中国科学 : 信息科学　2023 年　第 53 卷　第 9 期: 1688–1733
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
评述

隐私保护计算密码技术研究进展与应用

霍炜[1], 郁昱[2*], 杨糠[3], 郑中翔[4], 李祥学[2*], 姚立[2], 谢杰[2]

1. 清华大学计算机科学与技术系, 北京 100084
2. 复旦大学计算机科学技术学院, 上海 200438
3. 密码科学技术全国重点实验室, 北京 100878
4. 中国传媒大学计算机与网络空间安全学院, 北京 100024
* 通信作者. E-mail: yyuu@sjtu.edu.cn, xxli@cs.ecnu.edu.cn

收稿日期: 2022–11–13; 修回日期: 2023–02–12; 接受日期: 2023–03–31; 网络出版日期: 2023–09–06

国家杰出青年科学基金 (批准号: 62125204)、国家重点研发计划 (批准号: 2020YFA0309705) 和国家自然科学基金 (批准号: 61971192, 62102037, 62202437, 62272040, 12271306) 资助项目

摘要 “云、大、物、移、智、链” 等新技术的发展伴生了诸多安全问题特别是隐私泄露问题. 密码学为这些问题的解决提供了独特视角和可行路线. 这些新技术也促进了密码学研究的深入发展, 许多新型密码原语、功能强大的高等密码算法与协议新型构造不断涌现. 本文给出了具有隐私保护计算能力的几类高等密码算法与协议的研究进展综述, 特别是安全多方计算、同态加密、零知识证明、以及不可区分混淆四类算法与协议的设计和分析研究进展, 也通过具体示例讨论了它们的潜在应用场景. 本综述既着眼于各类算法与协议的不同层级安全属性, 也侧重于从模块化角度剖析具体构造的内在技巧逻辑甚至缺陷. 本文有助于读者掌握这些高等密码算法与协议的最新理论和技术进展、背后的发展逻辑, 并深悟其中的关键技术原理, 在密码学理论和实践的结合过程中得到有益启发.

关键词 隐私保护, 安全多方计算, 同态加密, 零知识证明, 不可区分混淆

1 引言

传统密码算法与协议广泛应用于各类互联网络及信息系统. 这些密码技术包括加密、签名、哈希、消息认证码、身份认证、密钥协商等, 能提供数据在传输、存储和访问等环节的保密性和完整性. 经过数十年发展, 众多密码算法与协议已形成了国际和国内标准, 为市场化应用提供了便利.

随着人工智能、区块链、隐私计算等新技术的发展和新应用的出现, 数据从传统的 “静态” 保护模式逐步向 “动态” 保护模式过渡, 比如, 在多个参与方动态计算数据的场景下保护各自数据的隐私性. 数据安全需求也呈现出跨学科融合趋势, 比如人工智能模型训练和预测过程中数据的隐私性和完整性需求、大规模分布式架构中的数据隐私保护以及可扩展性需求. 传统密码技术中的简单组件很难完全

引用格式: 霍炜, 郁昱, 杨糠, 等. 隐私保护计算密码技术研究进展与应用. 中国科学: 信息科学, 2023, 53: 1688–1733, doi: 10.1360/SSI-2022-0434
Huo W, Yu Y, Yang K, et al. Privacy-preserving cryptographic algorithms and protocols: a survey on designs and applications (in Chinese). Sci Sin Inform, 2023, 53: 1688–1733, doi: 10.1360/SSI-2022-0434

© 2023《中国科学》杂志社

满足越来越复杂多变的安全需求. 因此, 学术界和工业界逐渐给予高等密码算法的研究设计以及它们与不同学科场景的融合问题更多关注.

隐私计算的概念最早由 Li 等 [1] 提出, 是保护多个参与方在联合计算过程中各自隐私数据技术的统称. 这类技术主要面向数据驱动的行业和应用场景, 研究个人和商业敏感信息的隐私保护, 实现数据价值的安全高效传递. 从技术构成来看, 隐私计算属于跨学科信息技术, 以密码学为核心理论, 结合了大数据、人工智能、数据库、区块链等多领域知识. 广义的隐私计算除了密码学手段之外, 还包括联邦学习 [2,3]、差分隐私 [4~6]、可信执行环境 [7,8] 等技术. 即便如此, 高等密码算法和协议仍然是实现包括联邦学习在内的隐私保护计算的必要方法. 例如联邦学习常采用单同态加密、秘密分享等密码算法和协议保证各参与方的输入数据获得隐私保护, 允许各方实现基于梯度迭代的分布式机器学习.

数据隐私安全问题不仅涉及个人信息暴露, 更重要的是隐私暴露后会干预用户决策或使决策向隐私掌控方更偏好的方式倾斜. 数字信息经过人工智能算法等互联网技术放大之后更容易对受众产生影响, 形成 “信息茧房”. 合规使用隐私数据、完善算法推荐机制等有助于避免信息茧房的形成.

另一方面, 数据的合规共享仍存在不小困难, 利益、政策因素导致不同企业或机构之间没有动力甚至根本无法实现数据共享. 例如, 医疗机构存储的大量病患记录数据对于药厂来说具有极大价值, 但法律法规又要求病患记录不允许售卖给第三方. 再如, 不同国家的数据监管政策使得即使在不同国家拥有各自执行办公室的跨国企业, 其内部数据仍可能无法实现跨境传输. “数据孤岛” 现象导致现代商业数据无法合规流通, 制约了数字经济可持续发展.

从监管合规角度来看, 国内外对个人数据隐私的重视和保护日益严格. 2018 年 5 月生效的 GDPR [9] 被称为欧盟 “史上最严” 条例. Google 和 Facebook 在 GDPR 生效日分别收到了欧盟 39 亿欧元、37 亿欧元罚款诉讼, 产生巨大影响. 隐私监管机构还向苹果、亚马逊、LinkedIn 等公司提起诉讼. 诸多国内外知名企业不得不向欧洲用户更新隐私政策, 有的甚至关停欧洲服务器.

我国于 2017 年施行的《中华人民共和国网络安全法》强调了对基础设施及个人信息的保护. 2018 年实施的《信息安全技术个人信息安全规范》从国家标准层面明确了企业收集、使用、分享个人信息的合规要求. 2019 年发布的《数据安全管理办法》确立了数据分级分类管理以及风险评估、检测预警、应急处置等数据安全管理各项基本制度, 明确了开展数据活动的组织、个人的数据安全保护义务, 落实数据安全保护责任. 2020 年生效的《中华人民共和国密码法》从国家法律层面指出了密码技术的重要性, 明确了密码是国之重器, 是国家重要的战略资源, 直接关系国家政治安全、经济安全、国防安全和信息安全.

本文就安全多方计算、(全) 同态加密、零知识证明和程序混淆等典型的隐私保护计算技术的理论可行性和实际可用性等方面进行具体阐述. 就隐私保护计算领域而言, 安全多方计算、(全) 同态加密和零知识证明是应用最广也最具有代表性的高等密码算法和协议, 除此之外多为非密码技术, 如联邦学习、可信执行环境、数据脱敏等. 而程序混淆作为底层桥梁, 可以用于对上述 3 类技术的构造, 在推动这些高等密码算法和协议发展方面有不可忽视的作用. 具体而言, 安全多方计算作为隐私保护计算领域应用最广的密码技术, 广泛适用于多方之间的数据共享和计算, 如隐私保护下的各类通用运算、隐私集合求交、隐私信息检索, 以及隐私 AI 等应用. 同时安全多方计算本身包含了混淆电路、秘密分享、不经意传输等多种密码原语, 这些均能作为独立技术灵活运用于特定的场景, 同时也能与其他密码技术组合使用, 前者如采用不经意传输构造高效的隐私集合求交方案, 以实现高带宽模式下两方批量数据的标识符匹配, 后者如结合秘密分享和零知识证明用于构造满足恶意模型安全的门限签名方案, 从而增强密钥隐私保护. (全) 同态加密支持密文层面的运算, 其 “数据加密 – 密文计算 – 结果解密” 的模式, 在构建各类隐私保护应用场景方面具有直观优势. (全) 同态加密通常作为核心组件被用

于构造各类隐私保护计算技术和应用场景, 例如基于全同态加密的隐私集合运算方案可以支持两方数据体量相差较大 (非均衡模式) 时的数据匹配, 基于全同态加密的隐私信息检索方案支持单服务器模式下低通信量的数据查询, 以及各类联邦学习协议中经常采用 Paillier 加法同态加密算法保护训练过程中的梯度等中间参数不被泄露. 零知识证明可以广泛支持各类隐私安全的认证协议, 实现身份或业务数据的认证审核的同时遵循 “数据最小化” 原则不披露关键信息, 在各类隐私保护计算场景, 尤其是区块链方面获得了极大的应用, 出现了基于隐私保护认证或者利用其非交互性和简洁性的各类业务场景, 如区块链交易隐私保护、分布式身份、基于零知识证明的交易聚合验证, 基于零知识证明的区块链虚拟机等, 在近年来受到工业界的热切关注和实践优化. 程序混淆是继全同态加密之后的下一个密码学领域亟待攻克的关键技术, 其本身的强大之处在于能保证算法程序中隐藏了某些秘密信息, 而且即使拥有这个程序并能任意运行该程序的人也无法获得这些秘密信息. 这种允许在恶意条件下完成计算并隐藏秘密的安全优势, 使得程序混淆可以作为底层核心原语, 用于构造包含各类隐私保护计算技术在内的其他所有密码技术, 在隐私保护计算领域亦属于 “前沿” 技术, 值得长期研究探索.

本文就隐私保护计算领域中的典型密码技术 (特别是安全多方计算、全同态加密、零知识证明、不可区分混淆) 的设计以及研究进展进行了详细阐述, 也通过具体案例分析了它们的潜在应用场景. 从技术迭代演进的角度, 本文亦全面分析了这些密码算法和协议构造的内在技巧逻辑、不同层级的安全性, 以及不同方案之间的优劣差异, 作为隐私保护计算领域涉及的多类高等密码技术的全面剖析, 尚属首例. 希望本文对研究者和工程技术人员等各类读者有所收益, 在后续推动密码学理论和实践, 促进隐私保护计算行业发展的过程中, 获得借鉴和帮助.

2 应用场景

隐私计算技术可为数据驱动相关各行业提供隐私保护方法支撑. 下面我们根据具体计算类型介绍几类主要应用场景. 之后的几个章节将会具体讨论当前隐私保护计算领域各类密码技术的研究现状.

通用运算. 在实际应用中, 一个很自然的问题就是能否对各类通用运算提供隐私保护? 亦即, 在隐藏运算数 (operand) 甚至操作符 (operator) 的条件下, 多方协同计算仍可输出运算结果. 通用运算包括基本的加法和乘法等算术运算、各种比较类型的逻辑运算, 以及由这些基础运算 (通常也称为 “基础算子”) 组合而成的包含算术和逻辑类的衍生算子. 这些均为企业和机构在实际应用中的高频运算. 以金融行业常见的 “合格投资人” 为例, 银行 A 希望知道其客户 U 在银行 A、保险公司 B、证券公司 C 的金融资产总额是否超过一个阈值, 以便决定是否为该客户提供金融服务. 从隐私保护的角度来看, 银行 A 无法获得 U 在保险公司 B 和证券公司 C 侧的资产数据, 而保险 B 和证券公司 C 也无法获得 U 在 A 侧的资产数据以及要比较的阈值. 此时, A, B, C 三方可以通过安全的方式执行求和计算, 最后由 A 获得比较后的结果.

实际应用业务常常需要同时使用通用运算和其他类型的计算. 以上述的 “合格投资人” 为例, 在具体金融风控业务中, 机构需要对其自身的批量客户向另一方机构进行查询计算, 通过执行隐私集合求交操作安全地筛选出被查询方的命中用户群体, 在此基础上再进行其他计算.

通用运算的隐私保护主要依赖密码技术, 尤其是安全多方计算和 (全) 同态加密.

隐私集合运算. 隐私集合运算主要是指拥有各自集合数据的参与方在不泄露各自数据的前提下共同完成集合类运算. 常见的隐私集合运算主要是指隐私保护下的集合间交集运算. 隐私求交集 (private set intersection, PSI) 运算属于安全多方计算领域的特定应用问题, 不仅具有重要的理论意义, 也具有很强的应用价值. PSI 涉及到两个参与方, 即发送方 (sender) 和接收方 (receiver). 双方分别拥有标识

(比如身份 id) 集合 A 和 B, 并希望得到两者集合的交集 $A \cap B$, 除此之外不会得到其他信息 (在具体应用中, 也可仅由接收方得到交集内容). 有些业务场景中数据集合隐私保护是必须满足的要求. 例如, 当集合是金融监管用户名单或是关联基因数据的群体数据集时, 这样的集合作为计算输入就需要通过密码学的手段加以保护. 在多方协同的机器学习场景里, 各方在建模前需要先对各自的样本进行预处理, 特别是对于异构的数据源 (如金融数据和运营商数据) 来说, 预处理阶段需要执行样本对齐, 即选出各方相同的用户群体 (同一个用户从属于不同的机构) 后再进行下一步操作.

隐私信息检索. 隐私信息检索 (private information retrieval, PIR) 是指用户进行数据库检索时, 数据库在无法获得用户具体检索信息的前提下返回正确的查询结果, 从而保护用户查询隐私的一类计算问题. 例如, 病患想通过医药系统查询其疾病的治疗药物, 如果以该疾病名为查询条件, 医疗系统就会得知病人可能患有该类疾病, 泄露了病人隐私, 而隐私信息检索技术可以避免此类泄露问题. 又如在域名、专利申请过程中, 用户需向相关数据库提交自己申请的域名或专利信息以查询该申请是否已存在, 但是, 用户又不想让服务提供方知晓自己的申请名称以免被抢先注册. 在证券市场中, 用户既想查询某只股票信息, 又不能将自己感兴趣的股票泄露给服务方从而影响股票价格和自己的偏好. 以上这些需求都可以通过 PIR 技术实现. 现有的 PIR 协议主要包括满足信息论安全的隐私信息检索协议 (information-theoretic PIR) 和满足计算安全的隐私信息检索协议 (computational PIR). PIR 协议设计一般需要借助不经意传输、同态加密等高等密码算法和协议实现.

隐私 AI. 人工智能 (artificial intelligence, AI) 和机器学习 (machine learning) 技术广泛应用于数据相关的各个行业. 这些技术领域也涉及数据隐私问题. AI 计算一般分为训练和预测两个阶段. 训练阶段是通过对海量样本数据进行特征提取, 得到能够衡量或者描述样本信息的模型. 预测阶段使用该模型衡量和推测新样本的属性. 为使训练阶段得到更为精准的模型, 数据量越多越好, 数据越多样越好. 现实场景中, 数据往往分布于不同机构和企业, 而这些机构和企业往往不愿意分享持有的明文数据. 这是 AI 行业应用面临的发展困境之一. 隐私 AI 利用包括联邦学习和密码学中的安全多方计算、同态加密等在内的各类隐私计算技术, 可在保证各方数据隐私的前提下联合训练整体上虚拟融合的数据集, 训练完成后各方可利用各自得到的模型与其他方进行联合预测. 这既保证了模型隐私、又保证了数据隐私. 因此, 隐私 AI 技术可使各方数据在隐私保护的前提下得到充分利用和流通, 既保护企业和机构的核心数据资产、又能提升模型精度, 进而降低业务风险和成本.

隐私保护认证. 认证是指根据声明者或者声明对象所特有的特征信息, 确认声明者的身份或者声明对象的有效性, 比如用户身份认证、区块链交易认证等. 传统的身份认证模式往往会过多泄露认证主体的隐私信息. 例如, 在企业资质证明场景里, 很多时候仅需证明企业特定财务情况满足一定下限即可, 但传统信息披露方式却不得不要求企业披露完整的财务数据, 带来了不必要的隐私泄露风险. 在基于区块链的分布式交易系统里, 往往需要既隐藏包括交易双方身份、金额等在内的交易细节信息, 又保证交易逻辑的有效性, 但传统的公开透明和不可篡改的区块链和分布式账本难以满足这样的要求. 隐私保护认证以隐私数据最小化和选择性披露为目标, 通常需要借助零知识证明、盲签名或环签名等多种信息隐藏技术来实现.

隐私智能合约. 以超级账本 [10] 为代表的联盟链和以以太坊 [11] 为代表的公有链都属于可编程区块链, 但它们目前均缺乏隐私保护的能力. 特别地, 基于智能合约的事务处理无法提供对输入数据、输出数据或代码的隐私保护, 这大大限制了该技术的广泛应用. 当前的隐私保护手段主要是基于零知识证明来实现隐私保护下的简单业务 (如交易) 处理. 理论上来说, 不可区分混淆技术可以实现更广泛的隐私保护应用. 例如, 以太坊智能合约记录了一个机构账户 (包括其私钥), 并设置了在达成某一个条件时自动允许某个节点对机构账户进行交易. 如果我们使用混淆器 (obfuscator) 将合约加以混淆, 则

该节点只能获得交易输出结果而无法获得合约内容和账户私钥等信息.

3 安全多方计算

安全多方计算 (secure multi-party computation, MPC) 可实现多个参与方 (也称实体) 协同计算关于秘密数据的任意函数, 除函数输出外不泄漏任何其他秘密信息. 学界在 20 世纪 80 年代提出了安全多方计算的概念并给出了多种关于任意函数 MPC 协议的可行性设计方法 [12~17], 这些设计方法构成了后续大部分 MPC 协议的基本框架. MPC 支持任意 (多项式时间) 函数计算, 具有许多实际应用, 包括: 隐私保护机器学习 [18~45]、联邦学习 [23, 46~49]、基因分析 [50~52]、数据库安全 [53] 等. 此外, MPC 技术也能用于设计非交互零知识证明 [54~63]、可延展的交互零知识证明 [64~71]、隐私集合交集 [72~79] 等. 受到众多应用驱动, 安全多方计算在最近十多年里已从理论研究向实际可用方向迅速发展, 有望得到高效的实际部署. 目前, 已经出现了一系列 MPC 协议库, 包括: ABY [80], ABY^3 [22], EMP-toolkit [81], FRESCO [82], JIFF [83], MP-SPDZ [84], MPyC [85], SCALE-MAMBA [86], Sharemind [87], TinyGable [88] 等.

MPC 协议需要满足隐私性 (privacy) 和正确性 (correctness) 两个基本安全属性. 隐私性保证协议除函数输出外没有泄漏任何秘密信息, 正确性保证输出是预先协商函数的计算结果 (而不是其他函数). 即使存在内部攻击者, 这两个安全性质仍需成立. 具体来说, n 个参与方 $P_1, \ldots, P_n$ 运行 MPC 协议计算输出 $f(x_1, \ldots, x_n)$, 其中 x_i 是参与方 P_i 的秘密输入, f 是预先协商的函数. 隐私性保证不诚实参与方只能获得输出 $f(x_1, \ldots, x_n)$, 而不会泄漏诚实参与方 P_i 的秘密输入 x_i 的任何信息; 正确性保证参与方获得的输出结果为 $f(x_1, \ldots, x_n)$, 而不是 $g(x_1, \ldots, x_n)$, 其中 $g \neq f$ 为任意其他函数.

目前, 主要有两种方法设计 MPC 协议, 一种为秘密分享 (secret sharing) 方法, 另一种为混淆电路 (garbled circuit) 方法, 其中第 1 种方法的基本思想来源于文献 [13~15], 而第 2 种方法的基本思想来源于文献 [12, 17]. 一般而言, 秘密分享方法具有较低的通信带宽, 但轮数复杂度为 $O(d)$, 适用于低延迟网络 (例如: 局域网), 其中 d 表示电路深度; 混淆电路方法的轮数复杂度为 $O(1)$, 但需要较高通信带宽, 适用于高延迟网络 (例如: 广域网). 在恶意敌手模型下, 一系列研究工作 [89~98] 也将秘密分享方法融合到基于混淆电路的常数轮 MPC 协议中以获得更高效率. 与已知 MPC 综述 [99, 100] 比较, 本文的安全多方计算综述主要从模块化协议设计角度剖析 MPC 协议的基本设计框架和方法, 主要考虑基于秘密分享方法的 MPC 协议关键设计方法与研究进展 (基于混淆电路方法的 MPC 协议设计方法与研究进展可参考综述论文 [101]). 本文中基于秘密分享方法的 MPC 综述建立在之前综述论文 [101] 基础之上, 进一步凝练了基于秘密分享的实际高效 MPC 协议基本框架, 更加模块化地阐述了该类 MPC 协议的设计方法, 也加入了更多最新研究成果等.

3.1 安全多方计算分类

从定义安全性的不同角度, 安全多方计算 (MPC) 具有多种不同的分类方式. 具体而言, 根据敌手 (adversary, 即攻击者) 能力, MPC 可有以下分类方式.

• **敌手行为.** 如果敌手是半诚实 (semi-honest) 的, 那么它必须按照规范说明运行协议, 但可以通过观察协议记录获取更多的信息. 恶意 (malicious) 的敌手能够运行任何攻击策略和发送任意消息来攻击协议. 半诚实敌手也称被动 (passive) 敌手, 恶意敌手也称主动 (active) 敌手.

• **腐化数量.** 令 t 表示腐化门限 (即不诚实参与方数量的上界), n 表示所有参与方的总数. 根据 t 与 n 的关系, 主要分为两种情况: (1) 不诚实大多数情况, 其中 $n/2 \leqslant t < n$ (大部分 MPC 协议考虑

$t = n - 1$); (2) 诚实大多数情况, 其中 $t < n/2$ (即允许严格少于一半数量的参与方被敌手腐化).

• **敌手计算资源.** 即使敌手拥有无限计算能力, MPC 协议仍可保证安全的话, 则称该协议为信息论安全的 (或无条件安全的). 信息论安全协议需要假设大多数实体是诚实的. 计算安全的 MPC 协议则假设敌手具有概率多项式时间 (probabilistic polynomial time, PPT) 计算能力. 计算安全的 MPC 协议可在不诚实大多数情况下实现.

• **腐化策略.** 如果敌手需要在协议运行前确定腐化实体的集合, 则称该敌手是静态的 (static). 自适应的 (adaptive) 敌手能在协议运行过程中确定腐化哪些实体. 目前, 高效的 MPC 协议设计主要考虑的是静态腐化, 而满足自适应腐化安全性的 MPC 协议通常效率较低.

根据敌手能否阻止诚实参与方获得函数输出, 安全多方计算的安全性可分为 3 个不同等级.

• **中止安全** (security with abort)**.** 不诚实参与方在获得函数输出后, 可以阻止诚实参与方获得输出. 为了协议的高效性, 安全多方计算研究者通常考虑该安全性质. 从输出可达性角度来说, 中止安全是在不诚实大多数情况下能达到的最强性质, 即计算函数时存在不公平性 [102].

• **公平性** (fairness)**.** 所有参与方要么都获得函数输出, 要么都不能获得函数输出. 该安全性质要求大部分参与方是诚实的.

• **保证输出传送** (guaranteed output delivery)**.** 所有参与方一定获得函数输出. 该安全性质在诚实大多数情况下成立.

从输出可达角度来看, 有中止安全 < 公平性 < 保证输出传送; 从 MPC 协议效率上看, 满足中止安全的 MPC 协议效率最高, 而保证输出传送的 MPC 协议效率最低. 此外, 安全多方计算主要考虑 3 种计算模型: 布尔 (Boolean) 电路、算术电路和 RAM 程序, 针对前两种计算模型的 MPC 协议效率显著优于针对最后一种计算模型的 MPC 协议. 因此, 本文仅考虑两种电路计算模型, 在未来的文章中介绍基于 RAM 程序的安全多方计算研究进展.

由上可见, 根据敌手的攻击能力和输出的可达性, MPC 协议可分为多种类型, 相应地满足不同的安全性和效率. 本文主要考虑针对静态敌手、满足中止安全性的高效 MPC 协议设计, 不涉及渐进复杂度低或轮数复杂度最优但实际效率低的 MPC 协议.

3.2 线性秘密分享方案

基于秘密分享方法的实际高效 MPC 协议主要采用 3 类线性秘密分享方案 (linear secret sharing scheme, LSSS): (1) 加法秘密分享 (additive secret sharing), (2) Shamir 秘密分享 [103], (3) 复制秘密分享 (replicated secret sharing) [104, 105]. 本文将打包秘密分享 (packed secret sharing) [106] 看作 Shamir 秘密分享的一般化变形. 3 类秘密分享方案主要定义在一个域 $\mathbb{F}$ 上, 也能扩展到一般环上 (如整数环 $\mathbb{Z}_{2^k}$, k 一般为 32 或 64). 加法秘密分享主要用于不诚实大多数 MPC 协议中, Shamir 秘密分享和复制秘密分享主要用于诚实大多数 MPC 协议中. 对于一个秘密 $x \in \mathbb{F}$, 通常用 $[x]$ 表示其分享. 对于 n 个参与方 $P_1, \ldots, P_n$ 和腐化门限 t (即不诚实参与方的最大数量), 3 类 LSSSs 的 $\mathsf{Share}(x)$ 过程定义如下.

