GM/Y 5009-2024

区块链隐私保护机制中的范围证明算法和环签名算法研究

密码行业标准化技术委员会
CRYPTOGRAPHY STANDARDIZATION TECHNICAL COMMITTEE

2024 年 12 月

摘要

范围证明算法和环签名算法是隐私保护区块链系统中的常见密码算法，能够为区块链系统提供金额和账户的隐私保护功能，在金融、支付等场景中应用广泛。但是，强隐私性造成了难以监管的负面效果，使得隐私保护区块链难以适应当前的金融监管政策，因此需要算法兼具隐私保护与监管审计功能。

本研究报告对当前区块链隐私保护机制中已有的范围证明算法和环签名算法进行了调研和总结，包含各个算法的设计原理、困难问题、安全模型、实现效率等方面的优缺点分析，指出了当前算法效率不高以及缺乏监管审计功能的不足之处。为了解决效率和监管问题，进一步推动区块链隐私保护密码算法的应用推广进程，我们分别给出了范围证明算法、可审计范围证明算法、环签名算法、可链接环签名算法、可审计环签名算法、多环签名算法的应用推广要求，包含功能性要求、安全性要求、性能要求和参考指标。

根据本报告的应用推广要求，我们给出了新的算法设计思路和方法，算法对应的安全性模型和未来研究中需要攻克的难点问题，以及初步的性能评估结果，面向专家组和产业链各单位征求意见。

关键词：范围证明，可审计范围证明，环签名，可链接环签名，可审计环签名，多环签名

目录

前言

本研究报告是由密码行业标准化技术委员会根据国家密码管理局《2020 年密码行业标准制/修订计划》下达的 2020 年密码行业标准研究编制工作任务。项目名称为《区块链隐私保护机制中的范围证明算法和环签名算法研究》，项目类型为研究类项目，项目所属工作组为基础工作组。

本报告梳理区块链隐私保护机制中的范围证明算法和环签名算法规范体系，明确算法研究需求，能够保护用户的隐私数值信息，也能够适配监管需求，为国产化范围证明算法和环签名算法在隐私保护区块链中的应用提供指导。

本报告起草单位：兴唐通信科技有限公司、建信金融科技有限责任公司、中国人民银行数字货币研究所、北京航空航天大学、中国科学院信息工程研究所、武汉大学、中山大学、中国电力科学研究院有限公司。

本报告主要起草人：万兆泽、马永彪、李鸿利、高胜、王妮娜、刘海英、李武璐、王雪、刘明君、伍前红、徐海霞、何德彪、田海博、李智虎。

区块链隐私保护机制中的范围证明算法和环签名算法研究

1 背景和目标

1.1 研究背景

区块链分为公有链、私有链、联盟链三类，因其具有透明性、不可篡改性、去中心化等特性而受到广泛关注。区块链技术自从在 2008 年被首次提出之后，经过十余年的技术发展，已经逐渐应用于金融、贸易、存证、溯源等各个领域。然而，以比特币[1]、以太坊[2]、超级账本等项目为代表的传统区块链系统隐私保护功能较弱，基于明文账本架构，在实际应用中会泄露相关的交易金额和账户地址，造成用户隐私信息泄露，在技术层面制约了其进一步应用落地的空间，因此有必要深入研究区块链的隐私保护机制与相关密码算法，并积极推进相关算法的应用。

为了解决传统区块链系统的隐私保护问题，范围证明算法和环签名算法逐渐得到关注。范围证明是一类特殊的非交互零知识证明算法，用于证明某个隐私数值在确定的范围（区间）之内，而不泄露该隐私数值的任何其他信息。在生成证明之前通常使用杂凑[3]、承诺、加密等方式对数值进行随机化隐私处理，验证者可以获取范围证明并检验其合法性，即确认该隐私数值在确定的范围之内，但无法获取包括隐私数值在内的任何其他信息，实现了隐私保护功能。在隐私保护区块链实际应用中，范围证明常被用于隐藏交易金额等隐私数值，在门罗币[4]（Monero）、德罗币[5]（Dero）、Zether[6]等数字货币项目，以及具备隐私保护功能的供应链金融、货币兑换协议中应用广泛。

环签名是传统数字签名的功能扩展，在环签名算法中，签名人指定了一个可能的签名者公钥集合（或称为公钥环），并对消息进行环签名，验证者通过验证环签名的合法性，能够确信该环签名的签名人来自该公钥集合，但是无法指出真实签名人，实现了签名人的匿名性。可链接环签名是在普通环签名算法的基础上，在环签名中添加签名标签，实现了识别双重签名（使用同一私钥签名两次）的功能。在隐私保护区块链系统中，可链接环签名被用于隐藏签名人身份（账户或资金流），同时抵抗双重支付攻击，在门罗币等隐私保护数字货币项目中得到应用。

需要注意的是，当前隐私保护区块链系统的强隐私性会导致链上交易信息无法被监管机构监管审计，容易引起滥用隐私进行违法犯罪（洗钱、资产转移、非法交易、诈骗、赌博等）的潜在风险，难以有效打击犯罪，不利于隐私保护区块链应用和产业的健康与可持续发展。因此，在区块链隐私保护相关密码的应用推广进程中，在实现隐私保护安全性的同时，也要考虑监管审计与隐私保护相融合的技术路线，确保整个区块链系统的可监管与可审计性。此外，根据区块链的分布式特征，还需有效约束监管者的权利，在算法机制实现可审计性的同时，需要确保当监管者作恶（或被攻击）时的健壮性与可用性，具备抵抗恶意监管者的安全性。

1.2 研究目标

本项目聚焦于区块链隐私保护机制中的范围证明算法和环签名算法研究，评估当前主要算法的技术路线和优缺点，给出算法应用推广参考评价准则和建议候选算法，进一

步推进区块链隐私保护机制中的范围证明算法和环签名算法的应用。本项目的主要研究目标为：

a) 研究现有范围证明算法和环签名算法的优缺点，包括技术路线、安全模型、性能等方面，其中环签名算法包含普通环签名算法、可链接环签名算法和多环签名算法等；
b) 探索范围证明算法和环签名算法在实现隐私保护和监管审计相融合的功能特性，实现监管者无条件穿透式监管审计功能，最大程度符合金额监管相关政策法规，从而促进隐私保护区块链技术应用落地；
c) 给出范围证明算法和可审计范围证明算法应用推广要求，包含功能性要求、安全性要求、性能要求，并给出建议候选算法；
d) 给出环签名算法、可链接环签名算法、可审计环签名算法、多环签名算法应用推广要求，包含功能性要求、安全性要求、性能要求，并给出建议候选算法。

通过本项目的研究内容和研究成果，能够推动隐私保护区块链相关密码算法的应用推广进程，为相关技术和应用早日落地提供更坚实的密码算法保障。

2 场景和必要性

在本章中我们首先介绍隐私保护区块链系统以及范围证明、环签名等算法的主要应用场景，进而论证本项目研究内容的必要性与意义。

2.1 主要应用场景

2.1.1 隐私保护数字资产交易

对于区块链承载数字资产交易与流转功能的场景，其中数字资产包括数字货币、金融票据、有价证券、实物代币等种类。基于明文账本的传统区块链系统，例如比特币、以太坊、EOS、超级账本等，为了实现交易合法性的公开可验证特性，链上的交易信息以明文形式呈现，包含交易金额、交易双方账户地址、账户余额、UTXO（unspent transaction output）与所有者账户的绑定关系等信息，虽然能够实现快速的正确性与合法性校验，支持高 TPS（transaction per second）的交易吞吐量，但是会造成交易信息的隐私泄露，造成企业和用户“不敢上链、不愿上链”的现状。

为了解决这个问题，以范围证明算法和环签名算法为代表的隐私保护密码算法能够实现链上金额和账户地址的隐私保护，并且能够实现公开可验证性。具体而言，在隐私保护区块链数字资产交易中，范围证明算法能够在不公开金额隐私的前提下，确保金额数值处于合法区间，并且能够支持加法或乘法的同态操作；对于环签名算法，通过交易发起者执行环签名操作发起交易，验证者能够检验环签名的合法性，确认签名者公钥来自于公钥集，但是无法获取签名人的真实公钥，实现了签名人的隐私保护功能；在隐私交易模式下，为了防止对于链上资产进行双重支付的攻击，可以使用可链接环签名算法，通过在签名中添加签名标签，实现双重签名的可链接性，进一步提高系统的安全性和可靠性。

在实现隐私保护功能的同时，在数字资产交易和流转环节中，需要为相应的监管机构提供监管审计接入能力，实现在隐私保护的同时兼容穿透式监管功能，才能更加符合金融相关政策法规的要求。需要研究具备监管审计功能的范围证明算法和环签名算法，在原有隐私保护功能的基础上实现监管功能，避免监管缺位导致的洗钱、非法交易、诈

骗等难以追溯的违法犯罪行为，更符合实际的链上资产交易等场景的应用落地监管要求。

2.1.2 隐私保护投票和竞拍

基于区块链系统不可伪造、可追溯等技术特性，能够在投票、竞拍等场景中实现选票（或报价）防篡改和可追溯功能。然而选票（或报价）信息明文上链会造成用户隐私和商业信息泄露，不符合实际的应用需求，因此需要使用隐私保护密码算法来构建基于区块链的隐私保护投票和竞拍系统，实现投票人（报价人）的匿名性和选票内容（报价金额）的隐私性，仍然保持防篡改、可追溯和公开验证功能。在实际应用中，使用范围证明算法能够隐藏投票信息（报价金额），同时能确保隐私选票（隐私报价）的有效性；使用可链接环签名能够实现投票人（报价人）的身份隐私，并且能够通过可链接性防止双重投票（报价）行为的发生。

在实现隐私保护功能的同时，计票人需要具备足够的权限打开隐私选票内容并进行票数统计，拍卖机构需要有能力解密报价金额和报价人身份，对比金额得出获胜用户和成交金额。在区块链公开验证的架构之下，需要研究具备监管审计功能的范围证明算法和环签名算法，计票人（拍卖机构）承担相应监管者的角色，实现对于选票和报价信息的追溯审计，从而实现满足各方需求的区块链投票和拍卖系统。

2.2 研究必要性

我国高度重视区块链和分布式账本的技术创新与应用发展。2019 年 10 月 24 日，习近平在政治局集体学习中指出：“区块链技术应用已延伸到数字金融、物联网、智能制造、供应链管理、数字资产交易等多个领域。目前，全球主要国家都在加快布局区块链技术发展。我国在区块链领域拥有良好基础，要加快推动区块链技术和产业创新发展，积极推进区块链和经济社会融合发展。” 此外，《北京市区块链创新发展行动计划（2020-2022 年）》中提出：“围绕传统金融服务信息校验复杂、成本高、流程长等痛点，推动在供应链金融、资产证券化、跨境支付、贸易融资、智能监管等领域落地一批应用场景，支持相关项目申报金融科技创新监管试点（监管沙箱），促进政府、市场、机构之间多方互信和高效协同，提升金融服务效能。”2020 年 2 月 5 日，中国人民银行发布《金融分布式账本技术安全规范》（JR/T 0184—2020），强调了零知识证明和环签名技术在区块链中的应用价值，给出了区块链隐私保护机制与监管审计的安全规范要求。范围证明和环签名作为应用最广泛的隐私保护算法，将在未来区块链安全技术应用中起到重要作用。

因此，在隐私保护区块链技术领域，针对当前国内外缺乏范围证明算法和环签名算法的现状，在国家相关政策落地和产业支持的大环境下，亟需加强范围证明算法和环签名算法的密码技术攻关，形成原创技术成果，为日益强烈的隐私保护区块链应用需求提供相关算法，构建安全可靠的解决方案，推动区块链和隐私保护技术广泛落地。

3 国内外研究动态

3.1 区块链领域研究现状

区块链技术最早出现在以比特币为代表的虚拟货币中，使用哈希链式结构和默克树存放用户之间的交易信息，通过 UTXO 模型记录交易金额以及防止双重支付，使用数字

签名签发交易，使用共识机制实现全网一致性。比特币之后陆续出现了以太坊、EOS、超级账本等区块链系统，通过基于账户模型的交易体系实现了可以自动化执行的智能合约。但是上述系统都不具备隐私保护功能，交易双方的账户地址和交易金额都以明文呈现，为了实现隐私保护的区块链交易系统，以门罗币和零币为代表的隐私保护区块链系统先后出现。其中，门罗币使用 Pedersen 承诺技术[7]和范围证明技术[8]实现了交易金额的隐私性，使用可链接环签名技术实现了交易输入的匿名性，使用一次性地址技术实现了交易接收方的隐私保护功能。零币系统使用公钥加密以及哈希承诺实现了链上交易信息的完全隐藏，并利用简洁非交互零知识证明系统（zk-SNARKs）实现了整个交易账单的可验证性，但是 zk-SNARKs 方案存在证明生成时间较长，并且需要可信初始化和公共随机串（CRS）等技术，造成零币系统难以在移动设备上运行。对于以门罗币和零币为代表的隐私保护系统，主要存在两方面问题：1）门罗币和零币基于 UTXO 模型，无法支持智能合约；2）门罗币和零币过于强调隐私性，无法有效支持监管审计，容易造成洗钱、走私、非法交易等威胁，在多个国家被政策禁止。除了门罗币和零币之外，学术界和企业界也提出了其他类型的隐私保护区块链系统，其中包括基于混币技术的隐私保护货币（Coinjoin）、基于新型承诺技术的隐私保护货币（Aztec）、基于账户模型和零知识证明的隐私保护货币（Zether）、基于可验证加密技术的账户监管隐私货币（Traceable Monero[9]）等，在隐私性、智能合约适配性、监管性三个层面有所提升，但是都无法同时满足隐私性、完全可监管审计性和智能合约适配性，相关技术研究仍在初期，技术方案尚不成熟，无法有效支撑隐私保护与监管审计的相关应用需求。

