Fiore D, Querol A. LegoSNARK: Modular design and composition of succinct zero-knowledge proofs//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, UK, 2019: 2075-2092

[133] Delignat-Lavaud A, Fournet C, Kohlweiss M, et al. Cinderella: Turning shabby X. 509 certificates into elegant anonymous credentials with the magic of verifiable computation//Proceedings of the 2016 IEEE Symposium on Security and Privacy. San Jose, USA, 2016: 235-254

[134] Lee J, Choi J, Kim J, et al. SAVER: SNARK-friendly, additively-homomorphic, and verifiable encryption and decryption with rerandomization. https://eprint. iacr. org/2019/1270

[135] Costello C, Fournet C, Howell J, et al. Geppetto: Versatile verifiable computation//Proceedings of the 2015 IEEE Symposium on Security and Privacy. San Jose, USA, 2015: 253-270

[136] Lipmaa H. Prover-efficient commit-and-prove zero-knowledge SNARKs//Proceedings of the 8th International Conference on Cryptology in Africa. Fes, Morocco, 2016: 185-206

[137] Benarroch D, Campanelli M, Fiore D, et al. Zero-knowledge proofs for set membership: Efficient, succinct, modular. Designs, Codes and Cryptography, 2023, 91(11): 3457-3525

[138] Backes M, Hanzlik L, Herzberg A, et al. Efficient non-interactive zero-knowledge proofs in cross-domains without trusted setup//Proceedings of the 22nd IACR International Conference on Practice and Theory of Public-Key Cryptography. Beijing, China, 2019: 286-313

[139] Campanelli M, Faonio A, Fiore D, et al. Lunar: A toolbox for more efficient universal and updatable zkSNARKs and commit-and-prove extensions//Proceedings of the 27th International Conference on the Theory and Application of Cryptology and Information Security. Singapore, 2021: 3-33

[140] Aranha D F, Bennedsen E M, Campanelli M, et al. ECLIPSE: Enhanced compiling method for Pedersen-committed zkSNARK engines//Proceedings of the 25th IACR International Conference on Practice and Theory of Public-Key Cryptography. Virtual Event, 2022: 584-614

[141] Fiore D, Tucker I. Efficient zero-knowledge proofs on signed data with applications to verifiable computation on data streams//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles, USA, 2022: 1067-1080

[142] Jawurek M, Kerschbaum F, Orlandi C. Zero-knowledge using garbled circuits: How to prove non-algebraic statements efficiently//Proceedings of the 2013 ACM SIGSAC

Conference on Computer and Communications Security. Berlin, Germany, 2013: 955-966

[143] Giacomelli I, Madsen J, Orlandi C. ZKBoo: Faster zero-knowledge for Boolean circuits//Proceedings of the 25th USENIX Security Symposium. Austin, USA, 2016: 1069-1083

[144] Chase M, Derler D, Goldfeder S, et al. Post-quantum zero-knowledge and signatures from symmetric-key primitives//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 1825-1842

[145] Ames S, Hazay C, Ishai Y, et al. Ligero: Lightweight sublinear arguments without a trusted setup//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 2087-2104

[146] Chiesa A, Hu Y, Maller M, et al. Marlin: Preprocessing zkSNARKs with universal and updatable SRS//Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 738-768

ZHANG Zong-Yang, Ph. D., associate professor. His research interests include blockchain and cryptography.

ZHOU Zi-Bo, Ph. D. candidate. His research interests include zero-knowledge proofs and cryptography.

DENG Yi, Ph. D., professor. His research interests include zero-knowledge proofs and their applications in financial technology.

Background

With the continuous rapid growth of the digital economy, data circulation has become an important way to value data. While developing and utilizing data, how to enhance data security protection is a key issue faced by data circulation. Zero-knowledge proofs (ZKPs) are one of the key technologies for secure data circulation. As a fundamental cryptographic primitive, they allow one party in a communication to prove the validity of a certain statement to another party, without revealing any additional information. They are not only a critical component of many cryptographic protocols such as authentication, digital signature, and public-key encryption, but also provide important privacy protection and performance enhancement for anonymous transactions, smart contracts, machine learning, and many other practical applications.

Despite significant developments on the performance and functionality since their introduction, there are still issues like insufficient efficiency, poor scalability, and specialized functionality in practical applications. To effectively address these bottleneck issues, the techniques of recursion and composition in ZKPs have received extensive attention in recent years, which are the main focus of this paper.

So far, there has been many surveys on ZKPs, but none of them systematically review the techniques of recursion and composition. In this paper, we conduct a systematic and comprehensive survey on the recursive and composite techniques of ZKPs for the first time.

Firstly, regarding recursive zero-knowledge proofs for algebraic statements, we extensively survey inner product arguments, a kind of Σ-protocols proving that the inner product of two committed vectors equals a public scalar. Specifically, we compare and analyze inner product arguments based on the Pedersen commitment scheme in detail, from the perspectives of prover complexity, communication complexity, verifier complexity, etc.

Secondly, regarding recursive zero-knowledge proofs for non-algebraic statements, we thoroughly examine the research landscape of two predominant applications, i.e., incrementally verifiable computation schemes and circuit-based composition of proof systems. Specifically, we conduct an in-depth comparative analysis of the complexity, key technologies, and implementation skills of incrementally verifiable computation schemes.

Thirdly, regarding composite zero-knowledge proofs for composite statements, we provide a granular comparative analysis of commit-and-prove zero-knowledge proofs, a kind of ZKPs proving the basic and common statement "The openings of some algebraic commitments satisfy an arithmetic/Boolean circuit", from the perspectives of complexity, setup phase, key modules, etc.

Finally, we outline future research directions for recursive and composite techniques of ZKPs.

This work is supported in part by the National Key Research and Development Program of China (No. 2022YFB-2702702), the National Natural Science Foundation of China (Nos. 62372020, 61972017, 72031001, 61932019, 62372447), the Beijing Natural Science Foundation (Nos. M22038, L222050, M21033), the Fundamental Research Funds for the Central Universities (No. YWF-23-L-1032), the International Joint Doctoral Education Fund of Beihang University.