• **加法秘密分享** ($t = n - 1$)**.** 随机选取 $x^1, \ldots, x^n \in \mathbb{F}$ 满足 $\sum_{i=1}^{n} x^i = x$, 发送 x^i 给参与方 P_i 作为它的分享.

• **Shamir 秘密分享** ($t < n/2$)**.** 随机选取 $x^1, \ldots, x^t \in \mathbb{F}$, 利用拉格朗日 (Lagrange) 插值构造 t 次多项式 $f(\cdot)$ 满足 $f(0) = x$ 和 $f(\alpha_i) = x^i$ ($i = 1, \ldots, t$), 发送 $x^i = f(\alpha_i)$ 给 P_i 作为它的分享, 其中 $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ 表示 n 个不同的非零元素. 对于打包秘密分享 ($t < n(1/2 - \epsilon)$, $0 < \epsilon < 1/2$), 可以一次性分享 k 个秘密 $x_1, \ldots, x_k$. 给定额外 k 个不同的元素 $\beta_1, \ldots, \beta_k$, 插值出 $d = t + k - 1$ 次多项式 $f(\cdot)$ 满足 $f(\beta_i) = x_i$ ($i = 1, \ldots, k$), 以及 $f(\alpha_i) = s^i$ ($i = 1, \ldots, t$), 其中 $s^1, \ldots, s^t \in \mathbb{F}$ 为随机选取的元素. 同

样地, $s^i = f(\alpha_i)$ 作为 P_i 的分享.

• **复制秘密分享** ($t < n/2$)**.** 复制秘密分享是建立在加法秘密分享的基础之上, 但允许每个参与方获得多份分享. 以 $n = 3$ 和 $t = 1$ 为例, 随机选取 $x^1, x^2, x^3 \in \mathbb{F}$ 满足 $x^1 + x^2 + x^3 = x$, 发送 (x^{i-1}, x^{i+1}) 给 P_i, 其中索引 $i-1$ 及 $i+1$ 的计算在 mod 3 下执行.

在上述 3 类秘密分享方案中, 可以利用伪随机生成器 (pseudo-random generator, PRG) 和随机的种子生成均匀随机的分享, 从而减少分享传输带来的通信开销. 与 Shamir 秘密分享相比, 复制秘密分享主要用在参与方数量 n 比较小的情况, 因为复制秘密分享方案要求每方存储 $\binom{n-1}{t}$ 个分享. 当 n 比较大时, 诚实大多数 MPC 协议主要采用 Shamir 秘密分享. 以上 3 类秘密分享方案都是线性的. 也就是说, 所有参与方均可以本地方式计算以下分享:

• 给定 $[x]$ 和 $[y]$, 本地计算 $[z] = [x] + [y]$ 满足 $z = x + y$.

• 给定 $[x]$ 和公开已知的常数 $c \in \mathbb{F}$, 本地计算 $[y] = [x] + c$ 满足 $y = x + c$.

• 给定 $[x]$ 和公开已知的常数 $c \in \mathbb{F}$, 本地计算 $[y] = c \cdot [x]$ 满足 $y = c \cdot x$.

3 类 LSSSs 都支持 Open 过程, 允许所有参与方获得相关的秘密, 即所有参与方运行 Open($[x]$) 后获得秘密 x. 在半诚实敌手模型下, 3 类 LSSSs 通过让每方泄漏各自的分享可实现 Open 过程. 在恶意敌手模型下, 加法秘密分享不能保证 Open 过程的正确性 (即存在恶意实体发送错误的分享使得打开的秘密是错误的). 这是因为加法秘密分享没有提供任何关于秘密的认证性. 为了达到恶意安全性, 目前最有效的方法是嵌入信息论消息认证码 (information-theoretic message authentication code, IT-MAC) 到加法秘密分享中, 从而实现对秘密值的认证. 在诚实大多数情况下, 没有必要嵌入 IT-MACs 到 Shamir 或复制秘密分享中, 因为两类秘密分享已经在大部分实体诚实的情况下保证了打开秘密的正确性 (即大部分诚实实体的分享已经确定了秘密值).

信息论消息认证码. 在不诚实大多数情况下, 主要有两类信息论消息认证码用在恶意安全的 MPC 协议中: 一类是 BDOZ (Bendlin-Damgard-Orlandi-Zakarias) 类 IT-MACs [107], 另一类是 SPDZ 类 IT-MACs [108]. 前者主要用在常数轮 MPC 协议中以认证混淆电路, 后者主要用在基于加法秘密分享的 MPC 协议 (例如: GMW (Goldreich-Micali-Wigderson) 协议 [13]) 中以认证分享的秘密值. 本文主要介绍 SPDZ 类信息论消息认证码, 关于 BDOZ 类信息论消息认证码构造见文献 [91,101,107,109]. SPDZ 类信息论消息认证码定义如下:

• 随机选取密钥 $\Delta \in \mathbb{F}$, 计算 $m = x \cdot \Delta \in \mathbb{F}$ 作为消息 $x \in \mathbb{F}$ 的消息认证码.

• 对于参与方 $P_1, \ldots, P_n$, 每个参与方 P_i 获得 (x, m, Δ) 的一个分享, 记作 (x^i, m^i, Δ^i) 满足

$$\sum_{i=1}^{n} x^i = x, \quad \sum_{i=1}^{n} m^i = m, \quad \sum_{i=1}^{n} \Delta^i = \Delta.$$

相关的 MPC 协议保证: 即使值 x 被打开和敌手腐化了 $n-1$ 个实体, 敌手也不能获得密钥 Δ 或消息认证码 m.

• 如果密钥 Δ 是信息论安全的, SPDZ 类消息认证码方案是安全的 (在信息论意义下抗存在性伪造攻击). 特别地, 如敌手伪造了一个不同消息 $x' = x + e_0$ 的消息认证码 $m' = m + e_1$, 则有 $m' = x' \cdot \Delta$, $m = x \cdot \Delta$, 其中 $e_0 \neq 0$ 且 e_1 为敌手引入的错误. 从而, $\Delta = e_1/e_0 \in \mathbb{F}$ 被敌手泄漏, 这与 "Δ 是信息论安全的" 矛盾. 泄漏 Δ 的概率至多为 $1/|\mathbb{F}|$, 这对于大域 $\mathbb{F}$ 来说是可忽略的.

以上信息论消息认证码也可将消息 x 定义在一个子域 $\mathbb{F}$ 上 (如: $\mathbb{F} = \mathbb{F}_2$), 将密钥 Δ 和消息认证码 m 定义在 $\mathbb{F}$ 的扩域 $\mathbb{K}$ 上, 从而实现小域上元素的认证. 这里 $\mathbb{F}$ 是小域, $\mathbb{K}$ 是大域. 通常用 $[\![x]\!]$ 表示装备了 SPDZ 类信息论消息认证码的加法秘密分享. 当值 x 被打开后, 为了检查打开值 x 的正确性,

每个参与方 P_i 发送 $\sigma^i = m^i - x \cdot \Delta^i$ 给其他参与方, 然后验证等式 $\sum_{i=1}^n \sigma^i = 0$ 是否成立. 为了抵抗合谋攻击, 每个参与方 P_i 需要先承诺 σ^i, 然后打开承诺, 再验证等式. 如果有多个值 $x_1, \ldots, x_\ell$ 被打开, 可以采用随机线性组合方法将 $[\![x_1]\!], \ldots, [\![x_\ell]\!]$ 组合后, 再执行以上验证过程, 使得验证通信复杂度为 $O(1)$ (即与 ℓ 无关), 其中细节参见文献 [108,110,111].

3.3 基于秘密分享的安全多方计算协议框架

安全多方计算首先将计算任务转换为电路 $\mathcal{C}$, 然后在保护数据隐私的要求下计算电路输出. 目前, 主要有两种电路模型: 布尔电路和算术电路, 可统一为以下 4 部分.

• 电路输入线集合 $\mathcal{I}$: 每条线对应一个秘密输入值 $x \in \mathbb{F}$.

• 加法门 (i, j, k, ADD), 其中 i, j 为门输入线, k 为门输出线: 加法操作定义在域 $\mathbb{F}$ 上, 满足 $z_k = z_i + z_j$, 其中 z_i, z_j, z_k 为相应的门输入值和输出值.

• 乘法门 (i, j, k, MULT): 乘法操作也定义在域 $\mathbb{F}$ 上, 满足 $z_k = z_i \cdot z_j$, 其中 z_i, z_j, z_k 为相应的门输入值和输出值.

• 电路输出线集合 $\mathcal{O}$: 每条线对应一个输出值 $y \in \mathbb{F}$.

如果上述过程中 $\mathbb{F} = \mathbb{F}_2$, 就得到一个布尔电路, 其中加法门即为 XOR 门, 乘法门即为 AND 门. 如果 $\mathbb{F} \neq \mathbb{F}_2$, 就得到一个算术电路. 以上电路定义也容易扩展到一般环上 (例如: $\mathbb{Z}_{2^k}$). 通过执行 MPC 协议, 参与方 $P_1, \ldots, P_n$ 能够安全计算电路 $\mathcal{C}$, 获得电路输出 $y = \mathcal{C}(x_1, \ldots, x_n)$, 其中 x_i 为参与方 P_i 的秘密输入. 简单起见, 假设所有参与方获得相同的输出 y.

基于 LSSS 的 MPC 协议主要使用了 80 年代协议的基本思想 [13~15], 协议框架如下.

(1) 处理输入. 对于参与方 P_i 的输入 x, P_i 运行 $\mathsf{Share}(x)$ 使得所有参与方获得分享 $[x]$.

(2) 电路计算. 根据预先定义的电路拓扑序, 对于每个门 (i, j, k, T), 其门输入分享为 $[z_i]$ 和 $[z_j]$, 所有参与方 $P_1, \ldots, P_n$ 执行下面的计算.

• 若 $T = \mathsf{ADD}$, 则本地计算 $[z_k] := [z_i] + [z_j]$. 根据秘密分享的线性性, 加法门无需通信.

• 若 $T = \mathsf{MULT}$, 则所有参与方运行乘法子协议 Π_{milt} 计算输出分享 $[z_k]$, 满足 $z_k = z_i \cdot z_j$.

(3) 输出重构. 对于每个输出门的输出值 y, 所有参与方运行 $\mathsf{Open}([y])$ 使得每个参与方获得输出值 y.

对于不诚实大多数情况下的恶意安全 MPC 协议 (如: SPDZ 类协议), 输入线的处理通常采取以下步骤: (1) 在预处理阶段, 所有参与方生成认证分享 $[\![r]\!]$, 使得输入拥有者 P_i 获得 r; (2) 对于在线阶段, 参与方 P_i 广播 $d = x - r$ 给所有参与方, 然后所有参与方计算 $[\![x]\!] := [\![r]\!] + d$.

3.4 诚实大多数安全多方计算

对于诚实大多数情况, 本小节首先介绍在半诚实敌手模型下 MPC 协议的设计方法与研究进展, 然后介绍在恶意敌手模型下安全的 MPC 协议.

半诚实安全. 许多诚实大多数假设下的 MPC 协议采用 Shamir 秘密分享作为底层的 LSSS. 对于输入分享 $[x]$ 和 $[y]$, 参与方 $P_1, \ldots, P_n$ 本地计算输出分享 $[z] = [x \cdot y]$, 其中每个参与方 P_i 计算 $h(\alpha_i) = f(\alpha_i) \cdot g(\alpha_i)$, 多项式 f, g, h 分别对应于 Shamir 分享 $[x]$, $[y]$, $[z]$. 输出分享 $[z]$ 对应的多项式 h 次数为 $2t$, 其中 $t < n/2$ 为腐化门限 (即不诚实实体的最大数量). 从而, 对于半诚实安全性, 乘法子协议 Π_{milt} 主要考虑如何降低输出 Shamir 秘密分享对应多项式的次数. 目前, 主要有两种高效的方法降低多项式次数.

• **GRR 方法 [112].** 每个参与方通过 t 次 Shamir 秘密分享将 $h(\alpha_i)=f(\alpha_i)\cdot g(\alpha_i)$ 分享给所有参与方, 然后将所有收到的分享通过拉格朗日插值常数线性组合到一起作为 $[z]$ 的分享.

• **DN 方法 [113].** (1) 所有参与方生成随机双分享 (random double sharings) $([r]_t,[r]_{2t})$, 其中 $[r]_t$ 表示随机数 r 的 t 次 Shamir 分享; (2) 所有参与方本地计算 $[z]_{2t}=[x\cdot y]_{2t}$, 发送 $[z]_{2t}+[r]_{2t}$ 的分享给一个指定的参与方 $P_{\texttt{king}}\in\{P_1,\ldots,P_n\}$; (3) $P_{\texttt{king}}$ 重构分享获得 $z+r$ 后, 用 t 次 Shamir 秘密分享将 $z+r$ 分享给所有参与方, 使得所有参与方获得 $[z+r]_t$; (4) 所有参与方计算 $[z]_t=[z+r]_t-[r]_t$.

对于深度为 d 的电路计算: GRR 方法需要 d 轮通信, 每方每个乘法门通信复杂度为 $O(n)$; DN 方法需要 $2d+1$ 轮通信, 每方每个乘法门通信复杂度为 $O(1)$, 更适合于参与方数量较多的情况. 为了保证信息论安全, 原始的 DN 协议 [113] 计算一次乘法需要 6 个域元素的通信量, 后来改进到了 5.5 个域元素 [114, 115]. 最近, 通过每次计算 n 个乘法门, 让每个参与方在一次乘法门操作中充当 $P_{\texttt{king}}$ 的角色, Goyal 等 [116] 将 DN 协议的通信开销降低到每个乘法门 4 个域元素. 如要保证计算安全性, 通过利用伪随机生成器 (PRG) 和伪随机秘密分享方法 [105], DN 类协议的单次乘法操作只需 2 个域元素的通信 [116], 其中的随机分享由 PRG 和随机种子计算得到故不需通信开销. 以上诚实大多数 MPC 协议主要考虑在域上的算术电路, 也有学者提出一般环 (如 $\mathbb{Z}_{2^k}$) 上算术电路的 MPC 协议, 比如最近的协议 [117~119]. 基于 Shamir 秘密分享的重生成性质 [120] 及其 Galois 环一般化, Abspoel 等 [121] 设计了诚实大多数 MPC 协议, 需要 $d+O(1)$ 轮通信, 每方每次乘法通信复杂度为 $O(n/\log n)$ 个环元素.

简单起见, 对于基于复制秘密分享的 MPC 协议, 本文主要介绍安全三方计算 (secure three-party computation, 3PC) 的基本设计方法. 半诚实安全的 3PC 协议 [122, 123] 仅需一轮通信和每方每次乘法发送一个域元素. 由于加法秘密分享被复制, 每个参与方能本地计算 $[z]=[x\cdot y]$ 的分享, 但本地计算后的分享不满足复制特性. 不过, 每个参与方可用随机的零分享对自己 $[z]$ 的分享进行随机化, 然后发送给其他实体, 使得每方获得的分享满足复制特性. 这里, 随机的零分享可通过预先建立的随机种子和 PRG 本地计算. 对于超过三方的情况, 基于复制秘密分享的 MPC 协议可参见文献 [124, 125] 等.

恶意安全. 在诚实大多数情况下, 已知的半诚实安全 MPC 协议 (如: DN 协议 [113]) 在恶意敌手模型下满足隐私性, 但敌手可在门输出中引入与门输入无关的加法错误从而导致协议不满足正确性. 考虑到加法门是参与方本地计算, 自然满足正确性, 所以我们只需检查乘法门计算的正确性. 目前, 有多种方法可检查乘法门计算的正确性, 从而将半诚实安全的 MPC 协议转化为恶意安全的 MPC 协议. 在 2017 年以前, 这些方法包括 [113, 126~129] 等以及针对基于复制秘密分享半诚实安全 3PC 协议 [122] 的 "Cut-and-Bucketing" 方法 [130, 131] 等.

下面主要介绍最近几年实际高效的乘法门计算正确性检查方法. 如果腐化门限 $t<n/3$, 基于 Shamir 秘密分享的 MPC 协议 [132] 给出了目前效率最优的检查方法, 能够批量检查每个乘法门输出分享 $[z]_t$ 与输入分享乘积 $[x\cdot y]_{2t}$ 之差为 0 的分享. 该方法对于更大的腐化门限 $t<n/2$ 不适用.

如果腐化门限 $t<n/2$, Lindell 和 Nof [123] 采取了随机 Beaver 三元组和随机化检查技术去验证乘法门计算的正确性, 但需要 7 倍于半诚实安全协议的通信开销. 随后, Chida 等 [133] 提出了一种新的乘法门正确性验证方法, 执行两次半诚实安全的乘法协议, 并用一个相关的乘法三元组检查乘法门计算的正确性. 这个方法只需要两倍于半诚实安全协议的通信开销. Nordholt 和 Veeningen [134] 提出的乘法门正确性检查方法也能达到这样的通信开销. 以上方法主要考虑大域的情况, 对于小域 (如: $\mathbb{F}_2$) 则需要多次重复执行检查过程, 这就带来明显更大的通信开销. 利用具有亚线性 (sublinear) 通信复杂度的分布式零知识证明技术 [125] 可设计通信效率最优的恶意安全诚实大多数 MPC 协议, 其乘法门验证协议针对算术电路和布尔电路均具有亚线性通信复杂度. 具体而言, Boneh 等 [125] 利用 DN 协议的复制秘密分享版本和分布式零知识证明设计了通信效率最优的 3PC 协议, 布尔电路每个 AND 门约

1 比特通信开销, 乘法门验证的通信开销为 $O(n\sqrt{|C|}+n)$ ($|C|$ 为乘法门数量, n 为参与方数量). 文献 [135] 随后改进了该 3PC 协议并达到了目前最好的效率, 安全性也从中止安全加强到保证输出传送 [135]. 对于参与方数量 $n>3$ 的情况, 有两个通信效率最优的诚实大多数 MPC 协议 [115,136], 针对恶意敌手的乘法门验证协议均达到对数通信复杂度. 特别地, Goyal 等 [115] 的方法满足信息论安全性, 轮数复杂度为 $O(\log_k|C|)$, 通信复杂度为 $O(n\log_k|C|+n)$, 其中 $k\geqslant 2$ 为压缩参数. Boyle 等 [136] 的方法具有计算安全性, 利用 Fiat-Shamir 转换获得了 $O(1)$ 的轮数复杂度, 通信复杂度则为 $O(n\log|C|+n)$. 两个方法均基于分布式零知识证明技术 [125], 采用递归思想执行乘法门的验证, 其中 Goyal 等 [115] 的方法利用了 "半诚实安全 DN 协议计算向量内积的通信开销为常数" 的重要观察, Boyle 等 [136] 的方法利用了 "秘密值 x 的分享也能看作每方份额 x^i 分享" 的重要观察. 以上基于 Shamir 秘密分享的诚实大多数 MPC 协议要求域的大小 $|\mathbb{F}|>n$ 以支持存在至少 $n+1$ 个不同的元素, 使得计算布尔电路的通信开销为每个乘法门 $O(n\log n)$ 比特. 最近, 基于逆向乘法友好嵌入工具 (reverse multiplication friendly embedding, RMFE) [137], Polychroniadou 和 Song [138] 结合 Shamir 秘密分享和加法秘密分享设计了每个乘法门通信开销 $O(n)$ 比特的 MPC 协议.

当参与方数量达到成百上千量级时 (例如: 上千个参与方联合在各自秘密数据集上安全训练一个机器学习模型), 诚实大多数 MPC 协议可以采用打包秘密分享 (packed secret sharing) 获得 $O(|C|)$ 的通信复杂度, 而基于 Shamir 秘密分享的诚实大多数 MPC 协议则需要 $O(n|C|)$ 的通信复杂度. 为获得这样的低通信复杂度, 基于打包秘密分享的 MPC 协议仅容忍更低的腐化门限, 即 $t<n(1/2-\epsilon)$ ($0<\epsilon<1/2$, 如 $t=(n-1)/3$). 对于 SIMD (single instruction multiple data) 电路 (即一个电路由多个相同的子电路组成), Franklin 和 Yung [106] 从可行性层面设计了诚实大多数 MPC 协议, 达到了 $O(|C|)$ 的通信复杂度. 最近, Beck 等 [139] 提出了实际高效、针对 SIMD 电路的诚实大多数 MPC 协议, 达到了 $O(|C|)$ 的通信和计算复杂度. 他们还利用 Chida 等 [133] 的乘法门验证技术设计了恶意安全的协议, 但通信开销是半诚实安全的协议的 $1.7\sim 3$ 倍.

对于单个一般电路 (即没有特殊的重复结构), 文献 [129, 140, 141] 提出了信息论安全的 MPC 协议, 通信复杂度为 $O(|C|\log|C|)$ 或 $O(|C|\log^{1+\delta}n)$, 其中 δ 是任意正常数. Goyal 等 [142] 设计了首个通信复杂度 $O(|C|)$ 的信息论安全 MPC 协议, 通过扩展基于分布式零知识证明的乘法门验证技术 [115] 获得了恶意安全性, 通信复杂度是半诚实安全协议的 $1+o(1)$ 倍. 另外, Gordon 等 [143] 利用 DN 类乘法协议计算随机的 SPDZ 类认证三元组, 然后选择任意 $t+1$ 个参与方执行在线协议, 利用随机的 SPDZ 类认证三元组检查乘法门计算的正确性, 达到了在线通信的高效性, 但整体通信开销仍然是半诚实安全协议的 2 倍以上.

3.5 不诚实大多数安全多方计算

本小节主要从模块化设计思路介绍不诚实大多数安全多方计算协议的设计方法及研究进展, 首先考虑半诚实安全模型, 再考虑恶意安全模型. 在不诚实大多数情况下的 MPC 协议主要通过不经意传输 (oblivious transfer, OT) 协议或不经意线性计算 (oblivious linear evaluation, OLE) 生成 Beaver 乘法三元组. 如果 $\mathbb{F}=\mathbb{F}_2$, 则使用 OT, 否则使用 OLE. 要达到恶意安全的话, 则需要使用相关不经意传输 (correlated OT, COT) 协议或向量不经意线性计算 (vector OLE, VOLE) 协议以计算信息论消息认证码, 从而实现认证分享的生成. 这里 COT 或 VOLE 的选择同样取决于是否有 $\mathbb{F}=\mathbb{F}_2$. 本文首先给出 OT 和 OLE 及其变形的定义, 然后基于 OT/OLE 概念描述不诚实大多数 MPC 协议的设计方法与研究进展, 最后介绍 OT 和 OLE 及其变形的构造方法.

• **不经意传输及其相关变形.** 作为基础密码组件, OT [144,145] 允许发送方 P_S 输入两个消息 m_0 和

m_1, 接收方 P_R 输入一个选择比特 $b \in \{0,1\}$, 使得 P_R 仅能获得 m_b 而对 m_{1-b} 完全未知, 同时 P_S 不能获得比特 b 的任何信息. 作为 OT 的一类重要变形, COT 要求输入消息 m_0 和 m_1 满足一个固定的相关性 Δ, 即 $m_0 \oplus m_1 = \Delta$. 从向量角度可以这样理解: COT 协议输出 $(\boldsymbol{v}, \Delta)$ 给 P_S, 输出 $(\boldsymbol{w}, \boldsymbol{u})$ 给 P_R, 满足 $\boldsymbol{w} = \boldsymbol{v} + \boldsymbol{u} \cdot \Delta \in \{0,1\}^k$ 和 $\boldsymbol{u} \in \{0,1\}$.

• **不经意线性计算及其向量变形.** 作为 OT 的算术一般化, OLE 允许两方 P_S 和 P_R 获得两个秘密值乘积的加法分享. 具体地, 通过运行 OLE 协议, P_S 获得 $(x, z_0) \in \mathbb{F}^2$, P_R 获得 $(y, z_1) \in \mathbb{F}^2$, 满足 $z_0 + z_1 = x \cdot y$, 其中 $\mathbb{F}$ 为一般的域. VOLE [146, 147] 作为 COT 的算术一般化, 允许 P_S 和 P_R 分别获得 $(\boldsymbol{v}, \Delta)$ 和 $(\boldsymbol{u}, \boldsymbol{w})$, 满足 $\boldsymbol{w} = \boldsymbol{v} + \boldsymbol{u} \cdot \Delta \in \mathbb{F}$ 和 $\boldsymbol{u} \in \mathbb{F}$.