当前国际上的区块链数字货币（法币）发展呈现加速态势，相关国家和地区的监管机构逐渐为其应用落地提供政策支持。2019 年，以 Facebook 为首的国际巨头公司首次提出跨境数字货币“Libra”[10]，旨在降低跨境支付成本，提高支付效率；2020 年 Facebook 又发布了“Libra 2.0”白皮书，锚定以美元为首的法币，更加紧贴监管政策，被认为是在西方世界区块链层面推进“美元霸权”的一个有力武器；除此之外，瑞典、法国、韩国等国家以及 Visa、Paypal、摩根大通等金融巨头也在积极布局未来的区块链数字货币研究与设计，在密码技术、共识机制、交易模式、监管策略等方面进行技术积累。目前，包括马来西亚、新加坡、柬埔寨、泰国等国家金融监管机构已经承认了数字货币作为有价资产的合法性。由于相关金融监管政策所限，我国区块链隐私保护交易和监管技术的研究仍处于学术层面，原创的理论和技术成果有限，需要进一步进行密码理论与区块链技术攻关，在国家密码管理局和人民银行等国家管理单位的监管和指导下，推进相关技术和算法的应用，推进产学研结合，构建算法自主设计，同时能够满足隐私保护与可监管性，实现高效安全的区块链系统。

3.2 范围证明算法研究现状

范围证明（Range Proof）是一类特殊的非交互零知识证明协议，用于向验证者证明隐私数据a（a以隐私承诺或密文的形式存在）在某个具体区间$[0,2^n-1]$之内，但是不会泄露隐私数据a的值。基于椭圆曲线的 Pedersen 承诺（$c=g^x h^a$）技术应用于包括门罗币系统在内的各类隐私数据货币系统中，该承诺技术具备无条件隐藏性（unconditional hiding）和计算绑定性（computational binding）。Neother 等在 2016 年使用 Pedersen 承诺技术提出了 Borromean 范围证明，算法构造基于 Borromean 多环签名算法，证明尺寸与隐私金额的范围长度呈线性正比关系。2018 年，Bünz 等人[11]同样使用 Pedersen 承诺，提出了 Bulletproofs 零知识证明算法，实现了不需要可信初始化的非交互零知识证明系统，并且能够直接作为范围证明算法，证明尺寸关于证据长

度呈对数关系，并且在门罗币、德罗币等系统中应用落地，是当前应用最广泛的范围证明算法之一。当前，在其他的隐私保护区块链中，也存在其他类型的承诺机制和范围证明算法，例如 Qiusqius、Aztec、Zether，使用了基于类 El-Gammal 可验证加密算法实现范围证明。此外，还可以通过对于任意运算电路有效的简洁非交互零知识证明（论据）系统或可扩展透明零知识证明系统构造范围证明算法，例如：zk-SNARKs、Plonk、Supersonic、zk-STARKs、Hyrax[12]、Libra[13]等。但是目前的简洁非交互零知识证明（论据）系统存在需要可信初始化等额外假设，并且需要较长的证明生成时间的不足之处；可扩展透明零知识证明系统存在证明尺寸较长的缺点，不适合范围证明算法的高频使用场景。此外，上述算法均不具备监管审计功能，除非数据拥有方向监管者上传数据或者上交密钥，否则监管者无法打开隐私承诺（或密文）中的数据值。上述算法在区块链系统中需要额外的链下通信才能完成对隐私数据的监管审计功能，因此需要研究能够不依赖额外的链下通信的可审计范围证明算法，并给出算法应用推广要求。

3.3 环签名算法研究现状

3.3.1 环签名算法

环签名（Ring Signature）是一类特殊的数字签名，签名人选择公钥群组，并代表群组进行签名，但是签名人身份保持隐私。在传统环签名算法中，签名人自行选取公钥集合（包含自己公钥在内）$L_{\mathrm{PK}}=\{\mathrm{PK}_1,\cdots,\mathrm{PK}_n\}$作为全部的环元素，并且使用自身私钥$\mathrm{SK}_\kappa$进行签名，其中$\kappa\in\{1,\cdots n\}$，在此过程中验证者无法判断公钥集中的签名人身份（求出κ的值），实现了签名人身份隐私保护。环签名算法首先由 Rivest，Shamir 和 Tauman[14]在 2001 年提出，在随机预言机模型下，他们分别给出了基于 RSA 陷门置换和 Robin 陷门函数的环签名算法构造。Abe 等[15]在 2002 年提出了 AOS 环签名算法，该算法同时支持椭圆曲线（通过 Sigma 协议实现）以及 RSA 陷门函数（通过 Hash and sign 方法实现），方案的安全性同样建立在随机预言机模型下。Bender 等[16]在 2006 年给出了第一个在标准模型下安全的环签名算法构造，使用了椭圆曲线配对技术。此外，也有其他实现非线性签名长度的环签名算法构造，包括基于 RSA 累加器技术的环签名算法[17]，具备常数规模$O(1)$的渐进签名尺寸；渐进尺寸为$O(\sqrt{\mathrm{n}})$的标准模型环签名算法[18]，基于椭圆曲线配对技术以及公共随机串（CRS）；以及长度为$O(\log n)$的随机预言机模型环签名算法构造[19]，基于椭圆曲线上的Σ协议，但是签名和验证时间（椭圆曲线点乘数量）为超线性$O(n\log n)$。2019 年，Yuen 等[20]提出了新的环签名算法 Ring-CT 3.0，基于改进的 Bulletproofs 框架，具备对数渐进尺寸的签名长度和线性的签名生成与验证时间，在环元素数量较大时具备尺寸优势。

3.3.2 可链接环签名算法

可链接环签名（Linkable Ring Signature）是一类特殊的环签名算法，在签名过程中生成相应的签名标签（tag），在签名人的身份信息保持隐私的同时，在定义好的某个事件（event）中，使用同一个私钥两次或两次以上签名的行为会被验证者通过链接算法（link）有效识别（仅能识别发生了多次签名，在此过程中签名人身份仍然保持隐私），在不同的事件中多次签名不会被链接算法（link）识别为双重签名。可链接环签名的潜在应用场景包含基于隐私保护数字资产支付、电子投票、电子拍卖等，算法的可链接性可用于识别双重支付、双重投票等异常行为是否发生。在实际算法设计中，需要根据功能与场景的实际需求设计相应的签名标签生成算法（部分可链接环签名的签名标

签生成算法，由于临时公私钥的生成机制，会造成隐私泄露，无法应用于部分类型的隐私货币系统），因此需要在应用推广中考虑适合不同场景的签名标签生成算法。

2004 年，Liu 等[21]首次提出可链接环签名算法 LSAG，在随机预言机模型下，基于离散对数困难假设，使用的签名标签结构为$tag = h^{x_K}$。此后 Tsang 等[22]以及 Au 等[23]分别给出了基于累加器技术的常数尺寸可链接环签名构造方法，基于 RSA 和离散对数等困难问题。2013 年，Yuen 等[24]给出了标准模型下基于椭圆曲线配对技术的可链接环签名构造方法，具备$O(\sqrt{n})$的渐进签名尺寸。2013 年，Liu 等[25]给出了具备无条件隐私性的可链接环签名构造方法，并且给出了可链接环签名算法的形式化安全性模型，基于椭圆曲线困难问题，签名标签为固定生成元结构（$tag = H(\text{event})^{x_K}$）。Saberhagen 等在 2013 年提出了 CryptoNote 方案，其中包含了基于非一致生成元签名标签（$tag = H(g^{x_K})^{x_K}$）的可链接环签名算法，并在第一代门罗币中得到应用。2015 年，Back 等[26]给出了基于可链接环签名算法 LSAG 的改进版本，支持非一致生成元签名标签（$tag = H(g^{x_K})^{x_K}$），相比于 CryptoNote 方案中的签名尺寸有了大幅缩减。

3.3.3 多环签名算法

多环签名算法（Multi-ring Signature）是在单环签名算法的基础上，一次性对于多个公钥集合进行环签名，每个公钥集合中包含一个签名公钥。根据各个签名公钥在各个公钥集的位置，可以分为位置一致的多环签名和位置随机的多环签名两类，其中位置一致的多环签名是指签名公钥在各个公钥集中的位置相同，位置随机的多环签名是指签名公钥在各个公钥集中位置不固定。2015 年，Maxwell 等[27]给出了 Borromean 多环签名算法，各个环中签名公钥位置随机，是基于 AOS 环签名在多环模式下的推广，将签名长度从$nM + M$缩减到$nM + 1$，其中n代表环元素数量，M代表环的数量。2016 年，Noether 等[8]给出了高效的可链接多环签名算法 MLSAG，各个环中签名公钥位置一致，在此基础上实现了门罗币系统的多输入交易支持性，该框架也称为 Ring-CT 1.0。2017 年，Sun 等[28]提出了 Ring-CT 2.0，实现了基于累加器的和椭圆曲线配对技术的可链接多环签名构造方法，各个环中签名公钥位置随机，签名尺寸在渐进意义下短于 Ring-CT1.0，但是当环元素数量M较小时效率不高。2019 年，Yuen 等[20]提出的区块链隐私交易框架 Ring-CT3.0，基于改进的 Bulletproofs 框架，也可以衍生出签名公钥位置随机的多环签名算法，具备对数尺寸的签名长度，但是在常用参数下效率低于 MLSAG（当前门罗币中环元素数量$n = 11$）。2019 年，Goodell 等[29]提出了改进的可链接多环签名算法 CLSAG，实现了比 MLSAG 更紧致的签名尺寸和更高的运行效率，但是在多环中只能实现单个环的可链接性，无法实现全部环的可链接性。

3.4 具备追踪审计功能相关算法研究现状

3.4.1 范围证明算法

对于范围证明算法而言，当前尚不存在具备追踪（监管审计）功能的范围证明算法，可选的技术路线为使用可验证加密技术结合通用零知识证明框架，对于隐私数据$a \in [0,2^n - 1]$，在给出a的范围证明同时，使用监管者公钥PK加密a并给出密文$Enc_{\text{PK}}(a)$的合法性证明，可以保护隐私数据隐私性的同时，实现监管者的追踪审计功能，但是此种方法需要使用额外的零知识证明算法，造成尺寸扩张的同时也带来生成与验证时间的损失。

3.4.2 可追责环签名算法

可追责环签名算法（Accountable Ring Signature）是在环签名的基础上，实现了监管者能够打开签名人身份的可追责性，但是需要具备不可链接性保障多次签名无法被识别。可追责环签名算法首先由 Xu 和 Yung 在 2004 年提出，基于 RSA 陷门置换和随机函数，支持无条件追踪性以及不可链接性。2015 年，Bootle 等提出了基于 DDH 的可追责环签名算法，具备对数签名尺寸，基于随机预言机模型。2018 年，Kumawat 等提出了常数尺寸的可追责环签名算法，基于标准模型。在可追责环签名算法中，要求具备不可链接性（Unlinkability），即任何事件内的多重签名行为无法被验证者识别，在当前的应用场景中无法抵抗双重支付、双重投票、双重报价等潜在攻击，不适合隐私保护区块链系统，尚无区块链应用落地案例。

3.4.3 可追踪环签名算法

可追踪环签名算法（Traceable Ring Signature）是在环签名的基础上，对于同用户的多次签名行为（对不同消息签名），能够识别并追踪签名用户身份；对于同消息的多次签名仅能被识别为链接（linked），签名人身份能够得到保护；对于单次环签名行为，无法追踪签名人身份。2007 年，Fujisaki 等[30]提出了第一个可追踪环签名算法，基于椭圆曲线相关困难问题。可追踪环签名算法提供的是有条件的可追踪性（conditional traceability），但是在区块链隐私交易等场景中，监管者需要对于任意交易（合法和非法）拥有穿透式监管能力，仅对双花交易进行追踪审计在功能层面远远不够，并且可追踪环签名算法也未在区块链系统中落地应用。

3.4.4 多重监管相关算法

在国际贸易、支付结算等跨境场景，存在多个监管者共同监管链上隐私交易等信息，此时需要相关算法支持多方监管的扩展性。当前的相关可监管密码算法和可监管隐私支付方案均不支持在隐私保护基础上的多监管者模式。一个直接的解决方法是针对多个监管者，多次使用单监管者密码算法，但这会造成签名和证明尺寸，以及生成与验证时间随着监管者数量线性增长，影响系统的整体性能，因此需要研究解决一次签名（或证明）同时适配多重监管的算法方案。

4 研究定位和可行性论证

4.1 研究定位

本项目的研究定位是在隐私保护区块链场景下进行范围证明算法和环签名算法调研与应用推广研究，研究新的功能定义与安全性模型，研究新的算法设计方法，参考已有算法的设计思路与功能特性，结合实际应用场景和参数，给出算法的应用推广要求。

4.1.1 范围证明与可审计范围证明算法

在区块链隐私保护机制中的范围证明算法方面，本项目研究当前主流范围证明算法的设计方法、安全性与优缺点，结合实际场景和参数，提出相关算法应用推广要求。根据当前国际区块链领域的应用现状，我们重点研究 Borromean 范围证明算法和 Bulletproofs 范围证明算法，研究提升算法效率的新方法。

对于具备监管审计功能的范围证明算法，本项目给出可审计范围证明（Auditable

Range Proof）的定义与安全性模型，在此基础上给出算法应用推广要求，研究实现算法功能与安全性的关键技术，给出建议的算法设计方法路线，并且研究兼容多重监管的技术路径。

4.1.2 环签名算法与可链接环签名算法

在区块链隐私保护机制中的环签名算法方面，本项目研究当前主流的环签名与可链接环签名算法的设计方法、安全性与优缺点，结合实际场景和参数，提出相关算法应用推广要求。根据当前国际区块链领域的应用现状，在普通环签名方面，我们重点研究 AOS 环签名算法、GK 环签名算法、RingCT3.0 环签名算法。在可链接环签名方面，我们重点研究 CryptoNote 可链接环签名算法和 MLSAG 可链接环签名算法。研究适合区块链场景的公钥结构与签名标签结构，研究新的可链接环签名算法设计方法，研究提升算法效率的新方法。

4.1.3 可审计环签名算法

对于具备监管审计功能的环签名算法，参考已有可追踪环签名与可追责环签名在区块链应用中的功能缺陷，本项目给出可审计环签名（Auditable Ring Signature）的定义与安全性模型，在此基础上给出算法应用推广要求，并且研究实现算法功能与安全性的关键技术，给出建议的算法设计方法路线，并且研究兼容多重监管的技术路径。

4.1.4 多环签名算法

在多环签名算法方面，本项目研究当前主流的可链接多环签名算法的设计方法、安全性与优缺点，结合实际场景和参数，提出可链接多环签名算法应用推广要求。根据当前国际区块链领域的应用现状，我们重点研究 MLSAG 可链接多环签名算法、CLSAG 可链接多环签名算法。研究符合隐私保护区块链相关场景的可链接多环签名算法架构，设计新的多环签名算法，研究提升效率的新方法。此外，针对监管审计的功能点，研究支持无条件追踪性的可审计多环签名设计方法，并且研究兼容多重监管的技术路径。

4.2 可行性论证

4.2.1 范围证明算法应用推广研究

范围证明算法解决了链上隐私数据在保护隐私的同时实现合法性公开验证的问题，在隐私保护区块链和数字货币等领域正在扮演着越来越重要的角色，使用范围证明算法作为金额隐私保护核心组件的隐私货币总市值已超过千亿人民币规模，并在不断增长中。在票据支付、供应链金融、门限管理、质押贷款等多种应用场景前景广泛。

当前，范围证明算法经过十年以上的发展进程，技术上已经逐步走向成熟，已经实现了高效（生成与验证低于 1 毫秒）、紧致（对数证明长度）、安全（不依赖可信初始化等额外假设）的特性，有多个备选的技术路线和经典算法，能够支撑实际的应用场景。因此，进行范围证明密码算法应用推广工作具有较高的可行性与必要性。

此外，面对监管审计政策法规，不能仅仅强调范围证明算法的隐私性，还需要将无条件可追踪性考虑在内，实现隐私保护的同时兼容监管审计。项目组在可审计范围证明算法领域已经取得了一定的理论成果，进行相关算法应用推广研究具有较高的可行性。

4.2.2 环签名算法应用推广研究

环签名算法解决了签名账户身份隐私和签名合法性公开验证的问题，可链接性解决

了双重签名的识别问题，可链接多环签名解决了多输入交易问题，在隐私保护区块链和数字货币等领域起到了核心作用，在票据支付、隐私投票、隐私拍卖等多种应用场景前景广泛。

当前，环签名算法经过近 20 年的发展，在普通环签名、可链接环签名、可链接多环签名等方面都取得了一定的技术成果，技术上已经逐步走向成熟，实现了高效（生成与验证低于 1ms）、紧致（常数或对数证明长度）、安全（不依赖可信初始化等额外假设）的特性，有多个备选的技术路线和经典算法，能够支撑实际的应用场景。因此，进行环签名相关密码算法应用推广研究具有较高的可行性与必要性。

此外，在适合区块链的应用场景下，面对监管审计政策法规，还需要将具备无条件可追踪性的可审计环签名考虑在内，实现隐私保护的同时兼容监管审计。项目组在可审计环签名法领域已经取得了一定的理论成果，进行相关算法应用推广研究具有较高的可行性。

5 主要研究内容

5.1 范围证明算法研究

5.1.1 相关密码学概念和安全性定义

零知识证明是一类证明系统(P,V), 其中证明者向验证者证明其知道某个知识, 但是不会泄露知识本身。 形式化定义是给定语言和关系R, 对于任意$x \in L$, 存在证据w满足$(x,w) \in R$, 在不透露x的前提下向验证者证明$x \in L$的证明系统。证明者和验证者交互的文本为$\langle P(x,w),V(x)\rangle$, $\langle P(x,w),V(x)\rangle = 1\ or\ 0$代表证明是正确(或错误)的。零知识证明系统的安全性定义包括完备性（completeness）, 合理性(soundness)以及零知识性(zero-knowledge)。

范围证明是零知识证明的一类，证明某个隐私数据值属于特定的区间。本文涉及的范围证明算法基于椭圆曲线和 Pedersen 承诺[7]，下面给出 Pedersen 承诺的定义和性质，其中g为椭圆曲线群$\mathbb{G}$的生成元（这里群元素运算用乘法表示）， h是随机椭圆曲线元素, 其离散对数未知。

定义 1（Pedersen 承诺）

对于隐私数据a的 Pedersen 承诺记为$c = g^x h^a$, 其中$x \in \mathbb{Z}_q^*$是混淆元。基于离散对数问题的困难性, Pedersen 承诺具备以下性质:

a) 隐藏性：任意（计算资源无限）的攻击者$\mathcal{A}$无法有效区分$c = g^x h^a$和 $c' = g^{x'} h^{a'}$。

b) 绑定性：任意 PPT 攻击者$\mathcal{A}$无法生成另外一个a'实现与c之间的绑定 $c = g^x h^a = g^{x'} h^{a'}$。

c) 同态性：给定$c_1 = g^x h^a, c_2 = g^y h^b$, 则$c_1 \cdot c_2 = g^{x+y} h^{a+b}$是关于隐私数据$a+b$的新承诺。

5.1.2 Borromean 范围证明算法及其优缺点

Borromean 范围证明算法由 Neother 等[8]在 2016 年提出，基于 Pedersen 承诺技术和 Borromean 多环签名算法[27]，将隐私数据$a \in [0,2^n-1]$进行二进制展开，并生成n个子公钥环进行多环签名的方式完成范围证明过程，实现不透露a精确值的前提下证明其范围合法性。Borromean 范围证明算法作为 Ring-CT 方案的核心算法被用于门罗币系统中。

a) 算法概述

1) 给定椭圆曲线群$(\mathbb{G},q)$，对于隐私承诺$c=g^x h^a$，其中$a \in [0,2^n-1]$，证明者随机选取$x_0,\cdots,x_{n-1} \in \mathbb{Z}_q^*$，计算$\beta = x - x_0 - \cdots - x_{n-1}$，同时将$a$二进制展开为$a = a_0 + \cdots + 2^i a_i + \cdots + 2^{n-1}a_{n-1}$，其中$a_i = 0,1$；

2) 证明者计算$c_i = g^{x_i} h^{2^i a_i}$，计算$c_i' = g^{x_i} h^{2^i a_i - 2^i}$；

3) 对于每个$i=0,\cdots,n-1$，计算子公钥环为$L_{PK_i} = \{c_i, c_i'\}$，对应的私钥为x_i；

4) 证明者计算公钥环$L_{PK} = \{L_0,\cdots,L_{n-1}\}$；

5) 证明者运行 Borromean 多环签名算法$\sigma = RSIG_{\mathrm{B}}(L_{PK},x_0,\cdots,x_{n-1},c,\beta)$，其中$RSIG_{\mathrm{B}}$代表 Borromean 多环签名算法；

6) 证明者输出关于隐私承诺c的 Borromean 区间证明结果$\pi = (\beta, L_{PK}, \sigma)$。

上面叙述的是 Borromean 范围证明算法的证明环节，验证环节可以参考文献[8]，这里不再赘述。Borromean 范围证明的设计创新点是利用多环签名算法实现隐私数据逐比特的$0,1$证明，证明和验证过程支持比特位并行运算。

根据文献[31]，可以基于 AOS’环签名[15]算法的思想，对原版 Borromean 多环签名算法进行改进，进而实现改进的 Borromean 范围证明算法，能够实现更高的证明生成与验证速度，并且更好支持并行加速。

b) 安全性

Borromean 范围证明算法基于 Borromean 多环签名算法（Borromean 多环签名算法本质上是多轮的Σ协议），在随机预言机模型下，具备完备性、合理性和零知识性，安全性基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。改进的 Borromean 范围证明算法[31]与原版 Borromean 具备相似的安全性，这里不再赘述。

c) 性能

Borromean 范围证明算法的证明生成、验证时间以及证明尺寸，关于隐私数据比特长度n都是呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。具体如下表所示：

表 1 Borromean 范围证明

算法	隐私数据长度	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
Borromean[8]	n	$4n$	$4n+1$	$(n,2n+2)$	是
改进 Borromean[31]	n	$3n$	$3n+1$	$(n,2n+2)$	是

d） 优缺点总结

优点：Borromean 范围证明算法构造简单，易于实现，易于分析，具备较高的安全性和可扩展性。

缺点：Borromean 范围证明算法证明生成和验证时间，以及证明尺寸关于隐私数据长度呈线性，在证明尺寸方面落后于 Bulletproofs 等后续算法。

5.1.3 Bulletproofs 范围证明算法及其优缺点

Bulletproofs 范围证明算法[11]由 Bünz 等在 2018 年提出，基于 Pedersen 承诺技术和内积论据系统，将隐私数据$a \in [0,2^n-1]$进行二进制展开，通过生成向量承诺以及执行相应Σ协议的方式完成范围证明过程，并通过向量递归压缩的技术缩短至对数证明尺寸，最后利用批处理技术实现了多条隐私数据共同进行范围证明的能力。Bulletproofs 范围证明算法作为 Borromean 范围证明的替代算法被用于 2019 年之后的门罗币系统中。

a） 算法概述

1） 给定椭圆曲线群$(\mathbb{G},q)$，对于隐私承诺$c = h^{\gamma}g^{a}$（承诺表达式与 Borromean 范围证明略有不同，但是理论上没有本质区别），其中$a \in [0,2^n-1]$，证明者将a二进制展开为$a = a_0 + \cdots + 2^i a_i + \cdots + 2^{n-1}a_{n-1}$，其中$a_i = 0,1$；

2） 证明者计算向量承诺 $A = h^{\alpha}g_0^{a_0}g_1^{a_1}\cdots g_{n-1}^{a_{n-1}}h_0^{a_0-1}h_1^{a_1-1}\cdots h_{n-1}^{a_{n-1}-1} = h^{\alpha}\mathrm{g}^{a}\mathrm{h}^{a-1^n}$并发送给验证者，其中$\mathrm{g} = (g_0,\cdots,g_{n-1}),\mathrm{h} = (h_0,\cdots,h_{n-1})$；

3） 证明者随机生成$\rho \leftarrow \mathbb{Z}_q$以及随机向量$\mathrm{s}_L,\mathrm{s}_R \leftarrow \mathbb{Z}_q^n$，计算$S = h^{\rho}\mathrm{g}^{\mathrm{s}_L}\mathrm{h}^{\mathrm{s}_R} \in \mathbb{G}$并发送给验证者；

4） 验证者随机生成$y,z \leftarrow \mathbb{Z}_q^*$并发送给证明者；

5） 证明者计算多项式$l(X),r(X) \in \mathbb{Z}_q^n[X]$以及$t(X) = \langle l(X),r(X)\rangle = t_0 + t_1X + t_2X^2 \in \mathbb{Z}_q[X]$，然后计算$t_1,t_2$的承诺值$T_1,T_2$，并将$T_1,T_2$发送给验证者；

6） 验证者随机生成$x \leftarrow \mathbb{Z}_q^*$并发送给证明者；

7） 证明者计算$\tau_x,\mu,\hat{t},\mathrm{l},\mathrm{r}$，然后调用递归向量内积论据系统，给出向量$\mathrm{l},\mathrm{r}$的内积论据$\mathrm{w}$，并将$\tau_x,\mu,\hat{t},\mathrm{w}$发送给验证者。

上面叙述的是 Bulletproofs 范围证明算法的证明环节，其中$\tau_x,\mu,\hat{t},\mathrm{l},\mathrm{r},\mathrm{w}$等元素的计算过程，以及验证环节可以参考文献[11]，这里不再赘述。Bulletproofs 范围证明的设计创新点是利用递归的向量内积论据系统和基于椭圆曲线的Σ协议，给出了对于隐私金额长度n具备对数尺寸$O(\log n)$的范围证明，支持多份范围证明批处理合并，并且证明和验证过程支持比特位并行运算，是当前应用最广泛的范围证明算法之一。

b） 安全性

Bulletproofs 范围证明算法基于向量内积论据系统和椭圆曲线Σ协议，在随机预言机模型下，具备完备性、特殊合理性和零知识性，安全性基于椭圆曲线离散对数和（扩展）DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

c） 性能

Bulletproofs 范围证明算法的证明生成和验证时间关于隐私数据比特长度n都是呈线性关系，证明尺寸关于隐私数据比特长度n呈对数关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。具体如下表所示：