半诚实安全. 对于半诚实安全, 在不诚实大多数情况下基于秘密分享方法的 MPC 协议主要建立在 GMW 框架下 [13], 采取加法秘密分享实现电路的分布式计算. 加法秘密分享的线性性保证了加法门的计算无需通信. 原始的 GMW 协议 [13] 采用 4 选 1 不经意传输协议实现乘法门计算. 后续的改进工作 [148, 149] 采用 2 选 1 不经意传输协议或 Beaver 技术实现乘法门计算, 且能从布尔电路模型扩展到算术电路模型 [128]. 对于定义在域 $\mathbb{F}$ 上的电路计算, 基于 Beaver 三元组技术的 GMW 类 MPC 协议中的乘法子协议 Π_{milt} 由以下两个阶段组成:

(1) 预处理阶段. 通过运行半诚实安全的 OT 或 OLE 协议, 所有参与方 $P_1, \ldots, P_n$ 计算一个 Beaver 乘法三元组 $([a], [b], [c])$, 满足 $a, b \in \mathbb{F}$ 是随机的域元素, $c = a \cdot b \in \mathbb{F}$.

(2) 在线阶段. 对于乘法门 (i, j, k, MULT) 的输入分享 $[z_i]$ 和 $[z_j]$, 所有参与方运行 $\mathsf{Open}([z_i] - [a])$ 和 $\mathsf{Open}([z_j] - [b])$ 以分别获得 $\epsilon = z_i - a$ 和 $\sigma = z_j - b$, 然后计算 $[z_k] := [c] + \epsilon \cdot [b] + \sigma \cdot [a] + \epsilon \cdot \sigma$. 将 $\epsilon = z_i - a$, $\sigma = z_j - b$ 和 $c = a \cdot b$ 代入 $[z_k]$ 的计算公式, 可验证 $z_k = z_i \cdot z_j \in \mathbb{F}$ 成立.

以上 GMW 类协议也能直接扩展到整数环 $\mathbb{Z}_{2^k}$ 上 (如 $k = 32$ 或 $k = 64$). 当预处理阶段支持计算电路相关的随机数时, GMW 类 MPC 协议在线阶段的通信开销能从每方每个乘法门发送 4 个元素降低到 2 个元素 [37, 150]. 半诚实安全 GMW 类协议的主要效率瓶颈是在预处理阶段生成 Beaver 乘法三元组, 提高 OT/OLE 协议效率可直接改进 GMW 类协议效率.

恶意安全. 在恶意敌手模型下, 设计 MPC 协议的通用方法是利用编译器将半诚实安全的 GMW 协议转化为恶意安全的协议. 这样的编译器包括: GMW 类编译器 [13, 151]、IPS (Ishai-Prabhakaran-Sahai) 编译器及其优化 [57, 152~154]、HVW (Hazay-Venkitasubramaniam-Weiss) 编译器 [155] 等. 然而, 通过这些通用编译器转化得到的恶意安全 MPC 协议的具体效率通常较低. 为改进具体效率, 一种有效手段是利用信息论消息认证码保证秘密值的认证性. MPC 协议主要采用两类信息论消息认证码, 一类是 SPDZ 消息认证码 [108], 另一类是 BDOZ 消息认证码 [107], 其中 SPDZ 类信息论消息认证码主要适用于基于秘密分享的 MPC 协议, BDOZ 类信息论消息认证码主要适用于基于混淆电路的常数轮 MPC 协议. 鉴于本综述的 MPC 部分主要关注基于秘密分享的协议, 我们主要介绍 SPDZ 类安全多方计算协议. Damgård 等 [108, 110] 引入了 SPDZ 恶意安全的 MPC 协议, 仅考虑了大域情况 (即 $|\mathbb{F}| \geqslant 2^\rho$, 其中 ρ 为统计安全参数). 对于大域情况, SPDZ 类安全多方计算协议框架如下.

(1) 预处理认证分享. 参与方 $P_1, \ldots, P_n$ 运行 VOLE 协议生成随机的认证分享 $[\![r_1]\!], \ldots, [\![r_m]\!]$, 然后运行一致性检查协议验证这些认证分享中参与方分享的一致性和密钥的一致性.

(2) 预处理认证三元组. 所有参与方运行 OLE 协议计算多个 Beaver 乘法三元组 $([a_i], [b_i], [c_i])$, $i \in \{1, \ldots, \ell\}$, 然后运行 VOLE 协议将每个 $([a_i], [b_i], [c_i])$ 转化为认证三元组 $([\![a_i]\!], [\![b_i]\!], [\![c_i]\!])$, 最后运行一致性检查协议验证 $c_i = a_i \cdot b_i$ $(i = 1, \ldots, \ell)$ 成立. 有的 SPDZ 类协议中 OLE 与 VOLE 一起组合运行, 并在组合的协议中检查认证三元组的正确性.

(3) 在线阶段. 将 3.3 小节描述的电路计算中加法秘密分享替换为认证分享, 乘法子协议 Π_{milt} 将

以上 Beaver 计算方法中半诚实的乘法三元组替换为认证三元组, 其中参与方 P_i 对于他的每条输入线需要获得认证分享 $[\![r_i]\!]$ 中的随机值 r_i.

在大域的情况下, 有一系列工作 [111, 156~158] 改进了原始 SPDZ 协议的效率. 这些改进协议的在线阶段基本相同, 主要不同在于 VOLE 和 OLE 协议的构造方式 (相关研究现状在下一部分中给出). 对于 SPDZ 协议的在线阶段, 可利用电路相关随机数将通信开销从每方每个乘法门发送 4 个元素降低到 2 个元素 [150]. SPDZ 类协议也能从大域扩展到整数环 $\mathbb{Z}_{2^k}$ 上 [159], 其中整数环 $\mathbb{Z}_{2^k}$ 上运算快于大域上运算, 但 SPDZ 类协议在整数环 $\mathbb{Z}_{2^k}$ 上通信开销高于大域上的通信开销. 对于相等检测、比较、截断等机器学习常用函数, Damgård 等 [160] 证实了在环 $\mathbb{Z}_{2^k}$ 上 SPDZ 类协议比大域上 SPDZ 类协议更加有效. 后来, 多个协议又进一步改进了环 $\mathbb{Z}_{2^k}$ 上 SPDZ 类协议的效率 [161~163].

对于二元域 $\mathbb{F}_2$, 可用 COT 协议替换 VOLE 协议, 用 OT 协议替换 OLE 协议, 从而计算出秘密值在 $\mathbb{F}_2$ 上的认证分享和认证三元组, 在线阶段的计算可直接采用以上 SPDZ 类协议框架. 在大域 $\mathbb{F}$ 上, 可以采用 "牺牲方法 (sacrifice approach)" 检查认证三元组正确性 [108, 111]. 具体而言, 参与方运行 OLE 和 VOLE 协议生成一个随机的认证三元组, 记为 $([\![a]\!], [\![b]\!], [\![c]\!])$, 然后将其用于检查目标认证三元组 $([\![x]\!], [\![y]\!], [\![z]\!])$ 的正确性如下:

(1) 所有参与方运行投币协议生成一个公共的随机数 $r \in \mathbb{F}$.

(2) 所有参与方执行 $\epsilon \leftarrow \mathsf{Open}(r \cdot [\![x]\!] - [\![a]\!])$ 和 $\sigma \leftarrow \mathsf{Open}([\![y]\!] - [\![b]\!])$.

(3) 所有参与方执行 $\tau \leftarrow \mathsf{Open}(r \cdot [\![z]\!] - [\![c]\!] - \epsilon \cdot [\![b]\!] - \sigma \cdot [\![a]\!] - \epsilon \cdot \sigma)$, 然后检查 $\tau = 0$ 是否成立.

然而, 该牺牲方法并不适用于小域 $\mathbb{F}_2$ 情况. 特别地, 直接采用 "牺牲方法" 需要重复执行检查协议至少 ρ 次, 从而需要 ρ 倍的随机认证三元组, 一致性检查效率较低. 有两种方法可获得更高检查效率: 一是 TinyOT 类协议使用的 "Cut-and-Bucketing" 方法 [91~93, 95, 97, 109, 164, 165], 另一种是 MiniMAC 类协议使用的 "批量认证方法" [165~170]. 前者需要 $O(\rho/\log|C|)$ 倍额外通信与计算开销 ($|C|$ 为电路中 AND 门数量), 后者采用了特殊的 MiniMAC 类信息论消息认证码, 仅适用于 SIMD 电路和层级电路.

不经意传输及其算术一般化. OT 协议依赖于公钥密码操作 [171], 目前有基于各种困难假设构造的 OT 协议, 包括: DDH [172, 173], CDH [174, 175], LWE [176], LPN [177], CSIDH [178] 等. 当需要生成大量 OT 时 (特别是在 MPC 协议中), 这些基于公钥操作的 OT 协议效率就会比较低. OT 扩展的概念可解决这一问题 [179, 180]. OT 扩展允许有效扩展少量基于公钥操作的 "base OT" 到大量的 OTs. 例如, 对于 128 比特安全强度, 只需运行基于公钥组件的协议获得 128 个 "base OT", 然后运行 OT 扩展协议可得到一千万个 OTs, 通过迭代方法 [149, 181] 可进一步扩展得到任意多项式数量的 OTs. 目前几乎所有的 OT 扩展协议设计思路都是首先设计 COT 扩展协议[1)], 然后通过标准的方法转化 COT 协议到标准的 OT 协议. 因此, 下面我们仅关注 COT 扩展协议的设计方法. COT 扩展协议大致分为以下 3 类.

• **IKNP (Ishai-Kilian-Nissim-Petrank) 类.** 采用伪随机生成器实现 COT 扩展, 包括文献 [149, 180, 182~186] 等.

• **伪随机相关生成器 (pseudorandom correlation generator, PCG) 类.** 利用 LPN (learning parity with noise) 困难问题中噪音的稀疏性实现 COT 扩展, 包括文献 [181, 187~189] 等.

• **伪随机相关函数 (pseudorandom correlation function, PCF) 类.** 代表性工作包括文献 [190, 191], 同样利用 LPN 噪音稀疏性实现 COT 扩展, 但允许每次生成一个 COT (对比 PCG 类需要一次性扩展得到固定数量的所有 COTs).

1) 一个例外是最近提出的 IKNP 类 OT 扩展协议 [182], 首先设计了子空间 VOLE 协议, 然后转化得到标准的 OT 协议.

表 1　三类 COT 扩展协议效率比较

Table 1　Efficiency comparison of three types of COT extension protocols

COT types	Communication	Computation	Rounds	Assumptions
IKNP	Linear (large cost)	Linear (small cost)	$O(1)$	OWF
PCG	Sublinear (small cost)	Linear (large cost)	$O(1)$	LPN
PCF	Sublinear (small cost)	Linear (larger cost)	$O(1)$	LPN

表 1 比较了 3 类 COT 扩展协议的效率, 并列出了底层困难假设. 与其他两类协议比较, PCF 类 COT 扩展协议 [190, 191] 实际效率最低, 还没有公开已知的实现. PCG 类协议比 IKNP 类协议更适合计算上百万量级的 COTs. PCG 类 COT 协议具有亚线性通信复杂度但需要更大的计算开销, 而 IKNP 类 COT 协议具有线性通信复杂度但计算开销更小.

OT 协议主要用在关于布尔电路的 MPC 协议中, OLE 协议主要用在关于算术电路的 MPC 协议中. OLE 协议有多种不同的设计方法, 举例如下.

• **基于 OT 的比特分解方法.** 利用 Gilboa 乘法思想 [192] 可从 OT 协议构造 OLE 协议 [111]. 该方法计算效率高, 但通信开销也相当高.

• **同态加密方法.** 利用加法同态加密思想可在 RLWE (ring learning with error) 等困难假设下建立 OLE 协议 [156, 161~163, 193~196], 其中同态计算后的密文需要满足 "电路隐私" (circuit privacy) 性质以保证密文没有泄漏同态操作的秘密信息. 为保证恶意安全, 可采用零知识证明来证明密文的正确性. 该方法提供线性通信复杂度, 在利用快速傅里叶变换 (fast Fourier transform, FFT) 技术时可达到准线性 (quasi-linear) 计算复杂度.

• **噪音编码方法.** 利用带噪音的编码可基于 OT 协议构造 OLE 协议 [146, 197~199]. 不同于比特分解方法, 该噪音编码方法具有线性通信复杂度和准线性计算复杂度. 与同态加密方法相比, 该方法需要更多的通信轮数, 具体效率需要实现验证.

• **PCG 方法.** 在 PCG 框架下, 基于 LPN 噪音的稀疏性和分布式点函数, OLE 协议能获得亚线性通信复杂度 [187, 200, 201]. 当以上方法允许选择的输入时 (即允许部分 x, y 由参与方决定), 在 PCG 框架下 OLE 协议的输入是伪随机的. 利用 ring-LPN 困难假设可使 PCG-OLE 协议达到准线性计算复杂度 [200]. 对于计算上百万的 OLE 相关随机数, 该 PCG-OLE 协议 [200] 比以上方法有明显更低的通信开销, 在网络带宽较小时具有较大效率优势. 当扩展 PCG-OLE 协议以计算认证三元组时, Boyle 等 [200] 提出的协议仅包含了两个参与方. 最近的工作 [201] 突破了在 PCG 框架下认证三元组生成协议的参与方数量, 支持多个参与方且保持亚线性通信复杂度, 但具体的通信开销相当大.

VOLE 可看作 OLE 的向量形式, 也能利用比特分解方法、同态加密方法、噪音编码方法加以构造. 为达到恶意安全性, VOLE 协议需要额外运行一致性检查步骤以保证全局密钥 Δ 的一致性. 此外, 还可通过以下方法设计 VOLE 协议:

• **PCG 方法.** 在 PCG 框架下, VOLE 协议的设计方法类似于 PCG-COT 协议, 能达到亚线性通信复杂度和线性计算复杂度. 有一系列基于 LPN 困难假设的工作 [147, 181, 187, 188, 202, 203] 改进了 PCG-VOLE 协议的效率. PCG-VOLE 协议是计算大量 VOLE 相关随机数最有效的方法.

• **PCF 方法.** 在 PCF 框架下, VOLE 协议的设计方法也类似于 PCF-COT 协议. 与 PCG-VOLE 协议必须一次性输出所有相关随机数不同, PCF-VOLE 协议 [190] 能一次输出一个相关随机数, 并支持大量 VOLE 相关随机数生成. 然而, 该 PCF-VOLE 协议 [190] 的具体效率相当低.

4 同态加密

4.1 全同态加密简介

伴随着云计算的发展, 一类特殊的计算场景逐渐引起了学术界的关注: 客户端持有输入 x, 服务器端持有函数 f, 客户端需要在隐藏输入数据 x 的前提下获得运算结果 $f(x)$, 而服务器则需要保护其所持有的函数 f. 这类场景对应于云计算安全中的许多实际问题, 例如, 司机要在不泄露位置信息的前提下通过服务器获取导航数据等.

在解决上述问题的诸多密码学协议和算法中, "密文计算" 的概念无疑是最简单直接的. 这一概念的提出可追溯到 1978 年 Rivest 等 [204] 的工作: 客户端加密其输入 x 并将密文发送给服务器; 服务器能够将函数 f 用于评估 (evaluate) 密文, 并将所得结果发送给客户端; 客户端对其解密并得到 $f(x)$. 显然, 一般的密码算法无法提供针对密文的评估功能, 这就需要设计特定的密码算法. 观察到原始 RSA (Rivest-Shamir-Adleman) 加密算法 (将 x 加密为 $x^e \bmod N$) 可提供密文乘法运算这一现象后, Rivest 等提出了以下公开问题: 可否设计一种支持更通用函数的密文计算功能的加密算法? 使用这类算法能解决哪些实际问题?

延续 Rivest 等 [204] 的思路, 设计能够支持针对密文计算的加密算法 (即同态加密算法, homomorphic encryption, HE) 成为学术界比较感兴趣的一个问题. 除了普通加密算法共有的加密和解密程序外, 同态加密算法通常还提供一个评估函数. 该函数的输入为 x 的密文和函数 f, 输出为 "评估后的密文", 对该密文解密后恰好对应于 $f(x)$. 此外, 同态加密算法还应具备紧凑性 (compactness) 和函数隐私性 (function-privacy), 前者要求对 "评估后的密文" 的解密运算的复杂度与函数 f 无关, 后者要求 "评估后的密文" 不泄露函数 f 的任何信息.

自同态加密的概念提出后, 研究人员提出了许多支持某些函数密文计算功能的加密算法, 包括支持密文乘法计算的 RSA 算法、支持密文加法运算的 Paillier 算法等. 但构造一个能够有效支持任意函数密文计算的全同态加密算法 (fully homomorphic encryption, FHE) 却极为困难, 因此全同态加密曾一度被认为是密码学的 "圣杯".

Gentry [205] 于 2009 年首次给出了全同态加密算法的一个理论可行的蓝图, 这是对全同态加密算法探索的第一个重大突破. 在 Gentry 工作的基础上, 全同态加密研究得到飞速发展, 包括更高的计算效率、更一般的困难性假设以及更广泛的应用在内的一系列改进工作纷纷涌现.

自 2009 年 Gentry 的突破性工作至今, 全同态加密算法根据其构造方法可划分为四代. 第一代以 Gentry [205] 基于理想格的方案以及 van Dijk 等 [206] 基于整数的方案为代表, 这一代算法的通病是错误增长速度过快, 对算法的安全性和效率有较大负面影响. 第二代全同态加密算法起源于 2011 年 Brakerski-Vaikuntanathan [207] 以及 Brakerski 等 [208] 的工作, 这一代算法基于格困难问题, 使用了比第一代算法更通用的安全假设、更优的错误控制技术、更好的明文编码技术, 大幅改进了密文计算效率. 第三代全同态加密算法的研究开始于 Gentry 等 [209] 的工作, 这一代算法使用了与第二代不同的构造模式, 在控制错误增长方面具有很好的潜力. 第四代全同态加密算法以 CKKS (Cheon-Kim-Kim-Song) 算法 [210] 为代表, 其核心思路是使用近似计算取代原有全同态加密算法中的精确计算, 以取得更高的计算效率.

4.2 全同态加密算法的构造

4.2.1 第一代全同态加密

Gentry 的技术路线. 在其开创性工作中, Gentry 给出了基于理想格设计 FHE 的技术路线, 算法的安全性基于理想格上的稀疏子集和问题. 此后, Gentry 受到了 van Dijk 等 [206] 工作的启发, 在其博士论文中设计了基于整数全同态加密算法的雏形, 这一工作在他们随后的论文 [206] 中得以完善, 算法基本原理如下.

令大奇数 p 表示整数加密方案的私钥, 对于明文比特 $b\in\{0,1\}$, 其对应的密文 $c=pq+2r+b$. 当随机选取整数 q,r 满足 $|r|\ll q$ 时, 密文 c 与私钥 p 的整数倍 pq 接近. 因此, 可通过对密文执行模 p 运算得到 $c \bmod p=2r+b\in[-p/2,p/2)$, 再对结果模 2 得到 $(c \bmod p) \bmod 2=b$, 从而实现解密.

对于上述整数加密方案, 可以对两个相同密钥加密的密文执行加法或乘法运算, 即有 $c_i=q_ip+2r_i+b_i, i=1,2$, 令 $c^+=c_1+c_2, c^\times=c_1\cdot c_2$, 则 $c^+,c^\times$ 与原密文 c_1,c_2 结构相似且满足:

$$c^+=(q_1+q_2)p+2(r_1+r_2)+(b_1+b_2)=q'p+2r'+(b_1\oplus b_2).$$

$$c^\times=(q_1c_2+q_2c_1)p+2(b_1r_2+b_2r_1+2r_1r_2)+b_1b_2=q''p+2r''+b_1b_2,$$

其中 q',q'' 表示某个整数, $r'\approx r_1+r_2, r''\approx 2r_1r_2$. 因此当初始错误 r_1,r_2 相对于 p 足够小时, 则密文加法和乘法的结果 $c^+,c^\times$ 仍可正确地解密. 也就是说, 该整数加密方案可支持次数较低的多项式密文运算, 所得结果解密后与对明文比特执行在 $\mathbb{F}_2$ 上的多项式运算一致. van Dijk 等 [206] 证明了该方案的安全性可以归约到近似的最大公约数问题, 保证了算法的可证明安全性.

上述算法虽然能在一定程度上实现同态运算, 但不满足紧凑性, 即密文的大小随着评估函数次数增大. 同时, 该算法仅支持较低次数的多项式计算, 这是由于算法中的错误随着乘法次数快速增长, 当错误大于 $p/2$ 时, 算法就无法正确解密. van Dijk 等提出了一种解决紧凑性问题的方法: 首先公开多个不同长度的 p 的倍数, 再使用这些公开参数进行模数转换来控制密文的增长. 对于错误增长导致解密错误的问题, 则需要使用自举技术 (bootstrapping) 来加以解决.

自举技术. Gentry 在其开创性工作中提出: "对于任意的同态加密算法, 若算法支持评估其自身的解密函数电路 (以及一个额外的与非门), 则该算法可以转换为一个全同态加密算法. " 这句话中所提出的 "评估自身的解密函数电路" 指的正是自举技术. 因此, 自举技术是 Gentry 提出的全同态算法技术路线中的核心, 也是当前延续 Gentry 技术路线发展而来的三代全同态加密算法实现全同态计算的关键. 具体来说, 对于任意的密文 c_1,c_2 考虑以下函数:

$$D^*_{c_1,c_2}(\mathrm{sk})\stackrel{\mathrm{def}}{=}\mathsf{NAND}(\mathsf{Decrypt}(\mathrm{sk},c_1),\mathsf{Decrypt}(\mathrm{sk},c_2)).$$

函数 $D^*_{c_1,c_2}(\mathrm{sk})$ 的输入是私钥, 用该私钥解密密文 c_1,c_2 得到明文 b_1,b_2, 再输出 b_1,b_2 的与非结果. 如果同态加密的密文计算深度能够支持对任意密文对 c_1,c_2 执行函数 $D^*_{c_1,c_2}(\mathrm{sk})$ 评估, 则该同态加密方案为可自举的方案, 能够转化为全同态方案, 具体方法如下. 首先公开一个对私钥 sk 加密的密文, 则对于任意的密文 c_1,c_2, 都可以使用该 sk 的加密来同态地评估函数 $D^*_{c_1,c_2}(\mathrm{sk})$ 以实现对 c_1,c_2 的与非运算. 由于与非运算具有逻辑完备性, 可以通过与非门构造所有逻辑运算电路来完成全同态的目标. 同时, 函数 $D^*_{c_1,c_2}(\mathrm{sk})$ 中同态地评估了解密函数, 能够控制错误增长速度, 即当 c_1,c_2 可正确解密时, 函数 $D^*_{c_1,c_2}(\mathrm{sk})$ 的评估结果也能够正确解密. 因此, 可以使用该加密算法构造全同态加密方案.

第一代全同态加密的特点. 第一代全同态加密的主要工作有 Gentry [205] 基于理想格的方案、van Dijk 等 [206] 基于整数的方案及其变种. 这些方案的普遍缺陷是错误增长速度过快限制了同态计算的

深度. 具体来说, 对于初始错误为 δ 的密文, 经过次数为 d 的多项式运算后, 其结果中蕴含的错误大小约为 δ^d. 尽管这些算法的解密函数深度较浅, 但快速增长的错误依然限制了它们对自身解密函数的评估, 由此引起的自举困难降低了第一代全同态加密算法的实用价值.

4.2.2 第二代全同态加密

BV 算法的技术路线. 2011 年, Brakerski 与 Vaikuntanathan [208,211] 给出了基于 LWE 及 RLWE 问题的全同态加密算法的构造方法, 标志着第二代全同态加密算法的开端, 其基本思想如下:

对于一个 LWE 问题的样本 $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s}\rangle + 2e)$ 以及私钥 $\boldsymbol{s}$. 令 $m \in \mathbb{F}_2$ 表示明文, 其加密后的密文为:

$$\boldsymbol{c} = (\boldsymbol{a}, b' = b + m) \in \mathbb{Z}_q^n \times \mathbb{Z}_q,$$

其中 e 表示采样自错误分布 χ 的随机错误. 当解密时, 通过以下公式完成解密运算:

$$(b' - \langle \boldsymbol{a}, \boldsymbol{s}\rangle \bmod q) \bmod 2 = (2e + m) \bmod 2.$$

容易验证当错误 $e < q/2$ 时, 以上加解密流程可以正确得到明文. 若将 $(-\boldsymbol{s}, 1)$ 记为 $\boldsymbol{s}'$, 则有 $m = \langle \boldsymbol{c}, \boldsymbol{s}'\rangle \bmod 2$.