表 2　Bulletproofs 范围证明

算法	隐私数据长度	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
Bulletproofs	n	$6n+O(\log n)$	$2n+O(\log n)$	$(2\log n+4,5)$	是

d） 优缺点总结

优点：Bulletproofs 范围证明算法理论清晰，易于实现，验证速度快，同时具备对数尺寸的证明长度，并能够有效支持多份证明合并和批处理验证，能够有效降低链上的存储压力，具备较高的效率、安全性和可扩展性。

缺点：Bulletproofs 范围证明算法证明生成时间较长，并且算法主体框架基于多轮Σ协议，递归向量内积论据系统模块本身也依赖于$O(\log n)$轮的Σ协议，在考虑多轮嵌套分叉引理带来的归约损失下，理论安全性评估难度较大，目前尚未调研到相关的实际安全性结论。

5.1.4 范围证明算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出范围证明算法的应用推广要求。

a） 功能要求

1） 范围证明算法应支持对任意长度的隐私数据，都能给出对于任意区间范围的零知识证明；

2） 范围证明算法应兼容国密椭圆曲线，可复用相关的随机数生成算法，倍点和点乘运算，以及相应的安全参数。

b） 安全性要求

1） 范围证明算法应满足基于标准模型或随机预言机模型下的完备性、合理性和零知识性等安全性；

2） 范围证明算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c） 性能要求

1） 范围证明算法应满足证明尺寸关于隐私数据长度呈对数或常数渐进关系；

2） 范围证明算法应满足证明和验证时间关于隐私数据长度呈线性或对数关系。

根据上述范围证明算法应用推广要求，课题组建议将 Bulletproofs 范围证明算法（国密曲线版本）作为范围证明应用推广算法。

5.2 可审计范围证明算法研究

5.2.1 相关密码学概念和安全性定义

在范围证明的基础上，对于可审计范围证明算法，对于任意 PPT 攻击者$\mathcal{A}$，考虑到系统中存在能够恢复出隐私信息的监管者，因此零知识性只对不掌握监管私钥的攻击者有效(掌握监管私钥，能够恢复隐私信息，零知识性不再成立)，与此同时完备性和坚固性的定义保持不变。 对于可审计范围证明算法，我们给出新的密码学原语-可审计性(Auditability)，使得监管者能够追踪到范围证明中蕴含的隐私数据值，对于任意 PPT 攻击者$\mathcal{A}$ ($\mathcal{A}$掌握监管私钥)，需要保障$\mathcal{A}$无法篡改隐私数据，同时无法构造虚假证明以逃离监管。 下面我们给出可审计范围证明的可审计性安全性定义：

定义 2（可审计性）

可审计范围证明的可审计性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化算法为$\mathcal{A}$提供公共参数，$\mathcal{A}$可以访问随机预言机$\mathcal{RO}$。 我们定义$\mathcal{A}$获胜，当$\mathcal{A}$能够关于隐私金额a生成金额承诺c，生成可审计范围证明结果$\pi(c)$，并且满足以下条件：

a） $\text{Verify}\ (c,\pi(c))=1$；

b） $\text{Trace}(\pi(c),trapdoors)\neq a$。

我们给出$\mathcal{A}$在审计攻击下的优势：

$$\text{Adv}_{\mathcal{A}}^{audit}=\Pr[\mathcal{A}\text{获胜}].$$

对于任意 PPT 攻击者$\mathcal{A}$，可审计范围证明具备可审计性当且仅当$\text{Adv}_{\mathcal{A}}^{audit}=negl(\lambda)$。

5.2.2 可审计 Borromean 范围证明算法及其优缺点

基于Borromean范围证明算法，可以构造具备监管审计功能的可审计范围证明算法，通过逐比特添加用于追踪审计的追踪密钥，监管者能够通过自身的审计公钥和每比特的追踪密钥，恢复出隐私数据的精确值，并且能够抵抗恶意监管者的潜在攻击。

5.2.2.1 算法概述

a） 初始化：

1） 给定椭圆曲线群$(\mathbb{G},q)$，系统随机选择生成元$g,h\in\mathbb{G}$，监管者随机选择$y\in\mathbb{Z}_q^*$作为审计私钥，计算审计公钥$h_0=g^y$，保留y作为审计私钥，系统公开公共参数$(\mathbb{G},q,g,h,h_0)$。

b） 证明：

1） 对于隐私承诺$c=g^x h^a$，其中隐私数据$a\in[0,2^n-1]$，证明者将a二进制展开为$a=a_0+\cdots+2^i a_i+\cdots+2^{n-1}a_{n-1}$，其中$a_i=0,1$；

2） 证明者随机选取$x_0,\cdots,x_{n-1}\in\mathbb{Z}_q^*$，计算$\beta=x-x_0-\cdots-x_{n-1}$；

3） 对于每个$i=0,\cdots,n-1$，证明者计算$c_i=g^{x_i}h^{2^i a_i}$，$c_i'=g^{x_i}h^{2^i a_i-2^i}$，记$L_i=(c_i,c_i')$为子承诺集，$L=\{L_0,\cdots,L_{n-1}\}$为承诺集；

4) 对于每个$i=0,\cdots,n-1$，证明者计算子追踪密钥$TK_i=h_0^{x_i}$，对于全部$i=0,\cdots,n-1$，就可以得到n个子追踪密钥集$\mathrm{TK}=\{TK_0,\cdots,TK_{n-1}\}$；

5) 对于每个$i=0,\cdots,n-1$，证明者计算金额标签$I_i=h^{x_i}$，得到金额标签集$\mathrm{I}=\{I_0,\cdots,I_{n-1}\}$；

6) 证明者计算随机数$e_0=H(L;\mathrm{TK};\mathrm{I};0)$，$e_1=H(L;\mathrm{TK};\mathrm{I};1)$；

7) 对于每个$i=0,\cdots,n-1$，证明者计算子公钥组为$L_{PK_i}=\left\{c_i\cdot TK_i^{e_0}\cdot I_i^{e_1},c_i'\cdot TK_i^{e_0}\cdot I_i^{e_1}\right\}$,对应的子私钥为$x_i$,进而得到公钥组为$L_{PK}=\{L_{PK_0},\cdots,L_{PK_{n-1}}\}$，并且计算新的椭圆曲线生成元为$g_1=gh_0^{e_0}h^{e_1}$；

8) 证明者运行 Borromean 多环签名（也可以使用文献[31]中改进的 Borromean 多环签名算法进一步提升效率），得到签名结果$\sigma=RSIG(L_{PK},x_0,\cdots,x_{n-1},c,\beta,\mathrm{TK},\mathrm{I})$，使用生成元为$g_1=gh_0^{e_0}h^{e_1}$；

9) 证明者输出隐私承诺c的可审计 Borromean 范围证明结果$\pi_{Abo}(c)=(\beta,L,\mathrm{TK},\mathrm{I},\sigma)$。

c) 验证：

1) 验证者计算$g^{\beta}\cdot\prod c_i=c$是否正确；

2) 对于全部$i=0,\cdots,n-1$，验证$\frac{c_i}{c_i'}=h^{2^i}$是否成立；

3) 验证者计算$e_0^*=H(L;\mathrm{TK};\mathrm{I};0)$，$e_1^*=H(L;\mathrm{TK};\mathrm{I};1)$；

4) 对于所有$i=0,\cdots,n-1$，验证者计算$L_{PK_i}^*=\left\{c_i\cdot TK_i^{e_0^*}\cdot I_i^{e_1^*},c_i'\cdot TK_i^{e_0^*}\cdot I_i^{e_1^*}\right\}$，得到$L_{PK}^*=\{L_{PK_1}^*,\cdots,L_{PK_{n-1}}^*\}$；

5) 验证者检验 Borromean 多环签名σ的正确性，验证过程使用生成元为$g_1^*=gh_0^{e_0}h^{e_1}$。

d) 审计：

1) 对于每一位$i=0,\cdots,n-1$，监管者计算$TK_i^{y^{-1}}$；

2) 如果$c_i=TK_i^{y^{-1}}$，则输出$a_i=0$；

3) 如果 $c_i'=TK_i^{y^{-1}}$，则输出$a_i=1$；

4) 监管者计算$a=a_0+\cdots+2^ia_i+\cdots+2^{n-1}a_{n-1}$作为对于隐私数据的审计结果。

可审计 Borromean 范围证明算法基于原始的 Borromean 范围证明，在公钥集生成和签名环节通过随机化嵌入的方法添加追踪密钥TK和金额标签I，在实现追踪监管功能的同时，可以抵抗恶意监管者的潜在攻击。

5.2.2.2 安全性

可审计 Borromean 范围证明算法，在随机预言机模型下，具备完备性、特殊合理性、零知识性以及可审计性，其中零知识性仅对不掌握审计私钥的多项式时间攻击者成立，而完备性、特殊合理性和可审计性对于掌握审计私钥的攻击者仍保持成立。算法的安全性基于椭圆曲线离散对数和（扩展）DDH 等困难问题，以及哈希函数的伪随机性，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.2.2.3 性能

可审计Borromean范围证明算法的证明生成和验证以及审计所需时间关于隐私数据比特长度n都是呈线性关系，证明尺寸关于隐私数据比特长度n同样呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。我们在性能估计中，使用文献[31]中改进的 Borromean 多环签名算法作为模块，具体如下表所示：

表 3　可审计 Borromean 范围证明

算法	隐私数据长度	生成	验证	审计	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
可审计 Borromean	n	$7n+2$	$5n+3$	n	$(3n,2n+2)$	是

根据上表可以看出，与经典的 Borromean 范围证明算法相比，可审计 Borromean 范围证明算法在证明尺寸、生成和验证计算量上有不到一倍的增加，但是具备额外的监管审计功能，并能抵抗恶意监管者的潜在攻击。

5.2.2.4 优缺点总结

优点：可审计 Borromean 范围证明算法设计方法简单，易于实现，使用模块化的构造方法，底层的多环签名算法模块可替换，具备较高的安全性和可扩展性。

缺点：可审计 Borromean 范围证明算法证明和验证时间关于隐私数据长度n呈线性，并且证明尺寸也为线性，相对经典 Borromean 范围证明算法（无监管审计）有一定的效率损失。

5.2.3 可审计 Bulletproofs 范围证明算法及其优缺点

基于 Bulletproofs 范围证明算法和 Pedersen 承诺技术，将隐私数据$a \in [0,2^n-1]$进行二进制展开，结合逐比特追踪密钥生成方法，能够构造可审计 Bulletproofs 范围证明算法，使得原有 Bulletproofs 范围证明算法具备监管审计功能。

5.2.3.1 算法概述

a) 初始化：

1) 给定椭圆曲线群$(\mathbb{G},q)$，给定公共参数$\mathrm{g}=(g_0,\cdots,g_{n-1})$，监管者随机选择$y_0,\cdots,y_{\frac{n}{2}-1} \in \mathbb{Z}_q^*$作为审计私钥（假定$n$为偶数），计算审计公钥$h_{2i}=g_{2i}^{y_i},h_{2i+1}=g_{2i+1}^{y_i},i=0,\cdots,\frac{n}{2}-1$，得到$\mathrm{h}=(h_0,\cdots,h_{n-1})$，系统公开公共参

数$(\mathbb{G}, q, g, h, \mathrm{g}, \mathrm{h})$。

b) 证明：

1) 对于隐私承诺$c = h^{\gamma} g^{a}$（承诺表达式与可审计 Borromean 范围证明略有不同，但是理论上没有本质区别），其中$a \in [0, 2^{n} - 1]$，证明者将a二进制展开为$a = a_0 + \cdots + 2^{i} a_i + \cdots + 2^{n-1} a_{n-1}$，其中$a_i = 0,1$；

2) 证明者计算向量承诺$A = h^{\alpha} g_0^{a_0} g_1^{a_1} \cdots g_{n-1}^{a_{n-1}} h_0^{a_0 - 1} h_1^{a_1 - 1} \cdots h_{n-1}^{a_{n-1} - 1} = h^{\alpha} \mathrm{g}^{a} \mathrm{h}^{a - 1^{n}}$并发送给验证者，其中$\mathrm{g} = (g_0, \cdots, g_{n-1}), \mathrm{h} = (h_0, (-h_{n-1})$；

3) 对于每个$j = 0, \cdots, \frac{n}{2} - 1$，证明者计算$TK_{2j} = g_{2j}^{\alpha - a_{2j}} g_{2j+1}^{\alpha - a_{2j+1}}, TK_{2j+1} = h_{2j}^{-\alpha - a_{2j} + 1} h_{2j+1}^{-\alpha - a_{2j+1} + 1}$，一共计算产生$n$个$TK_i$；

4) 对于每个$TK_i, i = 0, \cdots, n-1$，证明者计算$\pi(TK_i)$的承诺证明，然后证明者计算$A \cdot \prod_{i \in [0, n-1]} TK_i = \left(\frac{h \prod_{i \in [0, n-1]} g_i}{\prod_{i \in [0, n-1]} h_i} \right)^{\alpha}$，并给出$\pi(A \cdot \prod_{i \in [0, n-1]} TK_i)$的承诺证明，将全部$\pi(TK_i)$和$\pi(A \cdot \prod_{i \in [0, n-1]} TK_i)$作为所有$TK_i$的合法性证明；

5) 证明者随机生成$\rho \leftarrow \mathbb{Z}_q$以及随机向量$\mathrm{s}_L, \mathrm{s}_R \leftarrow \mathbb{Z}_q^{n}$，计算$S = h^{\rho} g^{\mathrm{s}_L} h^{\mathrm{s}_R} \in \mathbb{G}$并发送给验证者；

6) 验证者随机生成$y, z \leftarrow \mathbb{Z}_q^{*}$并发送给证明者；

7) 证明者计算多项式$l(X), r(X) \in \mathbb{Z}_q^{n}[X]$以及$t(X) = \langle l(X), r(X) \rangle = t_0 + t_1 X + t_2 X^2 \in \mathbb{Z}_q[X]$，然后计算$t_1, t_2$的承诺值$T_1, T_2$，并将$T_1, T_2$发送给验证者；

8) 验证者随机生成$x \leftarrow \mathbb{Z}_q^{*}$并发送给证明者；

9) 证明者计算$\tau_x, \mu, \hat{t}, \mathrm{l}, \mathrm{r}$，然后调用递归向量内积论据系统，给出向量$\mathrm{l}, \mathrm{r}$的内积论据$\mathrm{w}$，并将$\tau_x, \mu, \hat{t}, \mathrm{w}, A, S, \{TK_i\}_{i=0,\cdots,n-1}, \{\pi(TK_i)\}_{i=0,\cdots,n-1}, \pi\left(A \cdot \prod_{i \in [0, n-1]} TK_i\right)$等元素作为可审计 Bulletproofs 算法输出（相比于 Bulletproofs，新增加的输出为$\{TK_i\}_{i=0,\cdots,n-1}, \{\pi(TK_i)\}_{i=0,\cdots,n-1}, \pi(A \cdot \prod_{i \in [0, n-1]} TK_i)$。

c) 验证：

1) 验证者检验全部$\{\pi(TK_i)\}_{i=0,\cdots,n-1}$的合法性；

2) 验证者计算$A \cdot \prod_{i \in [0, n-1]} TK_i$，并检验全部$\pi\left(A \cdot \prod_{i \in [0, n-1]} TK_i\right)$的合法性；

3) 验证者完成其他的 Bulletproofs 证明流程，得到可审计 Bulletproofs 范

围证明算法的验证结果。

d) 审计：

1) 对于每一个$j=0,\cdots,\frac{n}{2}-1$，监管者计算$TK_{2j+1}\cdot TK_{2j}^{y_j}$；

2) 如果$TK_{2j+1}\cdot TK_{2j}^{y_j}=h_{2j}h_{2j+1}$，则输出$(a_{2j},a_{2j+1})=(0,0)$；

3) 如果$TK_{2j+1}\cdot TK_{2j}^{y_j}=h_{2j}^{-1}h_{2j+1}$，则输出$(a_{2j},a_{2j+1})=(1,0)$；

4) 如果$TK_{2j+1}\cdot TK_{2j}^{y_j}=h_{2j}h_{2j+1}^{-1}$，则输出$(a_{2j},a_{2j+1})=(0,1)$；

5) 如果$TK_{2j+1}\cdot TK_{2j}^{y_j}=h_{2j}^{-1}h_{2j+1}^{-1}$，则输出$(a_{2j},a_{2j+1})=(1,1)$；

6) 输出$(a_0,\cdots,a_{n-1})$，计算金额$a=a_0+\cdots+2^i a_i+\cdots+2^{n-1}a_{n-1}$作为审计结果。

上面叙述中涉及原始 Bulletproofs 范围证明算法的证明和验证环节，其中$\tau_x,\mu,\hat{t},\mathrm{l},\mathrm{r},\mathrm{w}$等元素的计算过程，以及验证环节可以参考文献[11]，这里不再赘述。可审计 Bulletproofs 范围证明算法的优势在于较好的复用了 Bulletproofs 范围证明框架，具备较短的证明尺寸和验证时间，缺陷在于尚无法抵抗恶意攻击者，如果将协议改为抗恶意攻击版本则会带来过高的尺寸和性能损失，可用性会低于可审计 Borromean 范围证明算法。

5.2.3.2 安全性

可审计 Bulletproofs 范围证明算法基于向量内积论据系统和椭圆曲线Σ协议，在随机预言机模型下，具备完备性、特殊合理性、零知识性以及可审计性，安全性基于椭圆曲线离散对数和（扩展）DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.2.3.3 性能

可审计 Bulletproofs 范围证明算法的证明生成和验证以及审计所需时间关于隐私数据比特长度n都是呈线性关系，证明尺寸关于隐私数据比特长度n呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。具体如下表所示：

表 4 可审计 Bulletproofs 范围证明

算法	隐私数据长度	生成	验证	审计	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
可审计 Bulletproofs	n	$10n+O(\log n)$	$5n+O(\log n)$	$\frac{n}{2}$	$(n+2\log n+4{,}5+2n)$	是

5.2.4 可审计范围证明算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出可审计范围证明算法的

应用推广要求。

a) 功能要求
   1) 可审计范围证明算法应支持对任意长度的隐私数据，都能给出对于任意区间范围的零知识证明；
   2) 可审计范围证明算法应兼容国密椭圆曲线，可复用相关的随机数生成算法、倍点和点乘运算，以及相应的安全参数；
   3) 可审计范围证明算法应支持无条件监管审计功能，监管者利用监管私钥能够恢复出任意可审计范围证明中的隐私数据值。
b) 安全性要求
   1) 可审计范围证明算法应满足基于标准模型或随机预言机模型下的完备性、合理性、零知识性、和可审计性等安全性，并能够抵抗恶意监管者的攻击；
   2) 可审计范围证明算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。
c) 性能要求
   1) 可审计范围证明算法应满足证明尺寸关于隐私数据长度呈对数或常数渐进关系；
   2) 可审计范围证明算法应满足证明和验证时间关于隐私数据长度呈线性或对数关系。

根据上述可审计范围证明算法应用推广要求，课题组建议将可审计 Borromean 范围证明算法（国密曲线版本）作为可审计范围证明应用推广算法。

5.3 环签名算法研究

5.3.1 相关密码学概念和安全性定义

环签名算法是一类特殊的数字签名算法，签名人能够自主选取一个公钥集合完成签名，验证者无法定位集合中的实际签名人，实现了签名人身份的隐私性。

环签名算法的安全性包含匿名性（Anonymity）和不可伪造性（Unforgeability），再给出安全性定义之前，我们介绍攻击者能够被授权访问的几种预言机类型，实际上，攻击者能够访问以下几种预言机：

a) $c \leftarrow \mathcal{RO}(a)$: 称为随机预言机（Random Oracle），根据输入a，随机预言机返回随机输出；
b) $PK_i \leftarrow \mathcal{JO}(\perp)$: 称为添加预言机（Joining Oracle），收到询问，根据公钥生成算法向系统中添加新的用户公钥，返回新用户的公钥PK_i；
c) $SK_i \leftarrow \mathcal{CO}(PK_i)$: 称为破解预言机（Corruption Oracle），根据输入公钥PK_i（是由$\mathcal{JO}$返回的合法公钥），返回对应的私钥SK_i。
d) $\sigma \leftarrow \mathcal{SO}(PK_\kappa, \mu, L_{PK})$: 称为签名预言机（Signing Oracle），根据输入的用户公钥集合L_{PK}以及实际签名的公钥PK_κ，待签名消息μ，返回合法的环签名σ。

环签名算法的不可伪造性定义如下：

定义 3（不可伪造性）

环签名算法的不可伪造性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化并向攻击者$\mathcal{A}$提供初始化公共参数，$\mathcal{A}$被授权访问预言机$\mathcal{RO}$，$\mathcal{JO}$，$\mathcal{CO}$和$\mathcal{SO}$。 $\mathcal{A}$获胜当且仅当他能够成功伪造环签名$(\sigma^*, L_{PK}^*, \mu^*)$，并且满足以下条件：

a) $\text{Verify}(\sigma^*, L_{PK}^*, \mu^*) = 1$。
b) 全部$PK_i \in L_{PK}^*$是由$\mathcal{A}$访问$\mathcal{JO}$返回的。

c） 全部$PK_i \in L_{PK}^*$都没有通过$\mathcal{A}$访问$\mathcal{CO}$得出。

d） $(\sigma^*, L_{PK}^*, \mu^*)$不是由$\mathcal{A}$访问$\mathcal{SO}$得出。

我们给出攻击者$\mathcal{A}$在伪造攻击中的优势:

$$\mathrm{Adv}_{\mathcal{A}}^{forge} = \Pr[\mathcal{A}\text{获胜}].$$

对于任意 PPT 攻击者$\mathcal{A}$,环签名满足不可伪造性当且仅当$\mathrm{Adv}_{\mathcal{A}}^{forge} = negl(\lambda)$。

定义 4（匿名性）

环签名的匿名性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化并向攻击者$\mathcal{A}$提供初始化公共参数，$\mathcal{A}$被授权访问预言机$\mathcal{HO}$，$\mathcal{JO}$和$\mathcal{CO}$。

攻击者$\mathcal{A}$提供测试公钥集合$L_{PK} = \{PK_1, \cdots, PK_n\}$,模拟器$\mathcal{S}$随机选取$\kappa \in \{1, \cdots, n\}$，计算签名$\sigma = Rsign(SK_\kappa, \mu, L_{PK})$并且发送$\sigma$给$\mathcal{A}$，其中$SK_\kappa$是与公钥$PK_\kappa \in L_{PK}$相关的私钥，然后$\mathcal{A}$公布其猜测$\kappa^* \in \{1, \cdots, n\}$。 攻击者$\mathcal{A}$获胜当且仅当他猜中$\kappa^* = \kappa$。 其中对于测试公钥集$L_{PK}$，需要满足$L_{PK}$中的全部公钥由$\mathcal{A}$访问$\mathcal{JO}$得到，而不是由$\mathcal{A}$访问$\mathcal{CO}$得到。

我们给出攻击者$\mathcal{A}$在匿名性攻击中的优势：

$$\mathrm{Adv}_{\mathcal{A}}^{anon} = |\Pr[\kappa^* = \kappa] - \frac{1}{n}|.$$

对于任意 PPT 攻击者$\mathcal{A}$，环签名满足匿名性当且仅当$\mathrm{Adv}_{\mathcal{A}}^{anon} = negl(\lambda)$。

5.3.2 AOS 环签名算法及其优缺点

AOS 环签名算法由 Abe 等[15]在 2002 年提出，基于多轮循环的 Schnorr 签名算法框架，通过多轮 Fiat-Shamir 变换的Σ-协议实现环签名功能。在[15]的附录中同样给出了 AOS’算法，具备相同的尺寸和更高的性能，同时具备并行优化的空间。

5.3.2.1 算法概述

AOS 环签名算法（这里我们给出各方公钥生成元不同的情形，是一种更一般的情形）

a） 签名：

1） 给定椭圆曲线群$(\mathbb{G}, q)$，用户P_κ随机选取$x_\kappa \in \mathbb{Z}_q^*$，计算$g_\kappa^{x_\kappa}$，公私钥对为$(PK_\kappa, SK_\kappa) = (g_\kappa^{x_\kappa}, x_\kappa)$;

2） 当用户P_κ需要对于消息μ执行环签名时，他随机选取其他$n-1$个用户公钥，与自己的公钥PK_κ一起组成公钥集合$L_{PK} = \{PK_1, \cdots, PK_n\}$，其中$PK_\kappa \in L_{PK}$并且$\kappa \in \{1, \cdots, n\}$，其他公钥形为$PK_i = g_i^{x_i}$，然后他执行如下操作;

3） P_κ均匀随机选取$r_\kappa \in \mathbb{Z}_q^*$，计算$c_{\kappa+1} = H(g_\kappa^{r_\kappa}, L_{PK}, \mu)$;

4） 对于$i = \kappa+1, \cdots, n, 1, \cdots, \kappa-1$，$P_\kappa$随机选取$z_i \in \mathbb{Z}_q^*$，并且计算

$$c_{i+1} = H(g_i^{z_i}/(PK_i)^{c_i}, L_{PK}, \mu);$$

5） P_κ计算$z_\kappa = r_\kappa + x_\kappa c_\kappa$;

6） 签名者P_κ输出签名结果$\sigma = (c_1; z_1, \cdots, z_n)$。

b） 验证：

1） 对于环签名(μ, L_{PK}, σ), 对$i = 1, \cdots, n$, 验证者计算

$$c_{i+1}^* = H(g_i^{z_i}/(PK_i)^{c_i^*}, L_{PK}, \mu),$$

其中$c_1 = c_1^*$, 然后检查$c_1 =_? c_{n+1}^*$是否相等, 如果全部通过则输出 1, 反之输出 0。

AOS’环签名算法（AOS’环签名算法只能支持各方公钥生成元相同的情形）

a） 签名：

1） 给定椭圆曲线群$(\mathbb{G}, q)$，用户P_κ随机选取$x_\kappa \in \mathbb{Z}_q^*$，计算$g^{x_\kappa}$，公私钥对为$(PK_\kappa, SK_\kappa) = (g^{x_\kappa}, x_\kappa)$;

2） 当用户P_κ需要对于消息μ执行环签名时, 他随机选取其他$n-1$个用户公钥, 与自己的公钥PK_κ一起组成公钥集合$L_{PK} = \{PK_1, \cdots, PK_n\}$, 其中$PK_\kappa \in L_{PK}$并且$\kappa \in \{1, \cdots, n\}$, 其他公钥形为$PK_i = g^{x_i}$, 然后他执行如下操作；

3） 对于$i = 1, \cdots, \kappa-1, \kappa+1, \cdots, n$，$P_\kappa$均匀随机选取$c_i$，$P_\kappa$随机选取$\alpha \in \mathbb{Z}_q^*$，计算$R = g^\alpha \prod_{i \neq \kappa} PK_i^{c_i}$，以及$c = H(R, L_{PK}, \mu)$ ；