对于两个相同密钥加密的明文 $m_1 = \langle \boldsymbol{c}_1, \boldsymbol{s}'\rangle \bmod 2$, $m_2 = \langle \boldsymbol{c}_2, \boldsymbol{s}'\rangle \bmod 2$, 则有

$$\begin{aligned} m_1 + m_2 &= \langle \boldsymbol{c}_1 + \boldsymbol{c}_2, \boldsymbol{s}'\rangle \bmod 2, \\ m_1 \cdot m_2 &= (b_1' - \langle \boldsymbol{a}_1, \boldsymbol{s}\rangle)(b_2' - \langle \boldsymbol{a}_2, \boldsymbol{s}\rangle) \bmod 2 \\ &= (b_1' b_2' - \langle \boldsymbol{a}_1 b_2' + \boldsymbol{a}_2 b_1', \boldsymbol{s}\rangle + \langle \boldsymbol{a}_1, \boldsymbol{s}\rangle\langle \boldsymbol{a}_2, \boldsymbol{s}\rangle) \bmod 2 \\ &= (d_0 + \langle \boldsymbol{d}_1, \boldsymbol{s}\rangle + \langle \boldsymbol{d}_2, \boldsymbol{s}^2\rangle) \bmod 2. \end{aligned}$$

注意到对于相同密钥加密的明文, 其加法同态性是显然的, 而它们的乘积可以视为一组以 $(1, \boldsymbol{s}, \boldsymbol{s}^2)$ 为私钥的 LWE 问题样本, 若能够将乘积中 $\boldsymbol{s}^2$ 对应的项转换为密钥 $\boldsymbol{s}$ 的表达式, 则可以将密文乘法的结果转换为 LWE 问题的标准形式, 从而进一步执行其他的同态计算, 因此, 这一转换的过程是第二代全同态加密算法的关键技术, 被称为重线性化 (re-linearization technique) 或密钥转换技术 (key switching technique).

密钥转换和模转换技术. 重线性化或密钥转换技术的提出可以解决基于 LWE/RLWE 问题的第二代全同态加密算法在计算密文乘法时, 密钥规模以平方的规模快速增大的问题, 其核心思想是通过将原密钥中的每一项及由这些项所组成的全部二次项使用新密钥加密, 使原密钥加密下的密文乘法的结果可通过新密钥的线性函数表示, 此时可将结果转换为以新密钥加密的标准 LWE 问题样本.

此外, Brakerski 还提出了一项显著降低全同态加密错误增长速度的技术, 称之为模转换技术 (modulus switching techinque). 其核心思想是对于 $\mathbb{Z}_q^n$ 上的密文 $\boldsymbol{c}$, 通过令 $\boldsymbol{c}$ 中的各项分别乘以 p/q 并取整, 可以将密文 $\boldsymbol{c}$ 转换为 $\mathbb{Z}_p^n$ 上的密文, 此时因同态加密积累的错误总量也同步减少了约 q/p 倍. 通过精心选取参数 p 与 q 的大小, 能够将错误的增长速度控制在较小的规模.

第二代全同态加密的特点. 第二代全同态加密算法以 2011 年 Brakerski 等 [207,208,211] 的工作为代表. 第二代算法在第一代算法的基础上实现了多个实用性的优化, 包括更好的错误增长控制技术、更标准的困难性假设以及多项效率优化技术. 利用错误增长控制技术可使第二代全同态加密算法中错误增长速度从密文计算深度的线性量级降为对数量级, 这一技术是 "分层" 同态算法以及全同态算法得以实用化的关键. 第二代全同态加密算法使用的标准困难问题假设包括容错学习问题 (learning

with errors) 或 NTRU 问题等, 这些问题的困难性有着更完备的归约并经历了长时间的考验, 能够为全同态加密算法奠定更牢固的安全性基础. 第二代算法在效率优化方面的改进主要包括明文打包技术 (packing) [208, 212, 213] 以及自举效率优化 [214~216], 这些优化技术大幅改进了全同态加密算法的计算效率 [217, 218], 使算法的密文计算效率较明文计算效率在渐进复杂度上仅付出约 $O(\log^k n)$ 的额外开销, 其中 n 表示安全强度, k 为常数.

4.2.3 第三代全同态加密

GSW 算法的技术路线. 第二代全同态加密算法的核心技术是密钥转换和模转换, 2013 年, Gentry 等 [209] 给出了一种基于 LWE/RLWE 问题设计全同态加密算法的新思路, 在他们的技术路线中, 无需使用密钥转换和模转换技术仍可以有效控制同态加密的错误增长, 其主要设计思路如下:

对于一个 LWE 问题的样本 $(\boldsymbol{a}, b = \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e)$, 令公钥为 $\boldsymbol{A} = (\boldsymbol{a}, b)$, 私钥为 $\boldsymbol{s}' = (-\boldsymbol{s}, 1)$. 则有 $\boldsymbol{s}'\boldsymbol{A} = e \approx 0$.

令 $m \in \mathbb{F}_2$ 表示明文, 其加密后的密文为

$$\boldsymbol{C} = \boldsymbol{A}\boldsymbol{R} + m\boldsymbol{G},$$

其中 $\boldsymbol{R}$ 是 $\mathbb{F}_2$ 域上的随机矩阵, $\boldsymbol{G}$ 表示块对角矩阵.

解密时需要计算 $\boldsymbol{s}'\boldsymbol{C} = \boldsymbol{s}'\boldsymbol{A}\boldsymbol{R} + m\boldsymbol{s}'\boldsymbol{G}$, 当 $\boldsymbol{s}'\boldsymbol{A} \approx 0$ 且 $\boldsymbol{R}$ 较小时, 则有 $\boldsymbol{s}'\boldsymbol{A}\boldsymbol{R} \approx 0$. 因此 $\boldsymbol{s}'\boldsymbol{C} \approx m\boldsymbol{s}'\boldsymbol{G}$, 即可根据已知的私钥 $\boldsymbol{s}'$ 获得明文.

对于两个相同密钥加密的明文 $\boldsymbol{C}_1 = \boldsymbol{A}\boldsymbol{R}_1 + \boldsymbol{m}_1\boldsymbol{G}, \boldsymbol{C}_2 = \boldsymbol{A}\boldsymbol{R}_2 + \boldsymbol{m}_2\boldsymbol{G}$, 容易观察到:

$$\boldsymbol{s}'(\boldsymbol{C}_1 + \boldsymbol{C}_2) = \boldsymbol{s}'\boldsymbol{A}\boldsymbol{R}_1 + m_1\boldsymbol{s}'\boldsymbol{G} + \boldsymbol{s}'\boldsymbol{A}\boldsymbol{R}_2 + m_2\boldsymbol{s}'\boldsymbol{G} \approx (m_1 + m_2)\boldsymbol{s}'\boldsymbol{G}.$$

可知能够满足加法同态性. 而对于密文乘法, 则需要定义非对称乘法规则如下:

$$\boldsymbol{s}'\boldsymbol{C}^{\times} = \boldsymbol{s}'(\boldsymbol{C}_1 G^{-1}(\boldsymbol{C}_2)) \approx m_1\boldsymbol{s}'\boldsymbol{G} \cdot m_2 = (m_1 \cdot m_2)\boldsymbol{s}'\boldsymbol{G}.$$

即 $\boldsymbol{C}_1 G^{-1}(\boldsymbol{C}_2)$ 所得结果对应于两个明文乘积的密文.

非对称密文乘法. 第三代全同态加密算法的核心是非对称乘法, 该技术利用了以下原理: 对于任意的整数 $x \in \mathbb{Z}_q$, 可将其表示为一组二进制向量 $x = \sum_{i=0}^{\lceil \log q \rceil} 2^i x_i$, 即 $x = \boldsymbol{g} \cdot \boldsymbol{v}$, 其中 $\boldsymbol{g} = \{1, 2, 2^2, \ldots, 2^{\lceil \log q \rceil}\}$. 同理对于任意向量 $\boldsymbol{v} \in \mathbb{Z}_q^n$, 可以将 $\boldsymbol{g}$ 扩展为块对角矩阵 $\boldsymbol{G} \in \mathbb{Z}_q^{n \times n\lceil \log q \rceil}$, 满足 $\boldsymbol{v} = \boldsymbol{G}\boldsymbol{v}'$, 其中 $\boldsymbol{v}'$ 表示 $\boldsymbol{v}$ 中每个元素的二进制展开, 记为 $G^{-1}(\boldsymbol{v})$, 显然有 $GG^{-1}(\boldsymbol{v}) = \boldsymbol{v}$. 此外, 由于 $G^{-1}(\boldsymbol{v})$ 中各元素均取自 $\{0, 1\}$, 可知该矩阵的范数较小, 这是确保上述非对称乘法成立的关键.

第三代全同态加密的特点. 2013 年, Gentry 等 [209] 给出了一种不同于第二代全同态加密算法的设计思路, 其核心思想是采用非对称的乘法降低错误增长速度. 具体来说, 该方案的同态乘法具有非对称性, 即 $c_1 \otimes c_2$ 所得的密文不同于 $c_2 \otimes c_1$ 所得密文. 在该同态乘法中, 错误的增长速度也与乘法顺序有关, 被乘数对错误增长速度的影响较乘数更大. 利用这一性质可进一步降低密文乘法的错误增长速度, 从而使第三代全同态算法在参数选取和安全性归约上更具优势、算法流程更加简洁.

4.2.4 第四代全同态加密

CKKS 算法的技术路线. 第四代全同态算法以 CKKS 算法 [210] 为代表, 其加解密流程与第二代全同态加密算法类似, 主要区别在于其采取了近似计算的策略, 即所得计算结果附带一定的误差, 这一

特点虽然降低了算法的计算精度, 但大幅提高了算法的运算效率, 因此在一些全同态分类中将 CKKS 算法及其改进称为第四代全同态加密算法, 该算法的设计思想如下.

对于一个 LWE 问题的样本 $(\boldsymbol{a}, b=-\langle \boldsymbol{a}, \boldsymbol{s}\rangle+e \bmod q)$, 则公钥为 $(b, \boldsymbol{a})$, 私钥为 $s'=(1, \boldsymbol{s})$, 评估密钥为 $\mathrm{evk}=(b'=a's+e'+Ps^2 \bmod Pq, a') \in \mathbb{Z}_{Pq} \times \mathbb{Z}_{Pq}$. 令 $m \in \mathcal{R}$ 表示明文多项式, 其加密后的密文为:

$$\boldsymbol{c}=\mathcal{ZO}(0.5)(b, \boldsymbol{a})+(m+e_0, e_1),$$

其中 $\mathcal{ZO}(0.5)$ 表示以 0.5 的概率输出 1 或 -1 的随机函数. 解密时则需要计算 $\langle \boldsymbol{c}, s'\rangle=m+e_0+e_1 s+\mathcal{ZO}(0.5)e \approx m$.

对于两个相同密钥加密的明文, $\boldsymbol{c}_1=\mathcal{ZO}(0.5)(b, \boldsymbol{a})+(m_1+e_0, e_1)=(b_1, a_1), \boldsymbol{c}_2=\mathcal{ZO}(0.5)(b, \boldsymbol{a})+(m_2+e_0', e_1')=(b_2, a_2)$, 容易验证其加法同态性:

$$\langle(\boldsymbol{c}_1+\boldsymbol{c}_2), s'\rangle=\langle \boldsymbol{c}_1, s'\rangle+\langle \boldsymbol{c}_2, s'\rangle \approx m_1+m_2.$$

对于乘法, 令 $c^{\times}=(d_0, d_1)+\lfloor P^{-1} \cdot d_2 \cdot \mathrm{evk}\rceil$, 其中 $(d_0, d_1, d_2)=(b_1 b_2, a_1 b_2+a_2 b_1, a_1 a_2)$, 可以验证

$$\langle \boldsymbol{c}_1, s'\rangle \cdot \langle \boldsymbol{c}_2, s'\rangle=(b_1+a_1 s) \cdot(b_2+a_2 s)=b_1 b_2+(b_1 a_2+b_2 a_1) s+a_1 a_2 s^2 \approx m_1 \cdot m_2,$$

$$\langle c^{\times}, s'\rangle=d_0+d_1 s+\langle\lfloor P^{-1} \cdot d_2 \cdot(b', a')\rceil,(1, s)\rangle \approx d_0+d_1 s+d_2 s^2=\langle \boldsymbol{c}_1, s'\rangle \cdot\langle \boldsymbol{c}_2, s'\rangle,$$

即 $\langle c^{\times}, s'\rangle \approx m_1 \cdot m_2$.

第四代全同态加密的特点. 在经典的同态加密算法中, 为获得准确的计算结果, 通常有两类明文编码方式, 一种利用密文空间中的高比特位保存明文, 另一种利用密文空间中的低比特位保存明文. 例如: 明文空间模 t, 密文空间模 q, 则前者的解密运算可表示为 $\langle c, \mathrm{sk}\rangle=qI+(q/t)m+e$, 后者的解密运算则可表示为 $\langle c, \mathrm{sk}\rangle=m+te$. 为确保解密正确性, 两者均要求错误 e 较小, 这一条件限制了同态加密算法的参数选取与计算深度. 而在 CKKS 算法中, 解密运算表示为 $\langle c, \mathrm{sk}\rangle=qI+m+e$, 此时错误 e 与明文 m 直接求和, 无法通过取整计算将错误从结果中消去. 但在另一方面, 由于放弃计算 m 的精确结果, CKKS 算法仅需考虑 e 与 m 的相对大小 (即相对误差), 因此能够在参数选取和计算过程中采取更直接的方法控制错误的增长速度, 如: 使用模转换技术同步减小明文和错误, 从而令错误大小随算法深度呈线性增长而非指数增长.

4.2.5 全同态加密总览

当前, BGV (Brakerski-Gentry-Vaikuntanathan) [207], BFV (Brakerski-Fan-Vercauteren) [219], TFHE (fully homomorphic encryption over the torus) [220] 和 CKKS [210] 是应用最广泛的全同态算法, 涵盖了第二到四代全同态加密构造方案, 值得指出的是, 全同态加密算法的四代构造并非改进与替代的关系, 而是各具特点和适用场景, 正因如此, 当前学术界呈现出上述四种算法同步发展的现状.

总的来说, 第二代和第四代全同态加密算法均可通过高效的明文打包技术实现对多个明文的并行计算, 非常适合计算量较大的数值计算, 其中第二代适用于需要精确计算的场景而第四代面向近似计算场景; 第三代全同态加密算法不支持明文打包, 但其设计结构对于逻辑运算友好, 能够高效地完成逻辑门形式的密文运算, 如表 2 所示.

4.3 全同态加密的应用

同态加密算法为许多应用场景提供了简洁易理解的解决方案, 享有密码学的 “瑞士军刀” 称号. 下面介绍几种典型的全同态加密应用.

表 2　各代全同态加密方案比较

Table 2　Comparison of FHE schemes

Generation	2nd	3rd	4th
Schemes	BGV, BFV	TFHE	CKKS
PROS	Efficient packing Fast linear computation Efficient leveled design	Fast bitwise computation Fast bootstrapping	Efficient packing Fast (approximate) linear computation Efficient leveled design
CONS	Slow bootstrapping Slow non-linear computation	Not support packing	Slow bootstrapping Slow non-linear computation

外包存储及外包计算. 同态加密最直接的应用场景是外包计算. 该应用中, 云端提供大规模存储和高性能计算服务, 客户端 (如: 小型企业) 将私有数据以同态加密形式保存在云端. 云端可利用自身的计算能力和同态加密性质直接对这些密文数据执行操作并将密文结果返回给客户端, 客户端对密文解密即可获得需要的计算结果. 这里, 同态加密算法提供了一种简洁的云计算安全解决方案, 既保护了云上数据的安全性, 又使云平台具备了对云上数据操作的能力.

隐私信息检索及查询. 另一个全同态加密的典型应用场景是面向数据库或搜索引擎的隐私信息查询. 该应用中, 服务器拥有大型数据库并提供检索服务, 客户端可借助全同态加密技术实现对该数据库的隐私信息检索 (private information retrieval) [221], 既获取服务器检索数据又避免服务器得到关于查询条目的任何信息. 具体来说, 服务器可利用数据库构造密文检索函数 $f_{db}(i) = db[i]$, 客户端加密需要查询的索引 i 并发送给服务器, 服务器使用评估函数得到 $f_{db}(i)$ 的密文结果并将其返回客户端, 客户端解密后即达到隐私信息检索的目的. 类似做法也可应用于更复杂的查询操作 (如数据库 SQL 查询、搜索引擎的自由格式查询等) 以满足更实用化的隐私检索和查询任务的需要.

通用两方安全计算. 上述外包计算和隐私信息检索都属于安全多方计算的研究范畴, 可看作通用安全多方计算研究中的两种特例. 事实上, 一般化的安全多方计算模型也可以通过全同态加密算法实现. 在通用的两方安全计算模型中, 参与方 A 持有输入 x, 参与方 B 持有输入 y, 两方共同计算某个已知函数 F, 参与方 A 能获得结果 $F(x, y)$, 参与方 B 得不到任何额外信息. 在半诚实敌手模型中, 参与方 A 可使用其公钥加密输入 x 并发送给 B, 参与方 B 评估函数 $F_y(x) = F(x, y)$ 并将密文结果返回给 A. 这就实现了半诚实假设下基于全同态加密的通用两方安全计算模型. 在这个过程中, 同态加密算法的语义安全性 (semantic security) 保证了参与方 B 无法获取额外信息. 使用隐藏评估函数的同态加密方案可以防止参与方 A 获取除结果 $F(x, y)$ 外的额外信息. 利用标准的技术 [222] 可以将该协议进一步转化为恶意假设下的安全多方计算协议.

4.4　全同态加密不足与展望

自 2009 年 Gentry 的工作以来, 全同态加密在过去的十余年间经历了快速发展, 四代各具特色的全同态加密构造方案相继提出伴随着广泛的应用, 已充分体现出全同态加密在学术界和产业界的重要意义. 然而, 较高的计算复杂度和较大的密文规模仍然是全同态加密应用落地的一大障碍. 此外, 尽管第二到四代全同态加密在不同的应用场景中各有所长, 例如, 第二代和第四代高效的明文打包技术适用于矩阵运算、第三代支持高效的非线性函数运算等, 但仍然缺乏一种通用的方案能够同时提供上述优点. 一种可能的尝试是构造一种混合全同态加密架构, 通过实现这三代方案之间的自动切换以实现不同场景下的效率优势 [223, 224].

5 零知识证明

5.1 零知识证明简介

现代网络通信中, 经常需要一方向另一方提供一些特定的数据或是凭证. 如果这些数据中包含有隐私数据, 则会产生隐私泄露, 导致用户体验下降和服务方信用受损问题. 零知识证明协议 (zero-knowledge proof) 可有效解决这种问题. 这类协议通常包含一个证明者 (prover, P) 和一个验证者 (verifier, V), 证明者向验证者提供一系列 (交互或非交互) 信息来证明某个公开的数据 x 属于一个 **NP** 语言 L. 通过零知识证明, 证明者可以在不泄露自己的隐私数据、即 $x \in L$ 的证据 (witness) 的情况下向验证者证明 x 是符合条件的数据.

5.2 零知识证明定义

零知识证明协议需要满足以下基本要求:

(1) 完备性. 如果证明者和验证者都是诚实的且 $x \in L$, 验证者能以压倒性概率接受证明. 如果以概率 1 接受证明的话, 则称之为完美完备性.

(2) 可靠性. 任何 (恶意的) 证明者在 $x \notin L$ 的情况下不能 (或以可忽略小的概率) 通过验证. 如果以概率 0 通过验证, 则称之为完美可靠性.

(3) 零知识性. 任何 (恶意的、半诚实的) 验证者均不能从证明中得到任何额外的信息.

完备性和可靠性比较直观和容易理解. 我们通常利用模拟器来定义零知识性. 如果一个不知道证明者私有输入的模拟器能模拟出和验证者的交互, 且该交互的内容 (transcript) 与真实交互的内容计算不可区分, 则说明协议没有泄露额外信息. 在构造零知识证明协议时, 可优先构造具有完备性和可靠性的协议, 最后再添加零知识性, 下文中基于 GKR (Goldwasser-Kalai-Rothblum) 协议的构造即为这种类型.

零知识的知识证明. 有时, 我们需要 P 证明他自己不仅了解 $x \in L$, 同时了解一个证据 w 使得 $x \in L \Leftrightarrow (x, w) \in R$. 由计算复杂性的知识可知, 对于一个 **NP** 语言 L, 验证其对应的关系 R 是个 **P** 问题. 这种零知识证明被称作零知识的知识证明 [225], 我们对其有额外的 "知识可靠性" 要求.

(2′) 知识可靠性. 如果一个 (恶意的) 证明者 P 可使验证者以概率 p 接受证明, 那么存在一个提取器 E 满足以下条件: 给予提取器一个可退回步骤的证明者的黑盒, 他能以 $p - \epsilon$ 的概率提取出一个 w' 使得 $(x, w') \in R$.

在知识可靠性中, 对于不同计算能力的 (恶意的) 证明者使用不同的协议名称. 如果协议中的证明者是不限算力的, 这种协议称为知识证明 (proof of knowledge). 如果考虑的是概率多项式的证明者, 这种协议称为知识论证 (argument of knowledge).

(非) 交互式证明. 根据证明者和验证者是否存在交互可将零知识证明分为交互式和非交互式零知识证明 (NIZK): 只需一次消息发送的为非交互式证明, 需要双方交互信息的为交互式证明. 利用 Fiat-Shamir 转换 [226] 可将交互式证明[2)]转换为非交互式证明, 所以一般先构造交互式证明.

零知识证明的衡量标准. 衡量零知识证明协议效率的主要指标是证明时间、验证时间以及证明长度. 目前的零知识证明协议 (文献 [227∼229] 等) 大多有 (对输入长度) 对数或对数多项式级验证时间和证明长度, 以及拟线性级证明时间. 文献 [230] 首次引入了 zk-SNARK (zero-knowledge succinct non-interactive argument of knowledge) 这个名称用以描述高效的零知识证明. 最初使用的 zk-SNARK 记

2) 即验证者的挑战是公开随机的.

号只要求对于原问题运行时间的对数多项式级验证时间和证明长度, 近 5 年提出的大部分 zk-SNARK 能达到拟线性 ($n\log^c n$, 其中 c 为常数) 证明时间, 亚线性 (根号或对数) 证明长度和验证复杂度.

5.3 零知识证明的构造组件

5.3.1 承诺算法

承诺 (commitment) [231] 是零知识证明协议的关键组件之一. 承诺协议将需要被承诺的元素 x 打包为承诺 C, 还有一个打开算法用于验证 C 和 x 的关系 (即 C 由 x 承诺所得). 承诺通常需要具备以下两个性质.

(1) 隐藏性. 通过承诺 C 无法得知关于被承诺元素 x 的任何信息. 根据敌手的计算能力又可分为完美隐藏性和计算隐藏性.

(2) 绑定性. 对于一个承诺 C, 敌手难以找到一个不同于 x 的被承诺元素 x', 使得打开算法也能通过验证. 根据敌手的计算能力又可分为完美绑定性和计算绑定性.

5.3.2 多项式承诺与编码方案

多项式承诺 (polynomial commitment) 是对承诺协议的推广. 多项式承诺将一个需要被承诺的多项式 $P(X)$ 打包为一个承诺 C, 通过打开算法给出 $P(X)$ 在 z 点的取值 $P(z)$ 的凭证 π, 可用该凭证 π 和承诺 C 一起验证 $P(z)$ 的正确性. 多项式承诺也需要符合承诺的性质, 但是隐藏性会变得更为复杂, 也可能需要其他额外的性质 (如可提取性) [232]. 多项式承诺使我们可以在某一个点揭露函数的取值而不暴露函数本身. 许多多项式承诺协议基于公共参考串 (common reference string) 设计 [227, 232~234], 需要可信方来生成这个多项式. 如果零知识证明协议使用这样的多项式承诺的话, 就得需要一个可信第三方[3)][235]. 通过编码方案可以得到和多项式承诺功能相仿的协议 [236], 在通信复杂度上和证明时间上有更大的代价, 但是可以避免可信第三方问题以及获得后量子的特性.

5.3.3 谕示器 (oracle)

谕示器是一个理论模型, 提供了黑盒接口供用户 (验证者) 查询一个函数在特定点上的取值. 零知识证明的理论构造经常使用谕示器来进行初步构造 [227, 232], 在具体实现中再使用承诺算法替换谕示器.

5.3.4 交互式谕示器证明 (interactive oracle proof)

交互式谕示器证明. 零知识证明协议经常使用交互式谕示器证明 [237]. 该模型分为若干个交互步骤, 在每个步骤中由验证者给证明者发送一个挑战, 证明者根据挑战发送一个谕示器. 验证者可以在任意时间查询已经收到的谕示器, 并根据谕示器的回答决定接下来的挑战. 在所有交互步骤结束后, 验证者可以继续查询并根据所有的查询结果判断是否通过验证.

公开投币的交互式谕示器证明. 如果验证者的挑战为随机选取且不随谕示器的回答而改变, 则称其为公开投币的 (public-coin). 此时可将所有谕示器查询放至交互步骤以后, 变成一个查询步骤, 验证者根据查询步骤得到的结果进行判断.

5.4 简洁的零知识证明的构造

经过数十年发展, 零知识证明已经有了各类比较成熟的构造. 零知识证明协议的构造可分为前端

3) 也存在不需要可信第三方的多项式承诺, 如文献 [235].