4） P_κ计算$c_\kappa = c - \sum_{i \neq k} c_i$；

5） P_κ计算$z = \alpha - x_\kappa c_\kappa$；

6） 签名者P_κ输出签名结果$\sigma = (z; c_1, \cdots, c_n)$。

b） 验证：

1） 对于环签名(μ, L_{PK}, σ), 对$i = 1, \cdots, n$, 验证者计算$R^* = g^z \sum_{i=1}^n PK_i^{c_i}$, 然后检查$\sum_{i=1}^n c_i =_? H(R^*, L_{PK}, \mu)$是否相等, 如果全部通过则输出 1, 反之输出 0。

AOS 和 AOS’环签名算法的构造思路相似，使用 Fiat-Shamir 变换和Σ-协议实现了环签名的技术功能，结构简单，安全性易于分析。

5.3.2.2 安全性

AOS 和 AOS’环签名算法，在随机预言机模型下，具备不可伪造性和匿名性，安全性基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.3.2.3 性能

AOS 和 AOS’环签名算法的签名生成、验证时间以及证明尺寸，关于公钥集元素数量n都是呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。具体如下表所示：

表 5 AOS&AOS’环签名算法

算法	公钥集元素数	生成	验证	尺寸$(\mathbb{G}, \mathbb{Z}_q^*)$	支持并行

AOS	n	$2n-1$	$2n$	$(0,n+1)$	否
AOS’	n	n	$n+1$	$(0,n+1)$	是

5.3.3 GK 环签名算法及其优缺点

GK 环签名算法[19]由 Groth 和 Kohlweiss 在 2015 年提出，算法基于Σ-协议和多项式系数嵌入实现了高效的环签名算法框架，该算法关于公钥集元素数量n具备线性的签名和验证渐进时间，具备渐进意义下的对数签名尺寸，但是尺寸和性能都低于 Ring-CT3.0 环签名算法。

5.3.4 Ring-CT3.0 环签名算法及其优缺点

a) 算法概述

Ring-CT3.0[20]由 Yuen 等在 2019 年提出，该方案实现了基于 Bulletproofs 框架的隐私交易系统，该方案可以抽象出新的环签名算法，基于与 Bulletproofs 相似的多轮Σ-协议框架，具备对数尺寸的签名尺寸和线性的签名和验证时间，效率强于 GK 算法，同时具备并行优化的空间。

b) 安全性

Ring-CT3.0 环签名算法，在随机预言机模型下，具备不可伪造性和匿名性，安全性基于椭圆曲线离散对数和DDH等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

c) 性能

Ring-CT3.0 环签名算法的签名生成、验证时间以及证明尺寸，关于公钥集元素数量n都是呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。具体如下表所示：

表 6　GK&Ring-CT3.0 环签名算法

算法	公钥集元素数	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
GK	n	$O(n\log n)$	$O(n)$	$(4\log n, 3\log n+1)$	否
Ring-CT3.0	n	$O(n)$	$O(n)$	$(2\log n+7, 7)$	是

5.3.5 普通环签名算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出普通环签名算法的应用推广要求。

a) 功能要求

1) 普通环签名算法应支持对任意公钥集合规模，都能给出关于该公钥集合的环签名；
2) 普通环签名算法应兼容国密椭圆曲线，可复用相关的随机数生成算法，倍点和点乘运算，以及相应的安全参数。

b) 安全性要求

1) 普通环签名算法应满足基于标准模型或随机预言机模型下的不可伪造性和

匿名性；

2) 普通环签名算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c) 性能要求

1) 普通环签名算法宜满足签名尺寸关于公钥集合元素数量呈对数或常数渐进关系，如果呈线性关系，则需满足在常见应用参数下尺寸较短；

2) 普通环签名算法应满足签名生成和验证时间关于公钥环元素数量呈线性或对数关系。

根据上述环签名算法应用推广要求，课题组建议将 AOS、AOS'、Ring-CT3.0 环签名算法（国密曲线版本）作为普通环签名应用推广算法。

5.4 可链接环签名算法研究

5.4.1 相关密码学概念和安全性定义

在环签名算法的基础上，可链接环签名一方面满足环签名算法无法获取签名人公钥的匿名功能，同时需要满足存在签名标签和链接算法（Link），使得相同公钥的不同签名行为可被识别。在区块链系统中，可链接环签名能够解决防止区块链数字货币多重支付的攻击行为。

除了匿名性和不可伪造性，可链接环签名的安全性定义还包含可链接性（linkability）与不可诽谤性（nonslanderability）。

定义 5（可链接性）

可链接环签名的可链接性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化并向攻击者$\mathcal{A}$提供初始化公共参数，$\mathcal{A}$被授权访问预言机$\mathcal{RO}$，$\mathcal{JO}$，$\mathcal{CO}$和$\mathcal{SO}$。攻击者$\mathcal{A}$获胜当且仅当他能成功伪造k个可链接环签名$(\sigma_i, L_{PK}^i, \mu_i), i = 1, \cdots, k$，并且满足以下条件：

a) 全部σ_i不是由$\mathcal{A}$访问$\mathcal{SO}$得到；

b) 全部L_{PK}^i中的公钥都是由$\mathcal{A}$访问$\mathcal{JO}$得到；

c) $\text{Verify}(\sigma_i, L_{PK}^i, \mu_i) = 1, i = 1, \cdots, k$；

d) $\mathcal{A}$访问$\mathcal{CO}$的次数小于k；

e) 对于全部$i, j \in \{1, \cdots k\}$以及$i \neq j$，$link((\sigma_i, L_{PK}^i, \mu_i), (\sigma_j, L_{PK}^j, \mu_j)) = unlinked$。

我们给出攻击者$\mathcal{A}$在链接攻击中的优势：

$$\text{Adv}_{\mathcal{A}}^{Link} = \Pr[\mathcal{A}\text{获胜}].$$

对于任意 PPT 攻击者$\mathcal{A}$，可链接环签名满足可链接性当且仅当$\text{Adv}_{\mathcal{A}}^{link} = negl(\lambda)$。

定义 6（不可诽谤性）

可链接环签名的不可诽谤性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化并向攻击者$\mathcal{A}$提供初始化公共参数，$\mathcal{A}$被授权访问预言机$\mathcal{RO}$，$\mathcal{JO}$，$\mathcal{CO}$和$\mathcal{SO}$。攻击者$\mathcal{A}$提供测试公钥集合$L_{PK} = \{PK_1, \cdots, PK_n\}$，待签名信息$\mu$以及签

名公钥$PK_\kappa \in L_{PK}$给模拟器$\mathcal{S}$，$\mathcal{S}$返回相关的签名结果$\sigma = Rsign(SK_\kappa, \mu, L_{PK})$给$\mathcal{A}$。 $\mathcal{A}$获胜当且仅当他能够返回新的签名$(\sigma^*, L_{PK}^*, \mu^*)$，并且满足以下条件：

a) $\text{Verify}(\sigma^*, L_{PK}^*, \mu^*) = 1$；
b) 全部L_{PK}中的公钥都是由$\mathcal{A}$访问$\mathcal{JO}$得到；
c) PK_κ没有被$\mathcal{A}$作为输入访问$\mathcal{CO}$；
d) PK_κ没有被$\mathcal{A}$作为输入访问$\mathcal{SO}$；
e) $link((\sigma, L_{PK}, \mu), (\sigma^*, L_{PK}^*, \mu^*)) = linked$。

我们给出攻击者$\mathcal{A}$在诽谤攻击中的优势：

$$\text{Adv}_{\mathcal{A}}^{slander} = \Pr[\mathcal{A}\text{获胜}].$$

对于任意 PPT 攻击者$\mathcal{A}$，可链接环签名满足不可诽谤性当且仅当$\text{Adv}_{\mathcal{A}}^{slander} = negl(\lambda)$。

5.4.2 CryptoNote 可链接环签名算法及其优缺点

2013 年，Saberhagen 等[4]给出了基于循环Σ-协议的可链接环签名算法，并用于初期版本的门罗币系统中，具备不可伪造性、匿名性、可链接性与不可诽谤性，安全性基于椭圆曲线困难问题和随机预言机模型，算法的签名标签形式为$tag = H(g^{x_\kappa})^{x_\kappa}$，签名尺寸，签名和验证时间关于公钥集合元素数量呈线性，但是尺寸和性能弱于 MLSAG 算法。

5.4.3 MLSAG 可链接环签名算法及其优缺点

2016 年，Neother 等[8]提出了新的可链接环签名算法，基于与 CryptoNote 算法相似的框架和安全性，具备更短的签名尺寸，安全性基于椭圆曲线困难问题和随机预言机模型，算法的签名标签形式为$tag = H(g^{x_\kappa})^{x_\kappa}$，签名尺寸，签名和验证时间关于公钥集合元素数量呈线性，但是尺寸和性能弱于模块化可链接环签名算法（modular-LRS）。

5.4.4 模块化可链接环签名算法（modular-LRS）及其优缺点

5.4.4.1 算法概述

模块化可链接环签名算法（modular-LRS）基于全新的模块化构造思路，算法支持固定生成元签名标签以及非一致生成元签名标签。模块化可链接环签名算法的特性有以下几方面：

a) 模块化构造思路：不同于传统可链接环签名算法在签名环节添加签名标签并完成签名过程的特征，modular-LRS 算法在公钥集生成环节就完成签名标签的添加，计算新的环签名公钥集L_{RPK}，随后直接调用普通环签名算法对于L_{RPK}执行签名操作，实现了模块化的构造方法。可以选用 AOS，AOS'等普通环签名算法，也可以使用更短(更快)的其他普通环签名模块（例如 GK，Ring-CT3.0 等）作为普通环签名模块备选，算法选项多样，易于迁移；
b) 随机化嵌入方法：通过计算基于环签名公钥集和签名标签的随机数计算新的普通环签名公钥集，实现签名标签随机化嵌入到环签名公钥集中，进而调用普通环签名算法完成可链接环签名；
c) 可变生成元模式：在算法执行过程中，计算新的生成元（形如gh^e），才能完成后续的签名和验签操作；在非一致生成元 modular-LRS 算法中，需要在每个公

钥PK_i对应的位置使用互异的生成元gh_i^e完成签名和验签，能够实现更高的运算效率。

下面简述两类 modular-LRS 签名算法：

固定生成元签名标签版本：

a) Setup：系统选定素数阶椭圆曲线群$(\mathbb{G},q)$，随机选择生成元$g_1 \in \mathbb{G}$，然后再随机选择生成元$g_2 \in \mathbb{G}$，可选的，通过 hash to point 生成g_2，即$g_2 = H_1(g_1)$，系统公开公共参数$(\mathbb{G},q,g_1,g_2)$。

b) Gen：每一个用户P随机选取$\mathrm{x} \in \mathbb{Z}^*$，计算公钥$\mathrm{PK} = \mathrm{g_1^x}$，私钥为x。

c) Sign：

1) 用户P_i随机选取$n-1$个其它用户，将包含自己在内的n个公钥，组成可链接环签名公钥集合$L_{PK} = \{PK_1,\cdots,PK_n\} = \{g_1^{x_1},\cdots,g_1^{x_n}\}$，此时假定用户$P_i$的公私钥为$(g_1^{x_i},x_i)$；

2) 用户P_i计算签名标签$tag = I = g_2^{x_i}$，然后计算$e = H(L_{PK},I)$；

3) 用户计算普通环签名公钥集

$$L_{RPK} = \{PK_1 \cdot I^e,\cdots,PK_n \cdot I^e\} = \{g_1^{x_1}g_2^{ex_i},\cdots,g_1^{x_n}g_2^{ex_i}\};$$

4) 对于消息m，用户P_i计算普通环签名$\tau = RSIG(x_i,L_{RPK},m)$（普通环签名以$g_1g_2^e$为生成元），实际上，相应的签名公钥为$(g_1g_2^e)^{x_j}$，假定第$j$个用户为真实的签名用户），可使用 AOS 或 AOS’环签名作为普通环签名组件；

5) 用户输出可链接环签名为$\sigma = (L_{PK},m,\tau,I)$。

d) Verify：

1) 验证者计算$e = H(L_{PK},I)$，然后检验$L_{RPK} = \{PK_1 \cdot I^e,\cdots,PK_n \cdot I^e\} = \{g_1^{x_1}g_2^{ex_i},\cdots,g_1^{x_n}g_2^{ex_i}\}$是否计算正确；

2) 验证者检验普通环签名$\tau = RSIG(x_i,L_{RPK},m)$的正确性，以$g_1g_2^{e_2}$为生成元(或称基元素均可)；

全部验证通过后，验证者通过可链接环签名$\sigma = (L_{PK},m,\tau,I)$的正确性（合法性）验证。

e) Link：

1) 对于可链接环签名$\sigma = (L_{PK},m,\tau,I)$以及$\sigma' = (L'_{PK},m',\tau',I')$，如果$I = I'$则输出 Linked，反之输出 unlinked。

非一致生成元签名标签版本：

a) Setup：系统选定素数阶椭圆曲线群$(\mathbb{G},q)$，随机选择生成元$g \in \mathbb{G}$，系统公开公共参数$(\mathbb{G},q,g)$。

b) Gen：每一个用户P随机选取$\mathrm{x} \in \mathbb{Z}^*$，计算公钥$\mathrm{PK} = \mathrm{g^x}$，私钥为$\mathrm{SK} = \mathrm{x}$。

c) Sign：

1) 用户P_i随机选取$n-1$个其它用户，将包含自己在内的n个公钥，组成可链接环签名公钥集合$L_{PK}=\{PK_1,\cdots,PK_n\}=\{g^{x_1},\cdots,g^{x_n}\}$，此时假定用户$P_i$的公私钥为$(g^{x_i},x_i)$；