表 3 零知识证明协议比较

Table 3 ZKP comparisons

	Representative construction	Proving time	Communication cost	Verification time	Trusted initialization	Post-quantum
QAP + Linear encodings	Gro16 [233]	$O(N\log N)$	$O(1)$	$O(1)$	Each circuit	×
R1CS + Polynomial commitment	Plonk [227]	$O(N\log N)$	$O(1)$	$O(1)$	One-time setup	×
R1CS + Encodings	Fractal [236]	$O(N\log N)$	$O(\log N)$	$O(\log N)$	None	✓
Circuit + Polynomial commitment	Libra [238]	$O(N)$	$O(d\log N)$	$O(d\log N)$	One-time setup	×
AIR + Encodings	STARK [229]	$O(N\log^2 N)$	$O(\log^2 N)$	$O(\log^2 N)$	None	✓

和后端. 前端将不同的 NP 问题转化为便于构造零知识证明的 NPC 问题, 后端将 NPC 问题转化为零知识证明协议. 表 3 [227, 229, 233, 236, 238] 在证明复杂度、通信复杂度、验证复杂度、对可信方初始化[4)]的需求程度以及是否具有后量子特性 5 个方面对不同构造方式的零知识证明进行了简单对比.

5.4.1 前端

R1CS 与 QAP. R1CS (rank-1 constraint system) 和 QAP (quadratic arithmetic programming) [239] 是描述算术电路的两种方式, 二者几乎等价. R1CS 的实例包含 $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ 三个矩阵以及一个向量 $\boldsymbol{x}$, 满足 $\boldsymbol{Ax}\circ\boldsymbol{Bx}=\boldsymbol{Cx}$ ($\circ$ 表示元素相乘). 对于一个零知识证明, 可以构造一个验证其关系的算术电路, 以 (x,w) 为输入, 输出 0 或 1 表示是否通过验证. 通过这种做法, 可以将任意零知识证明通过电路转化为 R1CS 问题:

$$\boldsymbol{A}_{m\times n}(1,x,w)^{\mathrm{T}}\circ\boldsymbol{B}_{m\times n}(1,x,w)^{\mathrm{T}}=\boldsymbol{C}_{m\times n}(1,x,w)^{\mathrm{T}},$$

其中矩阵 $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ 根据电路生成, 列数和行数分别与电路规模和乘法门的数量有关. 然后, 再构造多项式相关的零知识证明去解决这类问题. QAP 方式则是将矩阵的每一列利用插值法写成一个多项式的形式, 文献 [239] 首次提出了可以将电路转化为 QAP 的模式. 这类前端易于将任意电路转化并构造零知识证明, 但是现有协议的转化过程中不会优化有特殊结构的电路 (如重复结构).

AIR 与 PAIR. AIR (algebraic intermediate representation) [240, 241] 是另一种刻画 NP 问题的方式. 这种方式更接近于寄存器模型. 构造一个矩阵, 具有程序运行时间的行数和存储单元的列数, 每两行之间 (即每两步的储存单元之间) 以一些多项式作为限制, 再对其中一些单元格的内容进行一定限制 (相当于初始值), 由此就可以将一个程序转化为一些多项式限制和一些初始值限制. 如果这个程序是 NP 问题的关系的验证程序, 我们就把一个 NP 问题转化为一个证明多项式限制和初始值限制可满足性问题. 将每一个储存单元对应的各时间的取值 (即每列) 通过多项式插值写成一个多项式[5)], 此时每个储存单元对应一个多项式, 每个时刻的储存单元对应一个多项式在某一个点上的取值. 我们可以借此将多项式限制中的每个元素更换为一个多项式在某个点的取值, 此时多项式限制转化为 (储存单元) 多项式的复合在某些点取值上的限制. 由于这类零知识证明的构造接近于 CPU 的运行方式, 故可用较直觉的方式构造 zk 虚拟机 (zkVM) [242], 把 CPU 的循环运行改写成多项式的形式.

4) 分为对每个需要证明问题都需要可信方初始化、只需要一次全局的可信方初始化以及不需要可信方初始化.

5) 通常使用 FFT 算法进行插值, 此时插值点构成一个循环群.

对于写成 AIR 形式过大的 NP 问题, 文献 [240] 中也给出了一种解决方案, 即 PAIR (preprocessed AIR). PAIR 引入了一个置换 (模拟了内存交互), 在保证渐进效率的前提下完成任意 NP 问题的模拟. 但该构造的其形式复杂性太高, 还没有具体的实现. 在不使用 zk 虚拟机的情况下, 基于 AIR 的零知识证明对于证明有周期性的 NP 关系更高效 [241].

5.4.2 后端

线性编码与线性证明. 这类后端通常以 QAP 问题作为前端. 证明者将 QAP 问题中的多项式编码为公共参考串中的元素组合, 以此为证明发送给验证者. 验证者通过计算公共参考串和证明是否满足特定的二次关系 (线性编码需要具有可用于这种验证的性质) 来验证整个协议 [239]. 文献 [233] 使用该后端给出了一个基于 QAP 的高效零知识协议, 这是目前为止基于 R1CS 或 QAP 的最高效的零知识协议. 这类后端的缺点在于: 需要分离电路中公开和非公开的输入并写入公共参考串, 因此对于不同的电路需要生成不同的公共参考串, 而公共参考串的生成需要可信第三方, 较为昂贵.

多项式承诺与代数证明. 这类后端通常以 R1CS 和 QAP 问题作为前端. 这类证明一般先构造一个 R1CS 或 QAP 问题的代数证明[6]. 由于多项式承诺可以在某一个点揭露函数的取值而不暴露函数本身, 这与谕示器的访问模式相似, 故可以将其中的谕示器用多项式承诺来替代. 不同于上一种方式中对公共参考串的使用, 这类后端中公共参考串仅用于生成和验证承诺[7], 不直接参加整个协议的验证, 且不需要再将输入写入公共参考串. 故这类后端可以对任意电路使用一个通用参考串. 文献 [243] 在理论上设计了这类后端, 但由于复杂度太高而未被实现. 文献 [234] 降低了复杂度, 文献 [227, 232] 进一步对方案加以改进. 文献 [235] 对多项式承诺进行了修改, 去除了公共参考串, 使协议不再需要可信方参与. 该类后端的多项式承诺具有高效的特点, 能够使通信和验证复杂度维持在常数 [232] 或是将证明复杂度维持在线性 [244]; 但是多项式承诺大都不是量子安全的, 因此使用多项式承诺的协议也不是量子安全的.

Sumcheck 协议 [245] 是直接将电路作为前端的可验证计算协议. 该协议由证明者向验证者提供一个多元函数 $f(x_1,\ldots,x_n)$ 在 $\mathbb{F}^n$ 上的和 (在 GKR 协议中使用 $\mathbb{F}=\mathbb{Z}_2$), 即 $H=\sum_{\boldsymbol{x}\in\mathbb{F}^n} f(\boldsymbol{x})$ 以及一个交互式的计算证明. 将其运用至分层算术电路上, 把每一层电路的输出值视为一个关于门的二进制编号的多项式, 在一个层多项式某个位置的门所对应的函数值仅依赖于上层的两个输入的值. 记第 i 层电路[8]中二进制编号为 x 的门的输出值为 $V_i(x)$, β 为判定相等的示性函数, add 与 mul 为判定第 i 层的 p 号 (加法、乘法) 门的输入是否为上一层编号为 ω_1,ω_2 的门示性函数, 则有

$$V_i(x)=\sum_{p,\omega_1,\omega_2}\beta(x,p)(\mathrm{add}_i(p,\omega_1,\omega_2)(V_{i+1}(\omega_1)+V_{i+1}(\omega_2))+\mathrm{mul}_i(p,\omega_1,\omega_2)(V_{i+1}(\omega_1)V_{i+1}(\omega_2))),$$

其中 $(p,(\omega_1,\omega_2))$ 分别遍历第 $(i,i+1)$ 层的门的编号. 由此利用多组示性函数的组合将其转化为 $\mathbb{Z}_2^n$ 上的上层函数的和, 逐层递归即得到 GKR 协议 [246]. GKR 协议是一个代数证明, 再加入多项式承诺即为零知识证明. 后来, 文献 [238] 将 GKR 通过零知识的多项式承诺的方式转化为零知识协议, 通过分步处理 Sumcheck 函数将分层一致电路的证明时间降至线性 (不影响证明长度与验证时间). 文献 [228] 去除了分层电路的限制, 使基于 GKR 的零知识证明能在任意一致电路上达到线性证明时间. 与基于 R1CS(QAP) 的协议不同, 该类协议的在一致电路上的 3 种复杂度不单取决于电路规模, 同时也受电路深度的影响.

6) 多指交互式谕示器证明.
7) 即验证 "谕示器" 的正确运算.
8) 输出层为 0.

哈希函数 (编码方案)、低度测试与代数证明. 这类后端可用于 AIR [240,241] 与 R1CS [236,247], 但实现上有较大区别. AIR 作为前端时, 多项式限制转化为 (储存单元) 多项式的复合在某些点取值上的限制. 设多项式 $P(X)$ 在集合 S 上对应的点值为 0, 则分式 $\frac{P(X)}{\prod_{s_i\in S}(X-s_i)}$ 是一个次数较低的多项式. 如果证明者能证明该分式是一个低次数的多项式, 就可以证明原来的多项式限制成立. 通过调用类似文献 [248] 中证明多项式次数的协议即可完成 AIR 问题的证明. R1CS 作为前端时, 后端利用低度测试构造一个单变量的 Sumcheck 协议 [247] 并调用 Sumcheck 完成线性关系 $Z_{\boldsymbol{M}} = \boldsymbol{M}x$ 的检查 (其中 $\boldsymbol{M} = \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$), 再直接验证 $Z_{\boldsymbol{A}} \circ Z_{\boldsymbol{B}} = Z_{\boldsymbol{C}}$. 文献 [247] 中的验证者时间复杂度为线性的. 文献 [236] 在此基础上结合 [232] 的技术将验证者时间降至对数多项式级. 这类协议由于哈希或是编码方案不需要可信第三方的加入, 总体协议也不需要可信第三方, 且具有后量子特性; 但是这类协议比起使用多项式承诺的协议具有额外的通信和验证复杂度开销.

5.4.3 利用安全多方计算的零知识证明

该类零知识证明的特点是利用多方计算完成证明 (如文献 [249]). 这类协议的流程通常是向验证者 (通过安全多方计算) 直接承诺门的输入输出线上的数值, 并验证数值之间的关系符合门的限制. 这类协议需要更少的本地内存资源, 但是由于基于安全多方计算[9)]的承诺和验证的特点, 这类协议需要消耗大量的通信资源, 且难以转化为非交互零知识证明. 最高效的此类证明 [250] 在任意电路上需要 $\mathcal{O}(N^{3/4})$ 的通信复杂度, 与其他协议的对数或是常数通信复杂度相比几乎线性于电路规模; 在重复证明同一电路时有较好的表现, 但是也同时线性于该电路的规模与重复次数. 因此该类零知识证明难以部署在区块链上[10)].

5.4.4 非通用零知识证明

这类零知识证明通常只用于证明一个特定的问题, 如范围证明[11)]. 由于只需要证明一个特定的问题, 因此构造通常比较简单. 一般利用 Σ- 协议来构造这类问题的协议. Σ- 协议有 3 次消息传递 (move), 遵循承诺 – 挑战 – 回应三阶段形式. 著名的 Schnorr 协议 [251] 就是这种协议的一个具体例子, 门罗币的范围证明也属于这种形式. Σ- 协议对于承诺的要求很少, 各种类型的承诺均可以应用过来, 因此利用格上的承诺就可以很简单地获得后量子特性 [252].

5.5 零知识证明的应用

零知识证明近年来发展迅速, 是隐私保护的重要工具, 具有广阔的应用前景. 以下介绍零知识证明的几种典型应用.

区块链与数字货币. 零知识证明已在部分密码货币中用于隐私保护, 如门罗币和 ZCash. 以太坊中也有链下应用零知识证明 zk-Rollup 以节省链上空间. 由于区块链不适合进行大量交互和储存, 非交互式零知识证明更为合理. 零知识证明用于保护区块链中交易者的隐私或是减少链上验证交易时间和储存空间的消耗 (计算扩容、存储扩容). 在保护隐私方面 (如 ZCash 中), 零知识证明可在证明交易合法的同时将交易双方和金额作为证据隐藏起来. 在计算扩容和存储扩容方面 (如 zk-Rollup 中), 交易者将交易发送给链下的收集者, 收集者将收到的一定数量的交易打包, 计算执行打包交易后系统的状态并将其提交至链上, 同时利用零知识证明来保证该状态修改的正确性. 验证可以在对数多项式时

9) 在协议实现中具体的形式多为 VOLE (vector oblivious linear evaluation)

10) 区块链系统将证明记录在链上, 故需要 (在交互情况下) 通信少的非交互式证明, 这也是零知识证明的主要部署环境.

11) 即证明一个秘密 (如被承诺的数字) 在某个特定的范围中, 如 $[0, 2^n - 1]$.

间内完成, 节约了对该打包交易的验证时间. 证明长度是对数多项式的或是常数, 这只需比直接将交易写入区块链更小的空间即可完成验证.

数据库访问. 利用零知识证明可以完成数据库隐私保护访问. 假设运营商的数据库需要保密, 其散列值是公开的. 用户想确定数据库中是否有特定数据. 此时运营商可以提供一个以数据库内容和查询内容为输入的证明, 证明由两部分电路构成: 第一部分电路证明数据库内容可以计算出该散列值; 第二部分电路证明数据库内容中有/无和查询内容相同的内容. 利用零知识证明, 就算不公开数据库也能保证查询的正确性和可靠性.

外包计算. 零知识证明技术也可用于外包计算, 此时对零知识性的要求较低. 验证者将输入 x 提供给计算者计算 $f(x)$. 计算者提供一个声明的输出值 $y = f(x)$ 以及一个零知识证明. 计算者可以选择公开部分计算过程或将计算过程保密, 只提供结果. 零知识证明的可靠性和完备性保证了计算结果的正确性.

6 不可区分混淆

6.1 程序混淆的简介与定义

程序混淆. 程序代码中通常包含有代码的框架结构、组件间的调用关系、算法思想等信息, 有时还有一些硬编码的字符, 这些硬编码的字符或算法等都可看作程序中隐藏的秘密信息. 软件行业往往并不希望软件中的算法随着软件的售卖而被泄露, 也不希望软件被任意修改 (例如运用反编译等手段对付费软件进行破解). 程序混淆 (program obfuscation) 可解决这一问题[12], 能保证程序中确实隐藏某些秘密信息, 而且即使拥有这个程序并能任意运行该程序的人也无法得知这些秘密信息.

程序混淆器 (program obfuscator) 可实现程序混淆. 一个程序混淆器 (记为 Obf) 可视为一个特殊的编译器, 其输入 (如一段代码) 对应于某个程序, 记之为 P. 编译器将 P 编译后会输出一个混淆后的程序, 记为 $\hat{P}$. 我们要求:

(1) 两个程序 P, $\hat{P}$ 的功能完全一致, 记为 $P \equiv \hat{P}$;

(2) 从实用性的角度出发, Obf, $\hat{P}$ 都应当是高效的;

(3) $\hat{P}$ 要尽可能地隐藏 P 中包含的秘密信息 ("尽可能" 这个说法十分模糊, 下面我们会讨论怎样刻画 "尽可能").

密码学中有两类刻画安全性的方式: 基于模拟的 (SIM-based) 和基于不可区分的 (IND-based). 我们分别讨论这两种安全性定义, 前者称为虚拟黑盒混淆 (virtual black-box obfuscation, $vbb\mathcal{O}$), 后者称为不可区分混淆 (indistinguishability obfuscation, $i\mathcal{O}$)[13].

虚拟黑盒混淆. 虚拟黑盒混淆是指: 对于一个得到了 $\hat{P}$ 的学习者和一个只能对 P 进行黑盒访问的模拟器, 二者的能力是完全一致的. 换言之, 得到 $\hat{P}$ 并没有给学习者带来任何通过黑盒调用无法得到的信息 (即使学习者可以研究 $\hat{P}$ 的运算步骤, 查看计算过程中某些变量的值的变化, 设置断点, 甚至在运算进行过程中直接修改某些寄存器的值并观察产生的影响, 等). 虽然这个安全性定义非常强, 但是事实上它是无法达成的. 比如, 至少有一项能力是学习者有而模拟器没有的, 那就是学习者可以输出一个和 P 功能一致的程序 (这个程序就是 $\hat{P}$), 而模拟器仅靠对 P 进行多项式次黑盒访问根本不

12) 也许有人认为一份写得极为糟糕的代码也能达到类似的效果. 但这种做法的有效性无法被严格证明, 同时也会增加软件产生 bug 的风险, 并带来极高的维护成本.

13) $i\mathcal{O}$ 既指不可区分混淆也可以指不可区分混淆器, 具体可结合上下文判断.

可能写出一份和 P 功能一致的代码. 在 2001 年, Barak 等 [253] 最早注意到了这一点并证明了不存在通用的虚拟黑盒混淆 ($vbb\mathcal{O}$ 和 $i\mathcal{O}$ 的定义就是他们在文献 [253] 中提出的).

不可区分混淆. 不可区分混淆是指: 对于两个功能完全一致的程序 $P_0 \equiv P_1$, 任何区分器都无法区分 $\hat{P}_0$ 和 $\hat{P}_1$. 这个安全性的定义直觉上并没有特别强 (事实上, 即使 $\mathsf{P} = \mathsf{NP}$, $i\mathcal{O}$ 仍然能够存在, 但这种情形下 $i\mathcal{O}$ 几乎隐藏不了什么信息). 但是在 2007 年, Goldwasser 等 [254] 证明了 $i\mathcal{O}$ 是有可能被实现的最好的混淆器. 为了简述其思想, 假设存在一个混淆器 Obf, 并证明此时 $i\mathcal{O}$ 至少和 Obf 一样安全. 对于程序 P 和 Obf, 显然有 $P \equiv \mathsf{Obf}(P)$. 根据 $i\mathcal{O}$ 的定义, 有 $i\mathcal{O}(P) \approx i\mathcal{O}(\mathsf{Obf}(P))$. 因为混淆不可能凭空产生新的信息, 因此 $i\mathcal{O}(\mathsf{Obf}(P))$ 蕴含的信息不会超过 $\mathsf{Obf}(P)$, 即 $i\mathcal{O}(\mathsf{Obf}(P))$ 安全性不低于 $\mathsf{Obf}(P)$. 因此, $i\mathcal{O}(P)$ 的安全性也至少和 $\mathsf{Obf}(P)$ 相当.

除此之外, 还有包括虚拟灰盒混淆 (virtual grey-box obfuscation, $vgb\mathcal{O}$)、差异输入混淆 (differing-input obfuscation, $di\mathcal{O}$) 等在内的其他定义. 然而, 一来比 $i\mathcal{O}$ 强的定义可能是不存在的 (例如 $di\mathcal{O}$ 被证明很可能是不存在的 [255]), 二来 $i\mathcal{O}$ 在可能存在的混淆中又是最强的, 因此 $i\mathcal{O}$ 成了程序混淆领域研究的核心与重点.

6.2 不可区分混淆的应用

混淆的应用范围非常广, 不仅仅表现在软件工程领域. 直觉上, 混淆之所以这么有用, 主要是因为其既能够保持程序的功能, 又能够隐藏程序的秘密. 日常生活中, 如果我们需要保护一个秘密, 我们可以把它写下来放进保险柜, 但这样的话我们使用这个秘密时就会非常麻烦. 而混淆给了我们一个非常强大 (甚至有些难以置信) 的能力, 即在不打开保险柜的情况下去使用保险柜里的秘密, 而且就算使用过这个秘密我们仍然不知道它是什么! 不幸的是, 这是虚拟黑盒混淆带来的一种直觉, $i\mathcal{O}$ 并不能将程序变为一个黑盒, 也就是说, 程序中包含的秘密也许是可见的. 那么, 如果我们只有 $i\mathcal{O}$, 能用它来做什么呢, 它会像虚拟黑盒混淆那般有用吗?

我们先回到 $i\mathcal{O}$ 的安全性定义上来. $i\mathcal{O}$ 的安全定义非常弱: 当用 $i\mathcal{O}$ 混淆一个程序后, 我们通常很难说清楚到底哪些信息被成功掩盖了. 这是因为, $i\mathcal{O}$ 只保证功能相同的程序在混淆后看上去是差不多的, 如果所有的程序在实现某个功能时都不得不泄露某个信息, 那即便用了 $i\mathcal{O}$ 也无法保护这个信息. $i\mathcal{O}$ 较弱的安全性一方面使得其存在成为可能, 另一方面也使得我们在利用 $i\mathcal{O}$ 时需格外小心. 如果我们需要论证程序 P 中的某个信息被保护起来的话, 往往要设计一个不包含这个信息、但功能与 P 保持一样的程序 P'. 为此, 研究人员提出了一些新的技术和方法.

2013 年, Garg 等 [256] 提出了双公私钥切换技术, 并通过这项技术完成了从只能用于混淆 NC^1 电路的 $i\mathcal{O}$ 到可用于混淆所有 $\mathsf{P/poly}$ 电路的 $i\mathcal{O}$ 的自举. 双密钥切换技术会将一个消息用不同的公钥分别加密, 而在使用时只需要用第一个私钥解密. 安全证明的第一步是换掉第二个公钥加密的消息. 由于程序只使用了第一个私钥解密, 因此程序中不包含第二个私钥的信息, 即使 $i\mathcal{O}$ 无法保护任何信息, 敌手在没有密钥的情况下也无法察觉密文发生了变化, 即这一步可以由公钥加密的安全性来证明. 安全证明的第二步是修改程序, 让程序去解密并使用第二个消息, 此时需要让程序在使用这两个消息时的功能是一致的, 这一步可以通过 $i\mathcal{O}$ 的安全性来证明. 安全证明的第三和第四步则是分别把第一个公钥加密的消息换掉以及把程序再改回解密并使用第一个消息. 通过这四个步骤, 就能成功地将消息一替换为消息二从而实现 IND-based 安全性的证明. 这一技术经过进一步发展、演变, 也出现了一些为程序设计两个分支的技术, 其中一个分支用于正常使用, 另一个分支则专门用于安全性证明.

2014 年, Sahai 和 Waters [257] 提出了穿孔技术, 其核心组件是可穿孔伪随机函数 (puncturable pseudorandom function, PPRF). GGM (Goldreich-Goldwasser-Micali) 树 [258] 是密码学中的经典构造,

可由 PRG 构造 PRF, 而文献 [257] 对 GGM 树稍加改造, 就将 PRF 变为 PPRF. PPRF 主要有下面两个特征.

• **功能一致性.** 在穿孔处之外的任何地方, PPRF 和相应 PRF 的功能是一致的.

• **伪随机性.** PPRF 在穿孔处的值是伪随机的, 即任何敌手就算得到 PPRF (穿孔密钥), 也无法预测相应 PRF 在穿孔处的值.

安全性证明的第一步通常会将 PRF 换成 PPRF, 由于二者的功能是一致的 (除了在穿孔处之外, 但可以通过硬编码的方式添加一个条件语句, 使得程序在穿孔处的值也和 PRF 一致), 因此这一步可以通过 $i\mathcal{O}$ 的安全性来证明. 安全性证明的第二步则是将 PPRF 穿孔处的值换为一个服从均匀分布的随机数, 这可以通过 PPRF 的伪随机性来证明. 通过这两个步骤, PRF 在某个输入上的取值就换成了一个服从均匀分布的随机数, 而均匀分布的随机数异或任意消息仍服从均匀分布, 从而完成了 SIM-based 安全性证明.

通过这些技术, 研究人员可用 $i\mathcal{O}$ (联合基础密码学组件, 如单向函数) 构造几乎所有的密码学应用. 例如文献 [257] 构造了可否认加密, 文献 [259] 构造了无需使用自举技术的全同态加密, 文献 [260] 构造了数字水印. $i\mathcal{O}$ 因此被称作是 "密码学完备" 的. 除了构造密码学应用, $i\mathcal{O}$ 对计算理论、博弈论等领域也产生了一定影响, 例如文献 [261] 将寻找纳什均衡的难度规约到攻破 $i\mathcal{O}$ 的难度, 换言之, 只要 $i\mathcal{O}$ 存在, 寻找纳什均衡就是困难的.

6.3 不可区分混淆的构造

研究人员在基于 $i\mathcal{O}$ 构造密码学组件方面已经取得了一些成果, 但 $i\mathcal{O}$ 的高效且可证明安全构造仍然是一个持续困扰着学界的开放问题, 要想实现这个终极目标还有很长的路要走. 第一个 $i\mathcal{O}$ 构造于 2013 年出现, 随后有了许多其他候选方案. 这些方案均无法兼顾安全与效率, 例如文献 [262] 中的方案效率最高, 但其安全性又是最差的一个. 下面分别介绍这些候选方案. 表 4 从可证明安全性、抗量子攻击、密码分析、性能等角度对几类典型的不可区分混淆构造进行了具体对比.

6.3.1 基于多线性映射

离散对数假设是密码学中一类非常经典的假设, 离散对数群有非常好的加法同态性, 从而可以安全地进行加法计算, 但无法安全地计算乘法. 随着椭圆曲线群及双线性映射 (bilinear map, pairing) 的提出, 研究人员构造了更高级的公钥密码原语, 如基于身份加密 (identity-based encryption, IBE)、基于属性加密 (attribute-based encryption, ABE) 等. 双线性映射既有加法同态性, 还允许安全地进行一次乘法计算.

$i\mathcal{O}$ 是一个如此强大的密码学原语, 以至于用 pairing 也难以构造出来. 因此, 研究人员想到了比 pairing 更强大的工具, 即多线性映射 (multilinear map). 多线性映射除了具有加法同态的性质以外, 还允许安全地进行多项式次乘法运算. 不幸的是, 在双线性映射被发现后, 人们发现其甚至难以推广到三线性映射. 在解决如何基于多线性映射构造 $i\mathcal{O}$ 这个问题之前, 首先需要一个多线性映射的候选方案. 2013 年, Garg 等 [263] 提出了第一个基于理想格的多线性映射候选方案. 同年, Coron 等 [264] 提出了基于中国剩余定理的多线性映射候选方案. 2015 年, Gentry 等 [265] 又提出了基于标准格的多线性映射候选方案. 这些多线性映射比标准的多线性映射更为特殊一些.

首先, 当我们表示群上元素的时候, 我们会为这个表示附带上一个噪声. 受此影响, 这类多线性映射候选方案还会提供一个算法检查元素是否为零. 如果两个元素相等但它们附带的噪声不同, 没有这样一个算法的话我们将无法判断两个元素是否相等. 而在传统的双线性映射中, 两个相等的元素一定

表 4 不可区分混淆各方案安全与效率对比

Table 4 Security and performance comparisons among $i\mathcal{O}$ constructions

	Based on multilinear map	Based on standard assumptions	Based on lattice problems	Based on matrix randomization
Provable security	Ideal model	Standard model	Nonstandard assumption	×
Cryptanalysis	Extensive	–	Few	Few
Quantum-resistant	×	×	✓	✓
Performance	High	Low	Low	High

是完全一样的.

其次, 这类候选方案有着被称作分级编码 (graded encoding) 的结构. 也就是说, 初始元素都是 1 级的, 一个 M 级的元素和一个 N 级的元素相乘就能得到一个 $M+N$ 级的元素, 只有处于同一级的元素之间可以进行加法运算. 对于一个最多支持 K 次乘法运算的多线性映射而言, $M+N$ 必须不大于 K. 前文提到的判断元素是否为零的算法也仅能工作在 K 级元素上, 而不能用于判断中间级的元素是否为零.

最后, 所有经典的假设在这些多线性映射候选方案上都无法成立, 例如 DLIN (decision linear) 或 SXDH (standard symmetric external Diffie-Hellman) 假设. 因此, 即使我们可以基于标准的多线性映射假设构造 $i\mathcal{O}$, 要是底层的多线性映射使用了这些候选方案的话, 我们仍无法实现可证明安全.

同样是 2013 年, 在提出第一个多线性映射候选方案后, Garg 等 [256] 又基于此提出了第一个 $i\mathcal{O}$ 候选方案, 该方案以分支程序作为计算模型. 这类分支程序的计算能力和 NC^1 的电路等价, 执行这类分支程序可视为将置换群 S_5 上多项式个元素相乘, 而多线性映射恰好能安全地进行多项式次乘法运算! 此外, Garg 等还提出了拼图 (jigsaw puzzle) 思想: 敌手不会诚实地执行计算, 比如它可能会交换两个置换群元素的位置, 而置换群不是阿贝尔群, 交换乘法的顺序可能会导致敌手计算出新的信息[14]. 为了防止出现这样的问题, 可设法给所有的元素附上噪声: 当这些元素按照我们希望的方式相乘时, 这些噪声恰能相互抵消, 否则这些噪声会堆积起来使敌手难以从得到的计算结果中获取有效信息. 这就好比拼图必须按照一定的规则才能严丝合缝地拼成一个有意义的能被辨识的图案. 此后, 又有一些基于多线性映射的候选方案被提出并经历了多轮攻击与修补过程 [266~271]. 目前, 针对文献 [256] 中的初始方案的某些变体方案还没有有效攻击, 不过也没有人能够用数学工具来证明这些变体的安全性 (除非借助于一些非常强的理想化模型).

6.3.2 基于标准假设

将 $i\mathcal{O}$ 规约到相对简单的密码学组件. 由于理想 (多线性映射) 与现实 (双线性映射) 之间存在着巨大鸿沟, 想要立刻构造出可证明安全的 $i\mathcal{O}$ 是十分困难的. 于是有学者尝试将 $i\mathcal{O}$ 规约到其他同样非常强大的密码学组件上. 2015 年, 文献 [272, 273] 指出利用函数加密 (functional encryption, FE) 便可以构造出 $i\mathcal{O}$. 通常的加密方案仅允许通过密钥来对消息进行解密并获取消息的全部信息. 函数加密除了支持用主密钥来解密消息外, 还支持依据主密钥生成与某个函数 f 绑定的密钥, 该密钥可用于对消息的解密, 但无法得到消息的完整信息, 只能得到消息经函数 f 计算的结果. 也就是说, 与传统加密方案要么获取全部信息要么一无所知不同, 函数加密中主密钥持有者可通过生成与各种函数绑定的密

14) 根据 $i\mathcal{O}$ 的定义, 两个功能相同的程序在混淆后功能仍然是一样的. 也就是说, 只有当两个程序都被诚实地计算, 其计算结果才能保证是一样的, 不诚实的计算可能会让敌手发现两个程序之间的差异.

钥来控制不同的用户可获取关于密文的哪方面信息. 这两篇文章也指出, 能够构造 $i\mathcal{O}$ 的 FE 必须满足抗合谋性或紧凑性. 抗合谋性是指 FE 支持生成多个与不同函数绑定的密钥. 紧凑性是指如果 FE 支持的函数 f 的规模至多为 s, 其加密消息时的耗时必须是 s 的亚线性级别.

这两篇文章启发了一些后续研究, 研究人员逐渐认识到了压缩和 $i\mathcal{O}$ 的关系, 甚至可以说 $i\mathcal{O}$ 的研究等价于压缩的研究. $i\mathcal{O}$ 的作用是其针对程序的每一个输入, 都能够安全地运行程序. 事实上, 有研究结果早就能够做到针对程序的某一个输入安全地运行程序, 比如 MPC 中的姚混淆电路 (Yao's garbled circuit) 即可完成隐私计算, 计算的执行者除了计算结果无法得到任何其他有用信息. 如果一个程序总计有 N 种不同的输入, 我们只要生成 N 次混淆电路, 就相当于实现了 $i\mathcal{O}$. 对于输入长为 n 比特的程序, 总共有 2^n 种不同的输入, 生成这么多次混淆电路需要指数时间, 我们必须想办法将其压缩为多项式时间. 研究人员将一次安全的程序运行抽象为随机化编码 (randomized encoding, RE). 随机化编码包含有两个算法: 一个算法是编码, 输入一个电路和一个输入, 输出一串编码; 另一个算法是解码, 输入这串编码, 输出这个电路在该输入上的输出. 任何人如果只看到编码, 都无法得到该输出以外的任何信息. 以混淆电路为例, 编码操作就是将一个电路的所有门的真值表混淆并为某个输入生成相应的标签; 解码则是依据混淆电路和输入标签计算出电路在该输入上的输出. 不支持抗合谋的 FE 也可以视作 RE: 编码就是输入的加密和一个与函数绑定的密钥; 解码则是用该密钥解密并得到函数作用在输入上的输出. 关于如何压缩时间, 研究人员再次想到了 GGM 树, n 层 GGM 树结构可用 PRG 构造出一个支持 2^n 个输入的 PRF, 类似地, 它也能够用 RE 构造出支持 2^n 个输入的 $i\mathcal{O}$. 经过分析发现: 如果随机化编码的耗时相对于电路大小 s 能进行亚线性的压缩, 那么就可以由此构造出 $i\mathcal{O}$; 如果随机化编码的编码长度相对于电路大小 s 能进行亚线性的压缩, 那么加上 LWE 假设就可以构造出 $i\mathcal{O}$ [274].

将 $i\mathcal{O}$ 规约到常数 ($\geqslant 3$) 层多线性映射. 2016~2018 年, Lin 等 [275~278] 发现了 RE 和具有特殊结构的 PRG 在降低多线性映射层数上的作用, 并成功将构造 $i\mathcal{O}$ 所需的多项式层多线性映射降低为常数层多线性映射. RE 可对一次电路运算进行编码, 这个编码过程往往会降低电路的复杂程度. 例如, 由于 NC^1 上存在 PRF, 因此混淆电路的复杂程度也可以控制在 NC^1, 这意味着我们可以把任意 $\mathsf{P/poly}$ 的运算编码为 NC^1 的运算. 另一方面, 他们注意到了文献 [279] 中提出的 RE 能够将 NC^1 的电路编码为 NC^0 的电路. 通过使用这两种 RE, 他们成功证明了: 简单的 FE (即只支持 NC^0 电路的 FE) 便足以构造出 $i\mathcal{O}$. 为了达到亚线性压缩的效果, 他们使用了 PRG 来压缩[15], 且此时 PRG 的伸展率 (strech) 必须是超线性的[16]. 由于 NC^0 上满足此条件的 PRG 的 locality 至少为 5[17], 而文献 [279] 中的 RE 方案的每个输出位至少关联 3 个伪随机数位 (每个伪随机数位都是 PRG 的一个输出位) 和一个电路的输入位[18], 因此总共需要关联 $3\times 5+1$ 个输入位. 实际上, 由于安全证明的需要[19], 还需要有一个输入位指明当前执行的是哪个分支, 于是文献 [276] 最终将多线性映射的层数降低到了 17. 后来, 他们又改进 PRG 的结构, 并通过预处理提前计算很多中间变量, 使单个输出位依赖的输入位的数量进一步降低. 例如, 计算 $abcde$ 本身需要一个 5 层的多线性映射, 但如果我们提前算出了 abc 和 de, 那就只需再将这两项相乘, 即只需双线性映射就够了. 最终, 文献 [278] 将多线性映射的层数降低到了 3 层.

将 $i\mathcal{O}$ 规约到双线性映射. 不过目前的成果仍然不能令人满意, 即便是三线性映射也没有十分可靠的候选方案. 2021 年, Jain, Lin 和 Sahai [280] 基于双线性映射 (更具体地说, SXDH 或 DLIN 假设) 以及 LPN, LWE 和 NC^0 上的 PRG 这 4 个假设构造出了可证明安全的 $i\mathcal{O}$. 虽然该构造十分复杂低效,

15) 只需要用较短的种子就可以生成 RE 中需要使用的较长伪随机数.
16) 如果 PRG 的种子是 n 比特, 其输出长度至少达到 $n^{1+\varepsilon}$.
17) PRG 的每个输出位至少关联 5 个输入位.
18) RE 是将电路和电路的输入一起编码的, 且这是一个随机算法.
19) 这里使用了双分支的技术, 一个分支用于正常使用, 另一个分支用于安全证明.

但是大大加强了我们对于 $i\mathcal{O}$ 存在的信心, 在理论层面具有很高价值. 他们此前就注意到可将同态加密 (homomorphic encryption, HE) 引入 $i\mathcal{O}$ 的构造中 [281, 282], 目的是利用 HE 和只能支持简单计算的 FE 构造能支持所有 NC^0 电路的 FE, 为此需要解决: 一是基于双线性映射的 FE 只能安全地进行一次乘法, 二是 HE 的同态计算过程非常复杂, 解密操作也不够简单, 远非一次乘法计算能够处理的. 对此, 他们将非常复杂的同态计算和不够简单的解密操作分开处理: 基于密文的同态操作过程可公开进行, 无需利用 FE 保护; 目前 HE 方案的解密是将密文和密钥做一次内积, 然后进行模数操作, 如果放弃模数操作, 则解密过程就只剩下一次乘法! 为了实现这个想法, 他们首先构造部分隐藏的 FE (partial hiding FE, PHFE). PHFE 加密的消息有公开部分和秘密部分两个部分, 对公开部分可进行任意常数次乘法运算, 也允许它们与秘密部分进行计算. 但对于秘密部分, 只允许它们内部计算一次乘法. 这种思想最早见诸于文献 [283], 这篇文章在文献 [284] 的基础上通过引入 HE 成功将 ABE 升级为谓词加密 (predicate encryption, PE), 其中就用到了将待加密的消息分为公开部分和秘密部分的思想.

至此, 还剩下一个问题: 由于解密时没有进行模数操作, 同态加密引入的噪声便会经过解密泄露出来, 而这个泄露对于安全性将是致命的. 文献 [280] 设计了一个有特殊结构的 PRG, 该 PRG 的种子也分为公开部分和秘密部分, 且 PRG 的输出位关于公开部分可以进行常数次运算, 关于秘密部分只能进行一次乘法计算. 这样, 通过给解密结果附上 PRG 的输出, 该输出作为大噪声会掩盖同态运算带来的小噪声, 从而保护 FE 的安全性. 不过, PRG 输出的噪声还不够显著, 只能部分掩盖同态运算产生的小噪声, 好在他们同时提出了一套强化安全性的方法 [285], 可以将敌手攻破一个 FE 方案的概率从多项式分之一降为可忽略的. 近期, Jain, Lin 和 Sahai [286] 三人又进一步改进了之前的工作, 将原本的 4 个假设中的 LWE 去掉.

6.3.3 基于格上困难问题

基于标准假设的方案依赖双线性映射, 从而无法抵御量子计算机的攻击. 有一些工作致力于基于格上困难问题构造 $i\mathcal{O}$. 与 6.3.2 小节介绍的利用 HE 构造 $i\mathcal{O}$ 的想法十分接近, 文献 [287] 提出使用 spilt-FHE 构造 $i\mathcal{O}$, 并利用密钥切换技术解决噪声掩盖问题[20]. 他们找到了一个密文空间接近全空间[21]的线性同态加密方案 (linear HE, LHE) 以及一个全同态加密方案. 线性同态加密方案只支持对线性函数进行同态运算, 但具有压缩的性质. 他们首先将消息 (带有小噪声) 对应的 FHE 的加密切换为 LHE 的加密以利用 LHE 的压缩性, 此时需要一个 LHE 加密下的大噪声去掩盖小噪声. 为此, 他们使用随机预言机 (ramdom oracle, RO) 得到了一个随机数, 并将其作为 LHE 的密文, 这个密文对应的明文也是个随机数. 但是这个随机数太大了, 不仅会掩盖小噪声, 甚至可能掩盖整个消息. 为了控制随机数的大小, 需要对其进行模数操作, 但 LHE 只支持线性操作. 于是, 他们将 LHE 的密文切换为 FHE 的密文并进行模数操作, 再将其切换回 LHE 的密文. 这样, 将带小噪声的消息的密文和小随机数的密文同态相加, 看上去小噪声就能被比它大得多又比消息小得多的小随机数给掩盖. 但是, 密钥切换需要使用加密的密钥, 这就涉及同态加密中一个经典的假设, 即循环安全假设: 如果将 FHE 的密钥用 LHE 加密并公开, 再将 LHE 的密钥用 FHE 加密并公开, 那么 FHE 和 LHE 各自还能安全吗? 这篇文章[22]没能给出相关的安全规约. 2021 年, 文献 [288] 找到了一个后量子安全的 LHE 方案, 同时将方案 [287] 中的 RO 换为公共随机串 (common reference string, CRS), 并将其安全性规约到了一个循环假设上. 但是, 这个循环假设强于 FHE 中常见的循环假设, 因此有可能他们提出的循环假设是不成立

20) 该技术最早可见于 Leveled-FHE, 即同态加密支持将密文的密钥从 sk_1 换为 sk_2.
21) 即一个随机数无法被解密的概率是可忽略的.
22) 文章中找的 LHE 方案是基于 DCR 假设的, 因此不能实现后量子安全.

的, 而 FHE 中常见的循环假设是成立的. 文献 [289] 给出的反例证明了这一点.

受这两篇文章的启发, 文献 [290] 进一步简化了 spilt-FHE 的定义, 指出函数编码 (functional encoding) 便足以构造 $i\mathcal{O}$, 也就是说, 加密这个属性并不是必须的. 他们使用同态承诺 (homomorphic commitment) 进行构造, 一个消息的承诺可在事实上对应于其他消息, 只是敌手并没有能力将其打开为其他消息. 类似地, 消息的承诺可以视为一个 LWE 的样本, 在打开时也会带上小噪声, 为了安全性我们也需要想办法掩盖这个小噪声. 为此, 他们提出了不经意 LWE 采样 (oblivious LWE sampling), 目的是产生新的 LWE 样本去掩盖原有样本带的小噪声. 为实现这一目的, 新的 LWE 样本对应的大噪声不能被敌手知道 (否则敌手只要将承诺打开的结果减去已知的大噪声就能得到小噪声), 这也正是不经意这个词的意思, 即噪声不是利用 $\boldsymbol{A}s+e$ 计算得到的, 敌手计算得到 LWE 的样本后仍然无法得知 s 的值. 他们构造了一种启发式的不经意 LWE 采样器, 并将安全性规约到了一个与循环假设有些相像的假设上. 不幸的是, 文献 [289] 针对这一假设同样给出了反例. 2021 年, 文献 [291] 进一步改进了文献 [290], 基于紧凑 LWE 采样 (succinct LWE sampling) 这一更弱的密码学原语构造出了 $i\mathcal{O}$, 同时也给出了一个紧凑 LWE 采样的候选方案, 该方案的安全性与求解多项式等式系统的困难性相关联.

有一些工作基于带噪声的线性函数加密 (noisy linear functional encryption) 来构造 $i\mathcal{O}$ [292, 293], 其安全性可规约到 NTRU 的变体上. 总的来看, 目前这类方案虽然有安全证明, 但是安全性都还无法规约到标准假设上.

6.3.4 基于矩阵随机化

最后, 我们还有一类特殊的候选方案. Bartusek 等 [262] 于 2020 年提出了用矩阵随机化来构造 $i\mathcal{O}$. 这类方案使用到了仿射行列式程序 (affine determinant program, ADP) 计算模型[23]. 一个 n 比特输入 1 比特输出的 ADP 包含 $n+1$ 个矩阵, 运行这个程序的方式是按规则挑选若干个矩阵相加 (仿射), 并计算结果矩阵的行列式, 根据行列式的结果决定输出是 0 还是 1 (例如奇数则输出 1, 偶数则输出 0). 这类方案和基于多线性映射的方案类似, 只能直接用于 NC^1 的电路[24], 同时也可通过自举技术来支持所有 $\mathsf{P/poly}$ 的电路. 为了混淆 ADP, 他们提出了 4 种不同的矩阵随机化变换, 这些变换都不会改变行列式的值, 从而不会改变程序的功能. 将这 4 种矩阵随机化变换合理地组合使用可起到 $i\mathcal{O}$ 的效果. 这 4 个变换中, 有的用来掩盖矩阵除了行列式和秩以外的信息, 有的用来保证敌手只能按规则挑选若干矩阵相加, 有的能够将矩阵都变为满秩的. 剩下一个最为特殊、最 Ad-hoc 的随机化操作, 称为随机局部替换 (random local substitution, RLS). 引入 RLS 的原因是作者发现如果只将其他 3 种随机化变换进行组合, 敌手计算行列式模 4 可提取出矩阵的校验信息, 进而攻破这类方案. 他们寄希望于通过 RLS 来给矩阵引入一定的随机性, 从而使敌手无法了解矩阵的全部信息, 这样敌手即使得到了矩阵的校验信息也难以利用. 然而, 文献 [294] 发现, Bartusek 等的 RLS 引入的随机性不够, 通过对他们提出的 "模 4 攻击" 稍加修改, 便可将原本的方案攻破, 同时该文也尝试提出改进版的 RLS 方案. 总之, 目前对这类方案的密码分析工作还很少, 现有的分析方法暂时也还无法帮助我们很好地理解这类方案的安全性.

7 总结与展望

近年来, 高等密码算法和协议在以隐私计算为代表的诸多新兴行业得到广泛应用并受到学术界和

23) ADP 可视为分支程序的一种变体.

24) 事实上, 这类方案中的分支程序与基于多线性映射方案的分支程序不完全相同, 这里的分支程序的计算能力还要更强些, 能表示所有对数空间计算.

工业界的普遍关注. 本综述重点讨论的安全多方计算、全同态加密、零知识证明、不可区分混淆等相关隐私保护密码研究均同时具有理论意义和现实应用价值. 近年来这些研究方向在包括美密、欧密在内的权威密码学术会议和 CCS, S&P, NDSS, USENIX 等顶级安全会议上经常有新成果展示. 安全多方计算、同态加密和零知识证明等算法和协议甚至已经在某些场景中得到商业化部署, 为工业界带来了巨大商业空间. 这反过来又进一步激发了学术界对它们的研究热情. 学术界和工业界的此类双向正反馈迭代式推进了高等密码算法的发展和应用.

不过, 这些密码算法的普及和落地仍然面临巨大挑战. 一方面, 现实场景比较复杂, 此类算法普遍性能不高, 很难满足数据量大、时延要求高的应用场景. 另一方面, 算法和协议的多样性与复杂性又使得难以对它们进行标准化, 这也是限制高等密码算法与协议广泛应用的最重要因素之一. 学术界、工业界和相关政府部门需要协作完成大量工作以推动这些算法与协议应用真正落地. 综合当前研究现状来看, 我们认为安全多方计算、全同态加密、零知识证明, 以及不可区分混淆在后续研究中需要解决的问题包括以下几个方面.

7.1 安全多方计算

在学术界和工业界的共同努力下, 安全多方计算协议的实际效率被不断提高, 已能在一些应用中实际部署. 但为了满足所有/大部分应用的功能、性能和安全需求, 安全多方计算仍有以下问题需要进一步开展研究. 一是安全多方计算协议的通信效率提高问题. 与计算资源比较, 通信资源不易更新, 在相当一段时间内通信带宽相对固定. 目前, 通信开销仍然是制约安全多方计算在实际应用中部署的重要因素之一, 特别是电路规模达到百亿门级别. 二是提高安全多方计算协议的扩展性, 支持更多参与方. 部分安全多方计算应用场景需要成百上千数量的用户参与, 从而要求安全多方计算协议能在这些应用场景中提供高的扩展性, 即在支持数百、上千参与方数量时, 通信和计算效率仍然满足实际应用需求. 目前, 已被实现的安全多方计算协议有效支持上百参与方数量, 但还未高效扩展到上千参与方数量. 三是安全多方计算与新型计算框架的结合问题. 以区块链为例, 区块链和安全多方计算均具有去中心化特点, 区块链的可验证、不可篡改等性质为安全多方计算解决了数据源认证问题, 安全多方计算的隐私保护性质能有效保护链上数据的隐私并支持链上数据计算.

7.2 全同态加密

在全同态加密研究中, 我们需要重点关注高效的全同态加密算法设计及硬件加速研究. 自 2009 年首个全同态加密算法提出之后, 提升全同态加密算法的性能一直是该领域专注的研究方向. 虽然理论上有多种不同的构造, 但考虑到具体的运行性能, 全同态加密算法主要集中于在 Ring-LWE 或 Torus-LWE 格上困难假设下的构造, 其中典型的算法包括: BFV, BGV, CKKS, TFHE 等.

BFV 和 BGV 均是基于 Ring-LWE 困难假设的全同态加密算法, 将明文数据编码到多项式环上, 利用多项式环的代数性质实现单指令多数据的并行操作 (SIMD). 因此, 该类算法较为适用于处理整数数据的场景, 比如密文检索、密文匹配、私有集合交集等应用, 但对于浮点数的处理则效率很低. CKKS 在和数值计算相关的场景中使用较为普遍, 比如统计数据计算、机器学习等. 这几种 FHE 算法均需要自举 (bootstrapping) 操作以实现任意电路的同态计算, 但该操作计算开销相当大, 难以满足实际应用需求.

TFHE 将明文空间限制在比特中 (即只加密 0 和 1), 通过 Torus 的特性将比特编码到实数多项式上. 虽然只能针对比特进行操作, 但是 TFHE 采用了特殊的自举方法, 使得对于比特上的每次自举操作能够达到毫秒级别. 从适用范围上看, TFHE 适用于布尔电路的同态运算, 但对于算术电路的计算开

销比 BFV, BGV, CKKS 高得多.

一个很自然的想法就是研究不同类型全同态加密算法的密文转换技术, 通过特殊的设计技巧可以在 BFV/BGV, CKKS, TFHE 的密文之间进行相互转化, 这样我们就可以将上述各类全同态加密算法的优势相结合以实现优势综合利用. 在针对整数数据的场景使用 BFV/BGV, 在浮点数的场景使用 CKKS, 在比特数据的场景中使用 TFHE. 另外, 全同态加密算法的硬件加速技术研究和全同态加密算法软硬件实现的内存优化方法研究对于全同态加密的具体应用场景也均具有现实意义.