2) 对于$j=1,\cdots,n$，用户P_i计算$h_j=H_1(PK_j)$，随后用户计算签名标签$Tag=I=H_1(PK_i)^{x_i}=h_i^{x_i}$，然后计算$e=H(L_{PK},I)$；

3) 用户计算普通环签名公钥集

$$L_{RPK}=\{PK_1\cdot I^e,\cdots,PK_n\cdot I^e\}$$
$$=\left\{g^{x_1}h_i^{ex_i},\cdots,g^{x_n}h_i^{ex_i}\right\}=\{RPK_i\};$$

4) 对于消息m，用户P_i随机选取$r_i\in\mathbb{Z}_q^*$，计算$c_{i+1}=H((gh_i^e)^{r_i},L_{RPK},m)$；

5) 对于$j=i+1,\cdots,n,1,\cdots,i-1$，用户$P_i$随机选取$z_j\in\mathbb{Z}_q^*$，并计算

$$c_{j+1}=H((gh_j^e)^{z_j}/\left(RPK_j\right)^{c_j},L_{PK},m),$$

每个公钥RPK_i对应的位置i使用生成元gh_i^e；

6) 用户P_i计算$z_i=r_i+x_ic_i$；

7) 用户P_i得到$\tau=(c_1,z_1,\cdots,z_n)$；

8) 用户输出可链接环签名为$\sigma=(L_{PK},m,\tau,I)$。

d) Verify：

1) 验证者计算$e=H(L_{PK},I)$，然后计算

2) $L_{RPK}=\{PK_1\cdot I^e,\cdots,PK_n\cdot I^e\}=\{RPK_i\}$;

3) 对于$j=1,\cdots,n$，验证者计算$h_j=H_1(PK_j)$；

4) 对于$j=1,\cdots,n-1$，验证者计算$c_{j+1}=H((gh_j^e)^{z_j}/\left(RPK_j\right)^{c_j},L_{PK},m)$；

5) 验证者计算$c_1=H((gh_n^e)^{z_n}/(RPK_n)^{c_n},L_{PK},m)$是否成立，全部验证通过后，验证者通过可链接环签名$\sigma=(L_{PK},m,\tau,I)$的正确性（合法性）验证。

e) Link：

1) 对于可链接环签名$\sigma=(L_{PK},m,\tau,I)$以及$\sigma'=(L'_{PK},m',\tau',I')$，如果$I=I'$则输出 Linked，反之输出 unlinked。

5.4.4.2 安全性

模块化可链接环签名算法（Modular-LRS），在随机预言机模型下，具备不可伪造性、匿名性、可链接性、不可诽谤性以及可审计性，并且除匿名性外，其他安全性质能够抵抗恶意监管者的攻击。安全性基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.4.4.3 性能

模块化可链接环签名算法（Modular-LRS）的签名生成、验证时间以及证明尺寸，

关于公钥集元素数量n都是呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。针对固定生成元 Modular-LRS 算法，我们采用 AOS’算法作为普通环签名模块，针对非一致生成元 Modular-LRS 算法，我们采用 AOS 算法作为普通环签名算法模块。

表 7　Modular-LRS&Ring-CT3.0 可链接环签名算法

算法	公钥集元素数	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
Modular-LRS 固定生成元	n	$n+3$	$n+3$	$(1,n+1)$	是
Modular-LRS 非一致生成元	n	$3n+1$	$3n+1$	$(1,n+1)$	否
Ring-CT3.0 可链接环签名	n	$>5n+4$	$>3n+2\log n+5$	$(2\log n+9,7)$	是

5.4.5 可链接环签名算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出可链接环签名算法的应用推广要求。

a) 功能要求
   1) 可链接环签名算法应支持对任意公钥集合规模，都能给出关于该公钥集合的可链接环签名；
   2) 可链接环签名算法应兼容国密椭圆曲线，可复用相关的随机数生成算法、倍点和点乘运算，以及相应的安全参数；
   3) 在区块链系统中，可链接环签名算法应具备双重签名可链接、双重支付可识别的功能特性。

b) 安全性要求
   1) 可链接环签名算法应满足基于标准模型或随机预言机模型下的不可伪造性、匿名性、可链接性和不可诽谤性；
   2) 可链接环签名算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c) 性能要求
   1) 可链接环签名算法宜满足签名尺寸关于公钥集合元素数量呈常数、对数或线性渐进关系，如果呈线性关系，则需满足在常见应用参数下尺寸较短；
   2) 可链接环签名算法应满足签名生成和验证时间关于公钥环元素数量呈线性或对数渐进关系。

根据上述可链接环签名算法应用推广要求，课题组建议将 Modular-LRS、Ring-CT 3.0 可链接环签名算法（国密曲线版本）作为可链接环签名应用推广算法。

5.5 可审计环签名算法研究

5.5.1 相关密码学概念和安全性定义

在可链接环签名算法定义的基础上，可审计环签名需要具备无条件的可审计性。在区块链系统中，可审计环签名能够允许监管者利用监管私钥无条件追踪环签名的签名公钥信息，实现穿透式监管审计。下面给出可审计性的定义：

定义 7（可审计性）

可审计环签名的可审计性定义为由模拟器$\mathcal{S}$和攻击者$\mathcal{A}$之间进行的下面一系列游戏，模拟器$\mathcal{S}$运行初始化并向攻击者$\mathcal{A}$提供初始化公共参数，$\mathcal{A}$被授权访问预言机$\mathcal{RO}$，$\mathcal{JO}$和$\mathcal{CO}$。攻击者$\mathcal{A}$提供测试公钥集合$L_{PK}=\{PK_1,\cdots,PK_n\}$。$\mathcal{A}$获胜当且仅当他能够使用公钥$PK_\kappa \in L_{PK}$生成新的签名$(\sigma,L_{PK},\mu)$，并且满足以下条件（其中$y$是监管私钥）：

a) $\mathrm{Verify}(\sigma,L_{PK},\mu)=1$；

b) $PK_i \neq PK_j$对$1 \le i < j \le n$成立；

c) $\mathrm{Audit}(\sigma,y)\neq\kappa$或者$\mathrm{Audit}(\sigma,y)=\perp$。

我们给出攻击者$\mathcal{A}$在逃离审计攻击中的优势：

$$\mathrm{Adv}_{\mathcal{A}}^{audit}=\Pr[\mathcal{A}\text{获胜}].$$

对于任意 PPT 攻击者$\mathcal{A}$，可链接环签名满足可审计性当且仅当$\mathrm{Adv}_{\mathcal{A}}^{audit}=negl(\lambda)$。

5.5.2 ARS 可审计环签名算法及其优缺点

5.5.2.1 算法概述

ARS 可审计环签名算法的构造方法与 Modular-LRS 算法相似，使用模块化的设计思路，底层普通环签名算法模块灵活可替换，可选取成熟的 AOS 或 AOS’环签名算法模块，也可以选取符合条件的对数尺寸高效环签名算法作为普通环签名算法模块。

通过添加监管者角色，根据监管者的监管公钥生成相应的追踪密钥并且随机化嵌入到环签名公钥集中，使得监管者能够追踪审计到任意环签名的实际签名公钥位置， 在区块链相关金融场景中能够实现交易追踪和审计，在保护隐私的基础上实现丰富的监管审计功能，并且实现抵抗恶意监管者的安全性。

a) Gen：

1) 系统选定椭圆曲线群$(\mathbb{G},q)$，随机选择生成元$g,h_2\in\mathbb{G}$，满足h_2关于g的离散对数未知；

2) 监管者随机选择$y\in\mathbb{Z}_q^*$作为审计私钥，计算审计公钥$h_1=g^y$，保留y作为审计私钥，系统公开公共参数$(\mathbb{G},q,g,h,h_1,h_2)$；

3) 每一个用户P随机选取$x\in\mathbb{Z}_q^*$，计算公钥$PK=g^x$，私钥为x。

b) Sign：

1) 用户P_i随机选取$n-1$个其它用户，将包含自己在内的n个公钥，组成可追踪可链接环签名公钥集合$L_{PK}=\{PK_1,\cdots,PK_n\}=\{g^{x_1},\cdots,g^{x_n}\}$，此时假定用

户P_i的公私钥为$(g^{x_i}, x_i) = (g^x, x)$；

2) 用户P_i计算$TK = h_1^{x_i}, I = h_2^{x_i}$，计算随机数$e_1 = H(L_{PK}, I, TK, 1), e_2 = H(L_{PK}, I, TK, 2)$；

3) 用户P_i计算普通环签名公钥集

$$L_{RPK} = \{PK_1 \cdot TK^{e_1} \cdot I^{e_2}, \cdots, PK_n \cdot TK^{e_1} \cdot I^{e_2}\}$$
$$= \{g^{x_1} h_1^{e_1 x_i} h_2^{e_2 x_i}, \cdots, g^{x_n} h_1^{e_1 x_i} h_2^{e_2 x_i}\};$$

4) 对于消息m，用户P_i计算普通环签名$\tau = RSIG(x_i, L_{RPK}, m)$（普通环签名以$gh_1^{e_1} h_2^{e_2}$为生成元）；

5) 投票人输出最终的可审计环签名为$\sigma = (L_{PK}, m, \tau, I, TK)$。

c) Verify：

1) 验证者计算$e_1 = H(L_{PK}, I, TK, 1), e_2 = H(L_{PK}, I, TK, 2)$；

2) 验证者计算$L_{RPK} = \{PK_1 \cdot TK^{e_1} \cdot I^{e_2}, \cdots, PK_n \cdot TK^{e_1} \cdot I^{e_2}\}$

$$= \{g^{x_1} h_1^{e_1 x_i} h_2^{e_2 x_i}, \cdots, g^{x_n} h_1^{e_1 x_i} h_2^{e_2 x_i}\};$$

3) 验证者检验普通环签名$\tau = RSIG(x_i, L_{RPK}, m)$的正确性，以$gh_1^{e_1} h_2^{e_2}$为生成元；

4) 验证者通过可审计环签名$\sigma = (L_{PK}, m, \tau, I, TK)$的正确性（合法性）验证。

d) Link：

1) 对于可审计环签名$\sigma = (L_{PK}, m, \tau, I, TK)$以及$\sigma' = (L'_{PK}, m', \tau', I', TK')$，如果$I = I'$则输出 Linked，反之输出 unlinked。

e) Audit：该步骤由监管者执行

1) 对于所有$L_{PK} = (PK_1, \cdots, PK_n) = (g^{x_1}, \cdots, g^{x_n})$，监管者计算并搜索第一个满足$PK_i = TK^{y^{-1}}$条件的$i$，输出追踪结果$i$。

5.5.2.2 安全性

可审计环签名算法（ARS），在随机预言机模型下，具备不可伪造性、匿名性、可链接性、不可诽谤性和可审计性，包括监管者在内的任何多项式时间攻击者无法伪造签名、进行双重签名、诽谤用户、逃离监管，实现了抵抗恶意监管者的安全性。基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.5.2.3 性能

可审计环签名算法（ARS）的签名生成、验证和审计所需时间以及证明尺寸，关于公钥集元素数量n都是呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。在下表中我们采用 AOS’算法作为普通环签名模

块。

表 8　ARS 可审计环签名算法

算法	公钥集元素数	生成	验证	尺寸 $(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
ARS	n	$n+5$	$n+5$	$(2,n+1)$	是

5.5.3 可审计环签名算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出可审计环签名算法的应用推广要求。

a) 功能要求
   1) 可审计环签名算法应支持对任意公钥集合规模，都能给出关于该公钥集合的可审计环签名；
   2) 可审计环签名算法应兼容国密椭圆曲线，可复用相关的随机数生成算法、倍点和点乘运算，以及相应的安全参数；
   3) 可审计环签名算法应满足监管者利用监管私钥，可以无条件获取签名人的公钥信息。

b) 安全性要求
   1) 可审计环签名算法应满足基于标准模型或随机预言机模型下的不可伪造性、匿名性、可链接性、不可诽谤性和可审计性，并能够抵抗恶意监管者的攻击；
   2) 可审计环签名算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c) 性能要求
   1) 可审计环签名算法宜满足签名尺寸关于公钥集合元素数量呈对数或线性渐进关系，如果呈线性关系，则需满足在常见应用参数下尺寸较短；
   2) 可审计环签名算法应满足签名生成和验证时间关于公钥环元素数量呈线性或对数渐进关系。

根据上述可审计环签名算法算法应用推广要求，课题组建议将 ARS 可审计环签名算法（国密曲线版本）作为可审计环签名应用推广算法。

5.6 多环签名算法研究

5.6.1 MLSAG 可链接多环签名算法及其优缺点

2015 年，Neother 等[8]给出了适配门罗币框架的可链接多环签名算法，在假定环数量为m时，MLSAG 算法的可链接性可对于全部环有效（即对于每个子环内部都支持可链接性，在区块链多输入交易中防止任意一个输入的双重支付），同时具备线性签名尺寸和签名与验证时间。

5.6.2 CLSAG 可链接多环签名算法及其优缺点

2019 年，Goodell 等[29]给出了 CLSAG 多环签名算法，是 MLSAG 算法的改进方案，具备更短的签名尺寸和运行时间。但是，在假定环数量为m时， CLSAG 算法仅对于第一个子环具备可链接性，在区块链交易中只能支持单输入交易的防止双重支付功能。