7.3 零知识证明

针对通用 NP 语言均可构造零知识证明. 但我们要将零知识证明协议落地应用于具体场景中时则会涉及到性能问题, 比如有的应用中可能在要求低存储的同时网络实时通信开销又会比较高, 此时就需要用到简洁非交互零知识证明. 当前, 简洁非交互零知识证明研究还需要解决下面几个问题. 一方面, 我们需要从性能方面来考虑协议优化设计问题, 这包括进一步降低简洁非交互零知识的证明、通信和验证复杂度, 如何利用批量验证的方式降低验证开销等. 另一方面, 我们需要考虑协议的安全性问题, 比如如何基于标准假设来构造简洁非交互零知识证明, 是否可以利用更为通用的难题假设构造 zk-SNARK 及系列可更新零知识证明等. 而在某些特殊需求环境的情况下, 比如大量重复结构的问题和固定的问题证明 (类似于签名算法中使用的零知识证明) 时, 这类零知识证明可以具有与通用的简洁非交互零知识证明的不同的结构来适应需求并进一步提升效率, 寻找并对这样的场景中的协议进行重新构造也是零知识证明的一个发展方向.

7.4 不可区分混淆

目前学界在根据 $i\mathcal{O}$ 来构造密码学组件的研究方面已经取得了已有很多成果. 比如, Sahai 等 [257] 在 2014 年通过 $i\mathcal{O}$ 和一些基础的密码学假设 (例如单向函数) 构造了公钥加密、数字签名、可否认加密等. $i\mathcal{O}$ 不仅能用于构造我们已经熟知的经典密码学组件, 更可以作为桥梁构造大量功能十分强大的 “新” 密码学组件 (甚至有一部分组件目前只能基于 $i\mathcal{O}$ 构造), 但是, 这些构造想要转化为现实应用都依赖于一个高效的 $i\mathcal{O}$.

在构造 $i\mathcal{O}$ 方面, 虽然 2021 年 Jain 等 [280] 首次提出了基于标准假设的 $i\mathcal{O}$ 构造, 但是这一构造的效率很低, 且不满足后量子安全. 因此 2020 年来学界的一个重要研究方向在于构造后量子安全的更高效率的 $i\mathcal{O}$. 然而, 种种迹象表明, 仅仅依靠标准假设 (如 LWE) 难以构造出可证明安全的 $i\mathcal{O}$, 因此这类尝试往往需要引入新的假设, 而针对这些新的假设就需要相应的密码分析来帮助我们认识这些假设的困难性. 总之想要得到高效的 $i\mathcal{O}$ 从而使其应用于具体场景我们仍有很长的路要走.

参考文献

1 Li F H, Li H, Jia Y, et al. Privacy computing: concept, connotation and its research trend. J Commun, 2016, 37: 1

2 McMahan B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 2017. 1273–1282

3 Yang Q, Liu Y, Chen T, et al. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol, 2019, 10: 1–19

4 Dwork C. Differential privacy. In: Encyclopedia of Cryptography and Security. Berlin: Springer, 2006. 1–12

5 Dwork C, Roth A. The algorithmic foundations of differential privacy. FNT Theor Comput Sci, 2014, 9: 211–407

6 Dwork C, Kenthapadi K, McSherry F, et al. Our data, ourselves: privacy via distributed noise generation. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2006. 486–503

7 Sabt M, Achemlal M, Bouabdallah A. Trusted execution environment: what it is, and what it is not. In: Proceedings of IEEE Trustcom/BigDataSE/ISPA, 2015. 57–64

8 Arm.com. Globalplatform based trusted execution environment and ready. 2020. https://community.arm.com/cfs-file/_key/telligent-evolution-components-attachments/01-2142-00-00-00-00-51-36/Globalplatform-based-trusted-execution-environment-and-R.pdf

9 Albrecht J P. How the GDPR will change the world. Eur Data Protection Law Rev, 2016, 2: 287–289

10 Androulaki E, Barger A, Bortnikov V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the 13th EuroSys Conference, 2018

11 Wood G. Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014, 151: 1–32

12 Yao A C C. How to generate and exchange secrets. In: Proceedings of 27th Annual Symposium on Foundations of Computer Science, 1986. 162–167

13 Goldreich O, Micali S, Wigderson A. How to play any mental game or a completeness theorem for protocols with honest majority. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987. 218–229

14 Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 1988. 1–10

15 Chaum D, Crépeau C, Damgård I. Multiparty unconditionally secure protocols. In: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, 1988. 11–19

16 Rabin T, Ben-Or M. Verifiable secret sharing and multiparty protocols with honest majority. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, 1989. 73–85

17 Beaver D, Micali S, Rogaway P. The round complexity of secure protocols. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, 1990. 503–513

18 Nikolaenko V, Weinsberg U, Ioannidis S, et al. Privacy-preserving ridge regression on hundreds of millions of records. In: Proceedings of IEEE Symposium on Security and Privacy, 2013. 334–348

19 Mohassel P, Zhang Y P. SecureML: a system for scalable privacy-preserving machine learning. In: Proceedings of IEEE Symposium on Security and Privacy, 2017. 19–38

20 Liu J, Juuti M, Lu Y, et al. Oblivious neural network predictions via MiniONN transformations. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 619–631

21 Juvekar C, Vaikuntanathan V, Chandrakasan A. Gazelle: a low latency framework for secure neural network inference. In: Proceedings of the 27th USENIX Conference on Security Symposium, 2018. 1651–1669

22 Mohassel P, Rindal P. ABY^3: a mixed protocol framework for machine learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 35–52

23 Zheng W T, Popa R A, Gonzalez J E, et al. Helen: maliciously secure coopetitive learning for linear models. In: Proceedings of IEEE Symposium on Security and Privacy, 2019. 724–738

24 Agrawal N, Shamsabadi A S, Kusner M J, et al. QUOTIENT: two-party secure neural network training and prediction. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 1231–1247

25 Schoppmann P, Gascón A, Raykova M, et al. Make some ROOM for the zeros: data sparsity in secure distributed machine learning. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 1335–1350

26 Riazi M S, Samragh M, Chen H, et al. XONN: XNOR-based oblivious deep neural network inference. In: Proceedings of the 28th USENIX Conference on Security Symposium, 2019. 1501–1518

27 Chandran N, Gupta D, Rastogi A, et al. EzPC: programmable and efficient secure two-party computation for machine learning. In: Proceedings of IEEE European Symposium on Security and Privacy, 2019. 496–511

28 Wagh S, Gupta D, Chandran N. SecureNN: 3-party secure computation for neural network training. Proc Priv Enhancing Technol, 2019, 2019: 26–49

29 Kumar N, Rathee M, Chandran N, et al. Cryptflow: secure Tensorflow inference. In: Proceedings of IEEE Symposium on Security and Privacy, 2020. 336–353

30 Rathee D, Rathee M, Kumar N, et al. CrypTFlow2: practical 2-party secure inference. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2020. 325–342

31 Mishra P, Lehmkuhl R, Srinivasan A, et al. DELPHI: a cryptographic inference service for neural networks.

In: Proceedings of the 29th USENIX Conference on Security Symposium, 2020. 2505–2522
32 Patra A, Suresh A. BLAZE: blazing fast privacy-preserving machine learning. 2020. ArXiv:2005.09042
33 Chaudhari H, Rachuri R, Suresh A. Trident: efficient 4PC framework for privacy preserving machine learning. In: Proceedings of the 26th Annual Network and Distributed System Security Symposium, 2020
34 Dalskov A, Escudero D, Keller M. Secure evaluation of quantized neural networks. Proc Priv Enhancing Technol, 2020, 2020: 355–375
35 Byali M, Chaudhari H, Patra A, et al. FLASH: fast and robust framework for privacy-preserving machine learning. Proc Priv Enhancing Technol, 2020, 2020: 459–480
36 Rathee D, Rathee M, Goli R K K, et al. SIRNN: a math library for secure RNN inference. In: Proceedings of IEEE Symposium on Security and Privacy, 2021. 1003–1020
37 Patra A, Schneider T, Suresh A, et al. ABY2.0: improved mixed-protocol secure two-party computation. In: Proceedings of USENIX Security Symposium, 2021. 2165–2182
38 Tan S J, Knott B, Tian Y, et al. CryptGPU: fast privacy-preserving machine learning on the GPU. In: Proceedings of IEEE Symposium on Security and Privacy, 2021. 1021–1038
39 Hussain S U, Javaheripi M, Samragh M, et al. COINN: Crypto/ML codesign for oblivious inference via neural networks. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021
40 Lehmkuhl R, Mishra P, Srinivasan A, et al. MUSE: secure inference resilient to malicious clients. In: Proceedings of the 30th USENIX Security Symposium, 2021. 2201–2218
41 Koti N, Pancholi M, Patra A, et al. SWIFT: super-fast and robust privacy-preserving machine learning. In: Proceedings of the 30th USENIX Security Symposium, 2021
42 Wagh S, Tople S, Benhamouda F, et al. FALCON: honest-majority maliciously secure framework for private deep learning. Proc Priv Enhancing Technol, 2021, 2021: 188–208
43 Chandran N, Gupta D, Obbattu S L B, et al. SIMC: ML inference secure against malicious clients at semi-honest cost. In: Proceedings of the 31st USENIX Security Symposium, 2022
44 Huang Z C, Lu W J, Hong C, et al. Cheetah: lean and fast secure two-party deep neural network inference. 2022. https://eprint.iacr.org/2022/207.pdf
45 Rathee D, Bhattacharya A, Sharma R, et al. SecFloat: accurate floating-point meets secure 2-party computation. In: Proceedings of IEEE Symposium on Security and Privacy (SP), 2022
46 Han K, Jeong J, Sohn J H, et al. Efficient privacy preserving logistic regression inference and training. 2020. https://eprint.iacr.org/2020/1396.pdf
47 Fereidooni H, Marchal S, Miettinen M, et al. SAFELearn: secure aggregation for private federated learning. In: Proceedings of IEEE Security and Privacy Workshops (SPW), 2021
48 Brunetta C, Tsaloli G, Liang B, et al. Non-interactive, secure verifiable aggregation for decentralized, privacy-preserving learning. In: Proceedings of the 26th Australasian Conference, 2021
49 Zheng W T, Deng R, Chen W K, et al. Cerebro: a platform for multi-party cryptographic collaborative learning. 2021. https://eprint.iacr.org/2021/759.pdf
50 Jha S, Kruger L, Shmatikov V. Towards practical privacy for genomic computation. In: Proceedings of IEEE Symposium on Security and Privacy, 2008. 216–230
51 Jagadeesh K A, Wu D J, Birgmeier J A, et al. Deriving genomic diagnoses without revealing patient genomes. Science, 2017, 357: 692–695
52 Cho H, Wu D J, Berger B. Secure genome-wide association analysis using multiparty computation. Nat Biotechnol, 2018, 36: 547–551
53 Archer D W, Bogdanov D, Lindell Y, et al. From keys to databases-real-world applications of secure multi-party computation. Comput J, 2018, 61: 1749–1771
54 Ishai Y, Kushilevitz E, Ostrovsky R, et al. Zero-knowledge from secure multiparty computation. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, 2007. 21–30
55 Giacomelli I, Madsen J, Orlandi C. ZkBoo: faster zero-knowledge for Boolean circuits. In: Proceedings of the 25th USENIX Conference on Security Symposium, 2016. 1069–1083
56 Chase M, Derler D, Goldfeder S, et al. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 1825–1842

57 Ames S, Hazay C, Ishai Y, et al. Ligero: lightweight sublinear arguments without a trusted setup. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 2087–2104

58 Katz J, Kolesnikov V, Wang X. Improved non-interactive zero knowledge with applications to post-quantum signatures. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 525–537

59 Bhadauria R, Fang Z Y, Hazay C, et al. Ligero++: a new optimized sublinear IOP. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2020. 2025–2038

60 Baum C, Nof A. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In: Proceedings of IACR International Conference on Public-Key Cryptography, 2020. 495–526

61 Gvili Y, Scheffler S, Varia M. BooLigero: improved sublinear zero knowledge proofs for Boolean circuits. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2021

62 Guilhem C D S, Orsini E, Tanguy T. Limbo: efficient zero-knowledge MPCitH-based arguments. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021

63 Baum C, Guilhem C D S, Kales D, et al. Banquet: short and fast signatures from AES. In: Proceedings of IACR International Conference on Public-Key Cryptography, 2021. 266–297

64 Jawurek M, Kerschbaum F, Orlandi C. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In: Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, 2013. 955–966

65 Frederiksen T K, Nielsen J B, Orlandi C. Privacy-free garbled circuits with applications to efficient zero-knowledge. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015. 191–219

66 Kondi Y, Patra A. Privacy-free garbled circuits for formulas: size zero and information-theoretic. In: Proceedings of Annual International Cryptology Conference, 2017. 188–222

67 Heath D, Kolesnikov V. Stacked garbling for disjunctive zero-knowledge proofs. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 569–598

68 Dittmer S, Ishai Y, Ostrovsky R. Line-point zero knowledge and its applications. 2020. https://eprint.iacr.org/2020/1446.pdf

69 Baum C, Malozemoff A J, Rosen M B, et al. Mac'n'cheese: zero-knowledge proofs for Boolean and arithmetic circuits with nested disjunctions. In: Proceedings of the 41st Annual International Cryptology Conference, 2021. 92–122

70 Weng C K, Yang K, Xie X, et al. Mystique: efficient conversions for zero-knowledge proofs with applications to machine learning. In: Proceedings of USENIX Security Symposium, 2021. 501–518

71 Baum C, Braun L, Munch-Hansen A, et al. Appenzeller to Brie: efficient zero-knowledge proofs for mixed-mode arithmetic and $\mathbb{Z}_{2^k}$. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021

72 Pinkas B, Schneider T, Weinert C, et al. Efficient circuit-based PSI via cuckoo hashing. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2018. 125–157

73 Pinkas B, Schneider T, Tkachenko O, et al. Efficient circuit-based PSI with linear communication. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019. 122–153

74 Pinkas B, Rosulek M, Trieu N, et al. SpOT-light: lightweight private set intersection from sparse OT extension. In: Proceedings of Annual International Cryptology Conference, 2019. 401–431

75 Pinkas B, Rosulek M, Trieu N, et al. PSI from PaXoS: fast, malicious private set intersection. In: Proceedings of the 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 739–767

76 Rindal P, Schoppmann P. VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE. In: Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 901–930

77 Garimella G, Pinkas B, Rosulek M, et al. Oblivious key-value stores and amplification for private set intersection. In: Proceedings of Annual International Cryptology Conference, 2021. 395–425

78 Chandran N, Gupta D, Shah A. Circuit-PSI with linear complexity via relaxed batch OPPRF. 2021. https://eprint.iacr.org/2021/034.pdf

79 Chandran N, Dasgupta N, Gupta D, et al. Efficient linear multiparty PSI and extensions to circuit/quorum PSI. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021

80 Demmler D, Schneider T, Zohner M. ABY — a framework for efficient mixed-protocol secure two-party computation.

In: Proceedings of Network and Distributed System Security Symposium, 2015
81 Wang X, Malozemoff A J, Katz J. EMP-toolkit: efficient multiparty computation toolkit. 2016. https://github.com/emp-toolkit
82 Institute A. FRESCO — a framework for efficient secure computation. 2020. https://github.com/aicis/fresco
83 Multiparty.org. Javascript implementation of federated functionalities. 2020. https://github.com/multiparty/jiff
84 Data61. Mp-spdz. 2019. https://github.com/data61/MP-SPDZ
85 Schoenmakers B. MPYC: secure multiparty computation in Python. 2020. https://github.com/lschoe/mpyc
86 Aly A, Keller M, Orsini E, et al. Scale-Mamba v1.14: documentation, 2021. https://github.com/KULeuven-COSIC/SCALE-MAMBA
87 Bogdanov D, Laur S, Willemson J. Sharemind: a framework for fast privacy-preserving computations. In: Proceedings of European Symposium on Research in Computer Security, 2008. 192–206
88 Songhori E M, Hussain S U, Sadeghi A-R, et al. TinyGarble: highly compressed and scalable sequential garbled circuits. In: Proceedings of IEEE Symposium on Security and Privacy, 2015. 411–428
89 Lindell Y, Pinkas B, Smart N P, et al. Efficient constant round multi-party computation combining BMR and SPDZ. In: Proceedings of Annual Cryptology Conference, 2015. 319–338
90 Lindell Y, Smart N P, Soria-Vazquez E. More efficient constant-round multi-party computation from BMR and SHE. In: Proceedings of Theory of Cryptography Conference, 2016. 554–581
91 Wang X, Ranellucci S, Katz J. Authenticated garbling and efficient maliciously secure two-party computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 21–37
92 Wang X, Ranellucci S, Katz J. Global-scale secure multiparty computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 39–56
93 Hazay G, Scholl P, Soria-Vazquez E. Low cost constant round MPC combining BMR and oblivious transfer. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2017. 598–628
94 Zhu R Y, Cassel D, Sabry A, et al. NANOPI: extreme-scale actively-secure multi-party computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 862–879
95 Katz J, Ranellucci S, Rosulek M, et al. Optimizing authenticated garbling for faster secure two-party computation. In: Proceedings of Annual International Cryptology Conference, 2018. 365–391
96 Hazay C, Scholl P, Soria-Vazquez E. Low cost constant round MPC combining BMR and oblivious transfer. J Cryptol, 2020, 33: 1732–1786
97 Yang K, Wang X, Zhang J. More efficient MPC from improved triple generation and authenticated garbling. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2020. 1627–1646
98 Ben-Efraim A, Cong K L, Omri E, et al. Large scale, actively secure computation from LPN and free-XOR garbled circuits. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 33–63
99 Lindell Y. Secure multiparty computation. Commun ACM, 2021, 64: 86–96
100 Orsini E. Efficient, actively secure MPC with a dishonest majority: a survey. In: Proceedings of International Workshop on the Arithmetic of Finite Fields, 2021. 42–71
101 Feng D, Yang K. Concretely efficient secure multi-party computation protocols: survey and more. Security Saf, 2022, 1: 2021001
102 Cleve R. Limits on the security of coin flips when half the processors are faulty. In: Proceedings of the 18th Annual ACM Symposium on Theory of Computing, 1986. 364–369
103 Shamir A. How to share a secret. Commun ACM, 1979, 22: 612–613
104 Ito M, Saito A, Nishizeki T. Secret sharing scheme realizing general access structure. Electron Comm Jpn Pt III, 1989, 72: 56–64
105 Cramer R, Damgård I, Ishai Y. Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Proceedings of Theory of Cryptography Conference, 2005. 342–362
106 Franklin M K, Yung M. Communication complexity of secure computation. In: Proceedings of the 24th Annual ACM Symposium on Theory of Computing, 1992. 699–710
107 Bendlin R, Damgård I, Orlandi C, et al. Semi-homomorphic encryption and multiparty computation. In: Proceedings

of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2011. 169–188
108 Damgård I, Pastro V, Smart N P, et al. Multiparty computation from somewhat homomorphic encryption. In: Proceedings of Annual Cryptology Conference, 2012. 643–662
109 Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation. In: Proceedings of Annual Cryptology Conference, 2012. 681–700
110 Damgård I, Keller M, Larraia E, et al. Practical covertly secure MPC for dishonest majority — or: breaking the SPDZ limits. In: Proceedings of European Symposium on Research in Computer Security, 2013. 1–18
111 Keller M, Orsini E, Scholl P. MASCOT: faster malicious arithmetic secure computation with oblivious transfer. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2016. 830–842
112 Gennaro R, Rabin M O, Rabin T. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography. In: Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing, 1998. 101–111
113 Damgård I, Nielsen J B. Scalable and unconditionally secure multiparty computation. In: Proceedings of Annual Cryptology Conference, 2007. 572–590
114 Goyal V, Song Y F. Malicious security comes free in honest-majority MPC. 2020. https://eprint.iacr.org/2020/134.pdf
115 Goyal V, Song Y F, Zhu C Z. Guaranteed output delivery comes free in honest majority MPC. In: Proceedings of Annual Cryptology Conference, 2020. 618–646
116 Goyal V, Li H J, Ostrovsky R, et al. ATLAS: efficient and scalable MPC in the honest majority setting. In: Proceedings of Annual Cryptology Conference, 2021. 244–274
117 Abspoel M, Cramer R, Damgård I, et al. Efficient information-theoretic secure multiparty computation over $\mathbb{Z}/p^k\mathbb{Z}$ via galois rings. In: Proceedings of Theory of Cryptography Conference, 2019. 471–501
118 Abspoel M, Cramer R, Damgård I, et al. Asymptotically good multiplicative LSSS over Galois rings and applications to MPC over $\mathbb{Z}/p^k\mathbb{Z}$. In: Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, 2020. 151–180
119 Escudero D, Soria-Vazquez E. Efficient information-theoretic multi-party computation over non-commutative rings. In: Proceedings of Annual Cryptology Conference, 2021. 335–364
120 Guruswami V, Wootters M. Repairing reed-solomon codes. In: Proceedings of the 48th Annual ACM Symposium on Theory of Computing, 2016. 216–226
121 Abspoel M, Cramer R, Escudero D, et al. Improved single-round secure multiplication using regenerating codes. 2021. https://eprint.iacr.org/2021/253.pdf
122 Araki T, Furukawa J, Lindell Y, et al. High-throughput semi-honest secure three-party computation with an honest majority. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2016. 805–817
123 Lindell Y, Nof A. A framework for constructing fast MPC over arithmetic circuits with malicious adversaries and an honest-majority. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2017. 259–276
124 Dalskov A P K, Escudero D, Keller M. Fantastic four: honest-majority four-party secure computation with malicious security. In: Proceedings of USENIX Security Symposium, 2021. 2183–2200
125 Boneh D, Boyle E, Corrigan-Gibbs H, et al. Zero-knowledge proofs on secret-shared data via fully linear PCPS. In: Proceedings of Annual Cryptology Conference, 2019. 67–97
126 Beerliová-Trubíniová Z, Martin Hirt M. Perfectly-secure MPC with linear communication complexity. In: Proceedings of Theory of Cryptography Conference, 2008. 213–230
127 Damgård I, Geisler M, Krøigaard M, et al. Asynchronous multiparty computation: theory and implementation. In: Proceedings of International Workshop on Public Key Cryptography, 2009. 160–179
128 Genkin D, Ishai Y, Prabhakaran M, et al. Circuits resilient to additive attacks with applications to secure computation. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, 2014. 495–504
129 Genkin D, Ishai Y, Polychroniadou A. Efficient multi-party computation: from passive to active security via secure SIMD circuits. In: Proceedings of Annual Cryptology Conference, 2015. 721–741
130 Furukawa J, Lindell Y, Nof A, et al. High-throughput secure three-party computation for malicious adversaries

and an honest majority. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2017. 225–255
131 Araki T, Barak A, Furukawa J, et al. Optimized honest-majority MPC for malicious adversaries-breaking the 1 billion-gate per second barrier. In: Proceedings of Symposium on Security and Privacy, 2017. 843–862
132 Furukawa J, Lindell Y. Two-thirds honest-majority MPC for malicious adversaries at almost the cost of semi-honest. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 1557–1571
133 Chida K, Genkin D, Hamada K, et al. Fast large-scale honest-majority MPC for malicious adversaries. In: Proceedings of Annual Cryptology Conference, 2018. 34–64
134 Nordholt P S, Veeningen M. Minimising communication in honest-majority MPC by batchwise multiplication verification. In: Proceedings of International Conference on Applied Cryptography and Network Security, 2018. 321–339
135 Boyle E, Gilboa N, Ishai Y, et al. Practical fully secure three-party computation via sublinear distributed zero-knowledge proofs. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 869–886
136 Boyle E, Gilboa N, Ishai Y, et al. Efficient fully secure computation via distributed zero-knowledge proofs. In: Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, 2020. 244–276
137 Cascudo I, Cramer R, Xing C P, et al. Amortized complexity of information-theoretically secure MPC revisited. In: Proceedings of Annual Cryptology Conference, 2018. 395–426
138 Polychroniadou A, Song Y F. Constant-overhead unconditionally secure multiparty computation over binary fields. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 812–841
139 Beck G, Goel A, Jain A, et al. Order-C secure multiparty computation for highly repetitive circuits. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 663–693
140 Damgård I, Ishai Y, Krøigaard M. Perfectly secure multiparty computation and the computational overhead of cryptography. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2010. 445–465
141 Garay J A, Ishai Y, Ostrovsky R, et al. The price of low communication in secure multi-party computation. In: Proceedings of Annual Cryptology Conference, 2017. 420–446
142 Goyal V, Polychroniadou A, Song Y F. Unconditional communication-efficient MPC via Hall's marriage theorem. In: Proceedings of Annual Cryptology Conference, 2021. 275–304
143 Gordon S D, Starin D, Yerukhimovich A. The more the merrier: reducing the cost of large scale MPC. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 694–723
144 Rabin M O. How to Exchange Secrets by Oblivious Transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981
145 Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts. Commun ACM, 1985, 28: 637–647
146 Applebaum B, Damgård I, Ishai Y, et al. Secure arithmetic computation with constant computational overhead. In: Proceedings of Annual Cryptology Conference, 2017. 223–254
147 Boyle E, Couteau G, Gilboa N, et al. Compressing vector OLE. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2018. 896–912
148 Schneider T, Zohner M. GMW vs. Yao? Efficient secure two-party computation with low depth circuits. In: Proceedings of International Conference on Financial Cryptography and Data Security, 2013. 275–292
149 Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer and extensions for faster secure computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2013. 535–548
150 Ben-Efraim A, Nielsen M, Omri E. Turbospeedz: double your online SPDZ! Improving SPDZ using function dependent preprocessing. In: Proceedings of International Conference on Applied Cryptography and Network Security, 2019. 530–549
151 Boyle E, Gilboa N, Ishai Y, et al. Sublinear GMW-style compiler for MPC with preprocessing. In: Proceedings of Annual Cryptology Conference, 2021. 457–485
152 Ishai Y, Prabhakaran M, Sahai A. Founding cryptography on oblivious transfer-efficiently. In: Proceedings of Annual

Cryptology Conference, 2008. 572–591
153 Lindell Y, Oxman E, Pinkas B. The IPS compiler: optimizations, variants and concrete efficiency. In: Proceedings of Annual Cryptology Conference, 2011. 259–276
154 Hazay C, Ishai Y, Marcedone A, et al. LevioSa: lightweight secure arithmetic computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 327–344
155 Hazay C, Venkitasubramaniam M, Weiss M. The price of active security in cryptographic protocols. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 184–215
156 Keller M, Pastro V, Rotaru D. Overdrive: making SPDZ great again. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2018. 158–189
157 Baum C, Cozzo D, Smart N P. Using TopGear in overdrive: a more efficient ZkPok for SPDZ. In: Proceedings of International Conference on Selected Areas in Cryptography, 2019. 274–302
158 Chen H, Kim M, Razenshteyn I P, et al. Maliciously secure matrix multiplication with applications to private deep learning. In: Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, 2020. 31–59
159 Cramer R, Damgård I, Escudero D, et al. SPD $\mathbb{Z}_{2^k}$: efficient MPC mod 2^k for dishonest majority. In: Proceedings of Annual Cryptology Conference, 2018. 769–798
160 Damgård I, Escudero D, Frederiksen T K, et al. New primitives for actively-secure MPC over rings with applications to private machine learning. In: Proceedings of IEEE Symposium on Security and Privacy, 2019. 1102–1120
161 Orsini E, Smart N P, Vercauteren F. Overdrive2k: efficient secure MPC over $\mathbb{Z}_{2^k}$ from somewhat homomorphic encryption. In: Proceedings of the Cryptographers' Track at the RSA Conference, 2020. 254–283
162 Catalano D, Di-Raimondo M, Fiore D, et al. Mon$\mathbb{Z}_{2^k}$a: fast maliciously secure two party computation on $\mathbb{Z}_{2^k}$. In: Proceedings of International Workshop on Public Key Cryptography, 2020. 357–386
163 Cheon J H, Kim D, Lee K. MHz2k: MPC from He over $\mathbb{Z}_{2^k}$ with new packing, simpler reshare, and better ZKP. In: Proceedings of Annual Cryptology Conference, 2021. 426–456
164 Larraia E, Orsini E, Mart N P. Dishonest majority multi-party computation for binary circuits. In: Proceedings of Annual Cryptology Conference, 2014. 495–512
165 Frederiksen T K, Keller M, Orsini E, et al. A unified approach to MPC with preprocessing using OT. In: Proceedings of the 21st International Conference on Advances in Cryptology, 2015. 711–735
166 Damgård I, Zakarias S. Constant-overhead secure computation of Boolean circuits using preprocessing. In: Proceedings of Theory of Cryptography Conference, 2013. 621–641
167 Damgård I, Lauritsen R, Toft T. An empirical study and some improvements of the MiniMac protocol for secure computation. In: Proceedings of International Conference on Security and Cryptography for Networks, 2014. 398–415
168 Damgård I, Zakarias R W. Fast oblivious AES a dedicated application of the MiniMac protocol. In: Proceedings of International Conference on Cryptology in Africa, 2016. 245–264
169 Frederiksen T K, Pinkas B, Yanai A. Committed MPC — maliciously secure multiparty computation from homomorphic commitments. In: Proceedings of International Workshop on Public Key Cryptography, 2018. 587–619
170 Cascudo I, Gundersen J S. A secret-sharing based MPC protocol for Boolean circuits with good amortized complexity. In: Proceedings of Theory of Cryptography Conference, 2020. 652–682
171 Impagliazzo R, Rudich S. Limits on the provable consequences of one-way permutations. In: Proceedings of Annual Cryptology Conference, 1990. 8–26
172 McQuoid I, Rosulek M, Roy L. Minimal symmetric Pake and 1-out-of-N OT from programmable-once public functions. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2020. 425–442
173 McQuoid I, Rosulek M, Roy L. Batching base oblivious transfers. In: Proceedings of International Conference on Advances in Cryptology, 2021. 281–310
174 Naor M, Pinkas B. Efficient oblivious transfer protocols. In: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, 2001. 448–457
175 Chou T, Orlandi C. The simplest protocol for oblivious transfer. In: Proceedings of International Conference on Cryptology and Information Security in Latin America, 2015. 40–58
176 Peikert C, Vaikuntanathan V, Waters B. A framework for efficient and composable oblivious transfer. In: Proceedings of Annual International Cryptology Conference, 2008. 554–571

177 Döttling N, Garg S, Hajiabadi M, et al. Two-round oblivious transfer from CDH or LPN. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 768–797
178 Lai Y F, Galbraith S D, de Saint Guilhem C. Compact, efficient and UC-secure isogeny-based oblivious transfer. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 213–241
179 Beaver D. Correlated pseudorandomness and the complexity of private computations. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996. 479–488
180 Ishai Y, Kilian J, Nissim K, et al. Extending oblivious transfers efficiently. In: Proceedings of Annual Cryptology Conference, 2003. 145–161
181 Yang K, Weng C K, Lan X, et al. Ferret: fast extension for correlated OT with small communication. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2020. 1607–1626
182 Roy L. Softspokenot: communication-computation tradeoffs in OT extension. 2022. https://eprint.iacr.org/2022/192.pdf
183 Kolesnikov V, Kumaresan R. Improved OT extension for transferring short secrets. In: Proceedings of Annual Cryptology Conference, 2013. 54–70
184 Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer extensions with security for malicious adversaries. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015. 673–701
185 Keller M, Orsini E, Scholl P. Actively secure OT extension with optimal overhead. In: Proceedings of Annual Cryptology Conference, 2015. 724–741
186 Scholl P. Extending oblivious transfer with low communication via key-homomorphic PRFS. In: Proceedings of International Workshop on Public Key Cryptography, 2018. 554–583
187 Boyle E, Couteau G, Gilboa N, et al. Efficient pseudorandom correlation generators: silent OT extension and more. In: Proceedings of Annual Cryptology Conference, 2019. 489–518
188 Boyle E, Couteau G, Gilboa N, et al. Efficient two-round OT extension and silent non-interactive secure computation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 291–308
189 Couteau G, Rindal P, Raghuraman S. Silver: silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In: Proceedings of Annual Cryptology Conference, 2021. 502–534
190 Boyle E, Couteau G, Gilboa N, et al. Correlated pseudorandom functions from variable-density LPN. In: Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS), 2020. 1069–1080
191 Boyle E, Couteau G, Gilboa N, et al. Correlated pseudorandomness from expand-accumulate codes. In: Proceedings of Annual Cryptology Conference, 2022. 603–633
192 Gilboa N. Two party RSA key generation. In: Proceedings of Annual Cryptology Conference, 1999. 116–129
193 Chase M, Dodis Y, Ishai Y, et al. Reusable non-interactive secure computation. In: Proceedings of Annual Cryptology Conference, 2019. 462–488
194 Baum C, Escudero D, Pedrouzo-Ulloa A, et al. Efficient protocols for oblivious linear function evaluation from ring-LWE. In: Proceedings of the 12th International Conference on Security and Cryptography for Networks, 2020. 130–149
195 Castro L D, Juvekar C, Vaikuntanathan V. Fast vector oblivious linear evaluation from ring learning with errors. In: Proceedings of the 9th on Workshop on Encrypted Computing & Applied Homomorphic Cryptography, 2021. 29–41
196 Branco P, Döttling N, Mateus P. Two-round oblivious linear evaluation from learning with errors. In: Proceedings of International Workshop on Public Key Cryptography, 2022. 379–408
197 Naor M, Pinkas B. Oblivious transfer and polynomial evaluation. In: Proceedings of the 31st Annual ACM Symposium on Theory of Computing, 1999. 245–254
198 Ishai Y, Prabhakaran M, Sahai A. Secure arithmetic computation with no honest majority. In: Proceedings of Theory of Cryptography Conference, 2009. 294–314
199 Ghosh S, Nielsen J B, Nilges T. Maliciously secure oblivious linear function evaluation with constant overhead. In: Proceedings of International Conference on Advances in Cryptology, 2017. 629–659
200 Boyle E, Couteau G, Gilboa N, et al. Efficient pseudorandom correlation generators from ring-LPN. In: Proceedings

of Annual Cryptology Conference, 2020. 387–416
201 Abram D, Scholl P. Low-communication multiparty triple generation for SPDZ from ring-LPN. In: Proceedings of International Workshop on Public Key Cryptography, 2022. 221–251
202 Schoppmann P, Gascón A, Reichert L, et al. Distributed vector-OLE: improved constructions and implementation. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 1055–1072
203 Weng C K, Yang K, Katz J, et al. Wolverine: fast, scalable, and communication-efficient zero-knowledge proofs for Boolean and arithmetic circuits. In: Proceedings of IEEE Symposium on Security and Privacy, 2021. 1074–1091
204 Rivest R L, Adleman L, Dertouzos M L, et al. On data banks and privacy homomorphisms. Foundation Sec Comput, 1978, 4: 169–180
205 Gentry C. Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 2009. 169–178
206 van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2010. 24–43
207 Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE. SIAM J Comput, 2014, 43: 831–871
208 Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012. 309–325
209 Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. In: Proceedings of Annual Cryptology Conference, 2013. 75–92
210 Cheon J H, Kim A, Kim M, et al. Homomorphic encryption for arithmetic of approximate numbers. In: Proceedings of International Conference on Advances in Cryptology, 2017. 409–437
211 Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSvp. In: Proceedings of Annual Cryptology Conference, 2012. 868–886
212 Smart N P, Vercauteren F. Fully homomorphic SIMD operations. Des Codes Cryptogr, 2014, 71: 57–81
213 Gentry C, Halevi S, Smart N P. Fully homomorphic encryption with polylog overhead. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2012. 465–482
214 Alperin-Sheriff J, Peikert C. Practical bootstrapping in quasilinear time. In: Proceedings of Annual Cryptology Conference, 2013. 1–20
215 Gentry C, Halevi S, Smart N P. Better bootstrapping in fully homomorphic encryption. In: Proceedings of the 15th International Conference on Practice and Theory in Public Key Cryptography, 2012. 1–16
216 Halevi S, Shoup V. Algorithms in HELIB. In: Proceedings of Annual Cryptology Conference, 2014. 554–571
217 Gentry C, Halevi S, Smart N P. Homomorphic evaluation of the AES circuit. In: Proceedings of Annual Cryptology Conference, 2012. 850–867
218 Gentry C, Halevi S, Vaikuntanathan V. i-Hop homomorphic encryption and rerandomizable Yao circuits. In: Proceedings of Annual Cryptology Conference, 2010. 155–172
219 Fan J F, Vercauteren F. Somewhat practical fully homomorphic encryption. 2012. https://eprint.iacr.org/2012/144.pdf
220 Chillotti I, Gama N, Georgieva M, et al. TFHE: fast fully homomorphic encryption over the torus. J Cryptol, 2020, 33: 34–91
221 Chor B, Goldreich O, Kushilevitz E, et al. Private information retrieval. In: Proceedings of the 36th Annual Foundations of Computer Science, 1995. 41–50
222 Goldreich O, Micali S, Wigderson A. How to play any mental game, or a completeness theorem for protocols with honest majority. In: Proceedings of Providing Sound Foundations for Cryptography: on the Work of Shafi Goldwasser and Silvio Micali, 2019. 307–328
223 Boura C, Gama N, Georgieva M, et al. CHIMERA: combining ring-LWE-based fully homomorphic encryption schemes. J Math Cryptology, 2020, 14: 316–338
224 Lu W J, Huang Z C, Hong C, et al. Pegasus: bridging polynomial and non-polynomial evaluations in homomorphic encryption. In: Proceedings of IEEE Symposium on Security and Privacy, 2021. 1057–1073
225 Bellare M, Goldreich O. On defining proofs of knowledge. In: Proceedings of Annual Cryptology Conference, 1993. 390–420

226 Fiat A, Shamir A. How to prove yourself: practical solutions to identification and signature problems. In: Proceedings of Annual Cryptology Conference, 1987. 186–194
227 Gabizon A, Williamson Z J, Ciobotaru O. Plonk: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. 2019. https://eprint.iacr.org/2019/953.pdf
228 Zhang J H, Liu T Y, Wang W J, et al. Doubly efficient interactive proofs for general arithmetic circuits with linear prover time. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2021. 159–177
229 Ben-Sasson E, Bentov I, Horesh Y, et al. Scalable, transparent, and post-quantum secure computational integrity. 2018. https://eprint.iacr.org/2018/046.pdf
230 Bitansky N, Canetti R, Chiesa A, et al. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, 2012
231 Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing. In: Proceedings of Annual Cryptology Conference, 1992. 129–140
232 Chiesa A, Hu Y C, Maller M, et al. Marlin: preprocessing zkSNARKs with universal and updatable SRS. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 738–768
233 Groth J. On the size of pairing-based non-interactive arguments. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2016. 305–326
234 Maller M, Bowe S, Kohlweiss M, et al. Sonic: zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2019. 2111–2128
235 Bünz B, Fisch B, zepieniec A. Transparent snarks from dark compilers. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 677–706
236 Chiesa A, Ojha D, Spooner N. FRACTAL: post-quantum and transparent recursive proofs from holography. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020
237 Ben-Sasson E, Chiesa A, Spooner N. Interactive oracle proofs. 2016. https://eprint.iacr.org/2016/116.pdf
238 Xie T C, Zhang J H, Zhang Y P, et al. Libra: succinct zero-knowledge proofs with optimal prover computation. In: Proceedings of Annual Cryptology Conference, 2019. 733–764
239 Gennaro R, Gentry C, Parno B, et al. Quadratic span programs and succinct NIZKs without PCPs. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013. 626–645
240 Ben-Sasson E, Bentov I, Horesh Y, et al. Scalable zero knowledge with no trusted setup. In: Proceedings of Annual Cryptology Conference, 2019. 701–732
241 StarkWare. EthSTARK documentation. 2021. https://eprint.iacr.org/2021/582.pdf
242 Goldberg L, Papini S, Riabzev M. Cairo — a turing-complete stark-friendly CPU architecture. 2021. https://eprint.iacr.org/2021/1063.pdf
243 Groth J, Kohlweiss M, Maller M, et al. Updatable and universal common reference strings with applications to zk-SNARKs. In: Proceedings of Annual International Cryptology Conference, 2018
244 Bootle J, Chiesa A, Hu Y C, et al. Gemini: elastic SNARKs for diverse environments. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2022
245 Lund C, Fortnow L, Karloff H, et al. Algebraic methods for interactive proof systems. J ACM, 1992, 39: 859–868
246 Goldwasser S, Kalai Y, Rothblum G. Delegating computation: interactive proofs for muggles. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, 2008. 113–122
247 Ben-Sasson E, Chiesa A, Riabzev M, et al. Aurora: transparent succinct arguments for R1CS. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019
248 Ben-Sasson E, Bentov I, Horesh Y, et al. Fast reed-solomon interactive oracle proofs of proximity. In: Proceedings of the 45th International Colloquium on Automata, Languages, and Programming, 2018
249 Yang K, Sarkar P, Weng C K, et al. QuickSilver: efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications

Security, 2021. 2986–3001
250 Weng C K, Yang K, Yang Z M, et al. AntMan: interactive zero-knowledge proofs with sublinear communication. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, 2022
251 Schnorr C-P. Efficient identification and signatures for smart cards. In: Proceedings of Annual Cryptology Conference, 1990. 239–252
252 Esgin M F, Steinfeld R, Liu J K, et al. Lattice-based zero-knowledge proofs: new techniques for shorter and faster constructions and applications. In: Proceedings of Annual Cryptology Conference, 2019. 115–146
253 Barak B, Goldreich O, Impagliazzo R, et al. On the (im)possibility of obfuscating programs. In: Proceedings of Annual Cryptology Conference, 2001. 1–18
254 Goldwasser S, Rothblum G N. On best-possible obfuscation. In: Proceedings of Theory of Cryptography Conference, 2007. 194–213
255 Bellare M, Stepanovs I, Waters B. New negative results on differing-inputs obfuscation. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2016. 792–821
256 Garg S, Gentry C, Halevi S, et al. Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Proceedings of the 54th Annual Symposium on Foundations of Computer Science, 2013. 40–49
257 Sahai A, Waters B. How to use indistinguishability obfuscation: deniable encryption, and more. In: Proceedings of the 46th Annual ACM Symposium on Theory of Computing, 2014. 475–484
258 Goldreich O, Goldwasser S, Micali S. How to construct random functions (extended abstract). In: Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1984. 464–479
259 Canetti R, Lin H J, Tessaro S, et al. Obfuscation of probabilistic circuits and applications. In: Proceedings of Theory of Cryptography Conference, 2015. 468–497
260 Cohen A, Holmgren J, Nishimaki R, et al. Watermarking cryptographic capabilities. In: Proceedings of the 48th Annual ACM Symposium on Theory of Computing, 2016. 1115–1127
261 Bitansky N, Paneth O, Rosen A. On the cryptographic hardness of finding a Nash equilibrium. In: Proceedings of the 56th Annual Symposium on Foundations of Computer Sciences, 2015. 1480–1498
262 Bartusek J, Ishai Y, Jain A, et al. Affine determinant programs: a framework for obfuscation and witness encryption. In: Proceedings of the 11th Innovations in Theoretical Computer Science Conference, 2020
263 Garg S, Gentry C, Halevi S. Candidate multilinear maps from ideal lattices. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2013. 1–17
264 Coron J-S, Lepoint T, Tibouchi M. Practical multilinear maps over the integers. In: Proceedings of Annual Cryptology Conference, 2013. 476–493
265 Gentry C, Gorbunov S, Halevi S. Graph-induced multilinear maps from lattices. In: Proceedings of Theory of Cryptography Conference, 2015. 498–527
266 Barak B, Garg S, Kalai Y T, et al. Protecting obfuscation against algebraic attacks. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2014. 221–238
267 Brakerski Z, Rothblum G N. Virtual black-box obfuscation for all circuits via generic graded encoding. In: Proceedings of Theory of Cryptography Conference, 2014. 1–25
268 Cheon J H, Han K, Lee C M, et al. Cryptanalysis of the multilinear map over the integers. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2015. 3–12
269 Hu Y P, Jia H W. Cryptanalysis of GGH map. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2016. 537–565
270 Miles E, Sahai A, Zhandry M. Annihilation attacks for multilinear maps: cryptanalysis of indistinguishability obfuscation over GGH13. In: Proceedings of Annual Cryptology Conference, 2016. 629–658
271 Chen Y L, Gentry C, Halevi S. Cryptanalyses of candidate branching program obfuscators. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2017. 278–307
272 Ananth P, Jain A. Indistinguishability obfuscation from compact functional encryption. In: Proceedings of Annual Cryptology Conference, 2015. 308–326
273 Bitansky N, Vaikuntanathan V. Indistinguishability obfuscation from functional encryption. In: Proceedings of the 56th Annual Symposium on Foundations of Computer Science, 2015. 171–190
274 Lin H J, Pass R, Seth K, et al. Output-compressing randomized encodings and applications. In: Proceedings of

Theory of Cryptography Conference, 2016. 96–124
275 Lin H J. Indistinguishability obfuscation from constant-degree graded encoding schemes. In: Proceedings of Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2016. 28–57
276 Lin H J, Vaikuntanathan V. Indistinguishability obfuscation from DDH-like assumptions on constant-degree graded encodings. In: Proceedings of the 57th Annual Symposium on Foundations of Computer Science, 2016. 11–20
277 Lin H J. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGS. In: Proceedings of Annual Cryptology Conference, 2017. 599–629
278 Lin H J, Tessaro S. Indistinguishability obfuscation from trilinear maps and block-wise local PRGS. In: Proceedings of Annual Cryptology Conference, 2017. 630–660
279 Applebaum B, Ishai Y, Kushilevitz E. Cryptography in Nc^0. In: Proceedings of the 45th Annual Symposium on Foundations of Computer Science, 2004. 166–175
280 Jain A, Lin H J, Sahai A. Indistinguishability obfuscation from well-founded assumptions. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021. 60–73
281 Jain A, Lin H J, Matt C, et al. How to leverage hardness of constant-degree expanding polynomials overa $\mathbb{R}$ to build $i\mathcal{O}$. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019. 251–281
282 Gay R, Jain A, Lin H J, et al. Indistinguishability obfuscation from simple-to-state hard problems: new assumptions, new techniques, and simplification. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 97–126
283 Gorbunov S, Vaikuntanathan V, Wee H. Predicate encryption for circuits from LWE. In: Proceedings of the Annual Cryptology Conference, 2015. 503–523
284 Boneh D, Gentry C, Gorbunov S, et al. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2014. 533–556
285 Jain A, Korb A, Manohar N, et al. Amplifying the security of functional encryption, unconditionally. In: Proceedings of the Annual Cryptology Conference, 2020. 717–746
286 Jain A, Lin H J, Sahai A. Indistinguishability obfuscation from LPN over $\mathbb{F}_p$, DLIN, and PRGs in NC^0. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2022. 670–699
287 Brakerski Z, Döttling N, Garg S, et al. Candidate IO from homomorphic encryption schemes. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 79–109
288 Gay R, Pass R. Indistinguishability obfuscation from circular security. In: Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, 2021. 736–749
289 Hopkins S B, Jain A, Lin H J. Counterexamples to new circular security assumptions underlying IO. In: Proceedings of the Annual Cryptology Conference, 2021. 673–700
290 Wee H, Wichs D. Candidate obfuscation via oblivious LWE sampling. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2021. 127–156
291 Devadas L, Quach W, Vaikuntanathan V, et al. Succinct LWE sampling, random polynomials, and obfuscation. In: Proceedings of the 19th International Conference, Theory of Cryptography Conference, 2021. 256–287
292 Agrawal S. Indistinguishability obfuscation without multilinear maps: new methods for bootstrapping and instantiation. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2019. 191–225
293 Agrawal S, Pellet-Mary A. Indistinguishability obfuscation without maps: attacks and fixes for noisy linear Fe. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2020. 110–140
294 Yao L, Chen Y L, Yu Y. Cryptanalysis of candidate obfuscators for affine determinant programs. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, 2022. 645–669

Privacy-preserving cryptographic algorithms and protocols: a survey on designs and applications

Wei HUO[1], Yu YU[2*], Kang YANG[3], Zhongxiang ZHENG[4], Xiangxue LI[2*], Li YAO[2] & Jie XIE[2]

1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China*;
2. *School of Computer Science, Fudan University, Shanghai* 200438*, China*;
3. *State Key Laboratory of Cryptology, Beijing* 100878*, China*;
4. *School of Computer and Cyber Sciences, Communication University of China, Beijing* 100024*, China*
* Corresponding author. E-mail: yyuu@sjtu.edu.cn, xxli@cs.ecnu.edu.cn

Abstract The emergence of technologies such as cloud computing, big data analytics, the Internet of Things (IoT), mobile internet, artificial intelligence (AI), and blockchain has given rise to several critical security concerns, especially regarding privacy breaches. Sophisticated cryptographic algorithms and protocols, including secure multiparty computation, homomorphic encryption, zero-knowledge proof, and indistinguishability obfuscation, provide effective solutions to address these challenges. The cryptographic community has already witnessed numerous commendable efforts aimed at mitigating privacy breaches through the utilization of advanced cryptographic algorithms and protocols. This study offers a comprehensive survey of the designs and applications of such algorithms and protocols, assessing them based on their security level, module rationale, and limitations regarding security and performance. This survey acts as a valuable resource for readers seeking a thorough understanding of the latest advancements in cryptographic algorithms and protocols. It also serves to bridge the gap between the theoretical underpinnings and practical applications of these advanced cryptographic primitives.

Keywords privacy preserving, secure multiparty computation, homomorphic encryption, zero-knowledge proof, indistinguishability obfuscation