5.6.3 模块化可链接多环签名算法（modular-MLRS）及其优缺点

5.6.3.1 算法概述

针对当前 MLSAG 和 CLSAG 算法的不足之处，使用模块化的算法设计思路，按照与 modular-LRS 相似的签名标签生成与嵌入方法，可以构造新的多环签名算法 modular-MLRS，支持多环扩展以及更高的签名与验证效率，并且实现与 CLSAG 同样的签名尺寸。

对于支持全部子环可链接的 modular-MLRS 算法版本构造，与 modular-LRS 完全相同，支持固定生成元签名标签和随机数生成元签名标签。对于每个子环L_{PK}，生成签名标签$tag = I = g_2^{x_i}$或$H_p(PK_i)^{x_i}$，然后计算$e = H(L_{PK}, I)$，并完成随机化嵌入得到普通环签名公钥集：

$$L_{RPK} = \{PK_1 \cdot I^e, \cdots, PK_n \cdot I^e\}$$

$$= \{g_1^{x_1} g_2^{ex_i}, \cdots, g_1^{x_n} g_2^{ex_i}\}或\{g_1^{x_1} H_p(PK_i)^{ex_i}, \cdots, g_1^{x_n} H_p(PK_i)^{ex_i}\}.$$

进而调用普通多环签名完成可链接多环签名的过程，支持全部子环的可链接性，支持多输入交易的抵抗双重支付攻击安全性，算法详情这里不再赘述。

针对 CLSAG 算法改进，当只需满足在非一致生成元签名标签下针对第一个具备环可链接性的情形，可以基于模块化的构造思路，采用多环压缩和随机化嵌入相结合的构造方法实现高效的可链接多环签名构造方法，算法构造如下（由于篇幅原因这里介绍双环的版本，多环版本可直接推广得到）：

a) Setup：系统选定素数阶椭圆曲线群$(\mathbb{G}, q)$，随机选择生成元$g \in \mathbb{G}$，系统公开公共参数$(\mathbb{G}, q, g)$。

b) Gen：

1) 每一个用户P随机选取$x \in \mathbb{Z}^*$，计算第一公钥$PK = g^x$，第一私钥为$SK = x$；

2) 每一个用户P随机选取$x' \in \mathbb{Z}^*$，计算第二公钥$PK = g^{x'}$，第二私钥为$SK = x'$。

c) Sign：

1) 用户P_i（P_i是本方案的签名人）随机选取$n-1$个其它用户，将包含自己在内的n个公钥，组成第一公钥集合$L_{PK} = \{PK_1, \cdots, PK_n\} = \{g^{x_1}, \cdots, g^{x_n}\}$，此时假定用户$P_i$的公私钥为$(g^{x_i}, x_i)$；

2) 对于$j = 1, \cdots, n$，用户P_i计算$h_j = H_p(PK_j)$，随后用户计算签名标签$I_1 = H_p(PK_i)^{x_i} = h_i^{x_i}$；

3) 用户根据其它用户的第二公钥信息，计算第二公钥集合$L'_{PK} = \{PK'_1, \cdots, PK'_n\}$，其中用户$P_i$知道$PK'_i = g^{x'_i}$中的私钥$x'_i$，然后用户$P_i$计算$I_2 = H_p(PK_i)^{x'_i}$；（满足第一和第二公钥集合中全部用户公钥位置相同）

4) 用户P_i然后计算$e_1 = H(L_{PK}, I_1, I_2, 1), e_2 = H(L_{PK}, I_1, I_2, 2)$；

5） 用户计算压缩环签名公钥集

$$L_{RPK}=\left\{PK_1\cdot I_1^{e_1}\cdot {PK_1'}^{e_2}\cdot I_2^{e_1e_2},\cdots,PK_n\cdot I_1^{e_1}\cdot {PK_n'}^{e_2}\cdot I_2^{e_1e_2}\right\}$$

$$=\left\{g^{x_1+e_2x_1'}h_i^{e_1x_i+e_1e_2x_i'},\cdots,g^{x_n+e_2x_n'}h_i^{e_1x_i+e_1e_2x_i'}\right\}=\{RPK_1,\cdots,RPK_n\};$$

6） 对于消息μ，用户P_i对L_{RPK}执行签名操作，分为以下几个步骤：

7） 用户P_i随机选取$r_i\in\mathbb{Z}_q^*$，计算$c_{i+1}=H((gh_i^{e_1})^{r_i},L_{RPK},\mu)$；

8） 对于$j=i+1,\cdots,n,1,\cdots,i-1$，用户$P_i$随机选取$z_j\in\mathbb{Z}_q^*$，并计算

$$c_{j+1}=H((gh_j^{e_1})^{z_j}/\left(RPK_j\right)^{c_j},L_{RPK},\mu);$$

9） 用户P_i计算$z_i=r_i+x_ic_i$；

10）用户P_i得到$\tau=(c_1,z_1,\cdots,z_n)$；

11）用户输出最终的可链接多环签名为$\sigma=(L_{PK},\mu,\tau,I_1,I_2)$。

d） Link：

1） 对于可链接多环签名$\sigma=(L_{PK},\mu,\tau,I_1,I_2)$以及$\sigma'=(L'_{PK},\mu',\tau',I'_1,I'_2)$，如果$I_1=I_1'$则输出 Linked，反之输出 unlinked。

e） Verify：

1） 验证者计算$e_1=H(L_{PK},I_1,I_2,1),e_2=H(L_{PK},I_1,I_2,2)$，然后计算

2） $L_{RPK}=\left\{PK_1\cdot I_1^{e_1}\cdot {PK_1'}^{e_2}\cdot I_2^{e_1e_2},\cdots,PK_n\cdot I_1^{e_1}\cdot {PK_n'}^{e_2}\cdot I_2^{e_1e_2}\right\}$;

3） 对于$j=1,\cdots,n$，验证者计算$h_j=H_p(PK_j)$；

4） 对于$j=1,\cdots,n-1$，验证者计算$c_{j+1}=H((gh_j^{e_1})^{z_j}/\left(RPK_j\right)^{c_j},L_{RPK},\mu)$；

5） 验证者计算$c_1=H((gh_n^{e_1})^{z_n}/(RPK_n)^{c_n},L_{RPK},\mu)$是否成立，全部验证通过后，验证者通过可链接多环签名$\sigma=(L_{PK},\mu,\tau,I_1,I_2)$的正确性（合法性）验证。

5.6.3.2 安全性

模块化可链接多环签名算法（modular-MLRS），在随机预言机模型下，具备不可伪造性、匿名性、可链接性和不可诽谤性。基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.6.3.3 性能

模块化可链接多环签名算法（modular-MLRS）的签名生成、验证时间以及证明尺寸，关于公钥集元素数量n都是呈线性关系，关于环数量m同样呈线性关系。对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。在下表中固定生成元签名标签的 modular-MLRS 算法我们采用 AOS’算法（多环版本）作为普通多

环签名模块，非一致生成元签名标签的 modular-MLRS 算法（支持第一个环可链接版本）我们使用 AOS 算法（多环版本）作为普通多环签名模块。

表 9　modular-MLRS&MLSAG&CLSAG 可链接多环签名算法

算法	公钥集元素数 &环数量	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
modular-MLRS 固定生成元	n,m	$mn+3$	$mn+m+2$	$(1,m+n)$	是
modular-MLRS 非一致生成元（首环可链接）	n,m	$mn+2m+2n-1$	$mn+m+2n$	$(m,n+1)$	否
MLSAG[8]	n,m	$4mn-m$	$4mn$	$(1,mn+1)$	否
CLSAG[29]	n,m	$mn+m+4n-2$	$mn+m+4n$	$(m,n+1)$	否

5.6.4 可链接多环签名算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出可链接多环签名算法的应用推广要求。

a) 功能要求

1) 可链接多环签名算法应支持对任意数量的公钥集合元素以及子环数量，都能给出可链接多环签名；
2) 可链接多环签名算法应兼容国密椭圆曲线，可复用相关的随机数生成算法、倍点和点乘运算，以及相应的安全参数；
3) 可链接多环签名算法应支持全部子环或者任意子环的可链接性，支持区块链多输入交易或单输入交易的防止双重支付安全性。

b) 安全性要求

1) 可链接多环签名算法应满足基于标准模型或随机预言机模型下的不可伪造性、匿名性、可链接性、不可诽谤性；
2) 可链接多环签名算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c) 性能要求

1) 可链接多环签名算法应满足签名尺寸关于公钥环元素数量呈对数或线性渐进关系；
2) 可链接多环签名算法应满足证明和验证时间关于公钥环元素数量呈线性或对数渐进关系；
3) 可链接多环签名算法应满足签名尺寸关于环数量呈线性或对数渐进关系；
4) 可链接多环签名算法应满足签名和验证时间关于环数量呈线性或对数渐进关系。

根据上述可链接多环签名算法应用推广要求，课题组建议将 modular-MLRS 算法（国密曲线版本）作为可链接多环签名应用推广算法。

5.7 多重监管审计算法研究

5.7.1 多重监管算法设计方法研究

在支持单监管者的可审计范围证明和可审计环签名基础上，需要探索支持多监管者多重审计的算法设计方法，以支持区块链跨境支付结算和供应链金融等多监管者共同监管审计的应用场景，经过课题组场景调研和需求分析讨论，区块链多重监管隐私保护算法需要满足以下几方面：

a) 需要设计研发支持多重监管的可审计范围证明算法和可审计环签名算法，丰富现有技术的监管功能，更好适配隐私保护区块链的多重监管场景；
b) 各个监管者无需链下通信或联合计算即可分别实现穿透式监管审计，确保全部监管者的监管权益；
c) 对于m个监管者而言，多重监管可审计范围算法和可审计环签名算法的生成与验证时间，证明和签名尺寸要低于分别执行m次单监管者算法的运算量和尺寸；
d) 多重监管可审计范围算法和可审计环签名算法需要满足抵抗恶意监管者的安全性，即使全部监管者合谋作恶，也无法打破相关算法的完备性、合理性、匿名性、不可伪造性、可审计性等安全性要求。

基于上述算法设计要求，结合模块化的算法设计思想，我们使用多监管者追踪密钥随机化嵌入的方式，实现了多重监管可审计范围证明和多重监管可审计环签名的构造，具备较高的性能和抵抗恶意监管者合谋的安全性。

5.7.2 MARP 多重监管可审计范围证明算法

下面我们给出多重监管可审计 Borromean 范围证明算法 MARP（Multi-Auditable Range proof）的描述。

5.7.2.1 算法概述

a) Setup：
   1) 系统选定椭圆曲线群$(\mathbb{G},q)$，随机选择生成元$g \in \mathbb{G}$；
   2) 假设系统中有m个监管者，各自都能监管链上的隐私数据，我们记每个监管者为$R_i, i = 1,\cdots,m$，每个监管者R_i随机选择$y_i \in \mathbb{Z}_q^*$作为监管者R_i的监管陷门，计算$h_i = g^{y_i}$，保留y_i作为陷门；
   3) 系统再随机选取$h \in \mathbb{G}$，使得任何人无法获得g与h的离散对数关系，可选的使用随机哈希生成h，即使用 Hash-to-Point 计算$h = H_p(g, h_1, gh_m)$，系统公开公共参数$(\mathbb{G},q,g,h,h_1,\cdots,h_m)$。 其中$H_p$是 Hash to point，用于随机生成椭圆曲线点。
b) Prove：
   1) 证明者计算金额承诺$c = g^x h^a$，其中$a \in [0,2^n - 1]$，随机选取$x_0,\cdots,x_{n-1} \in \mathbb{Z}_q^*$，计算$\beta = x - x_0 - \cdots - x_{n-1}$，同时将$a$比特展开为$a = a_0 + \cdots + 2^i a_i + \cdots + 2^{n-1} a_{n-1}$，$a_i = 0,1$；

2) 对于每个$i=0,\cdots,n-1$，证明者计算$c_i=g^{x_i}h^{2^i a_i}$，$c_i'=g^{x_i}h^{2^i a_i-2^i}$，记$L_i=(c_i,c_i')$为子承诺集，$\mathrm{L}=\{L_0,\cdots,L_{n-1}\}$为承诺集；

3) 证明者对于每个$i=0,\cdots,n-1$，$j=1,\cdots,m$，证明者计算追踪密钥$TK_{i,j}=h_j^{x_i}$和子数值标签$I_i=h^{x_i}$，记追踪密钥集为$\mathrm{TK}=\{TK_{i,j}\}_{i=0,\cdots,n-1,j=1,\cdots,m}$，数值标签集为$\mathrm{I}=\{I_0,\cdots,I_{n-1}\}$然后，证明者计算$k$个随机数

$$e_k=H(\mathrm{L};\mathrm{TK};\mathrm{I};k),\ k=1,\cdots,m+1;$$

4) 对于每个$i=0,\cdots,n-1$，证明者计算子公钥组为$L_{PK_i}=(c_i\cdot I_i^{e_{m+1}}\cdot\prod_{j=1}^{m}TK_{i,j}^{e_j},c_i'\cdot I_i^{e_{m+1}}\cdot\prod_{j=1}^{m}TK_{i,j}^{e_j})$私钥为$x_i$，相应的生成元为$g_1=gh^{e_{m+1}}\prod_{j=1}^{m}h_j^{e_j}$；

5) 证明者计算公钥组$L_{PK}=\{L_{PK_1},\cdots,L_{PK_{n-1}}\}$；

6) 证明者运行多环签名$\sigma=RSIG(L_{PK},x_0,\cdots,x_{n-1},c,\beta,\mathrm{TK},\mathrm{I})$，使用生成元为$g_1=gh^{e_{m+1}}\prod_{j=1}^{m}h_j^{e_j}$；

7) 证明者输出可追踪区间证明结果$(c,\beta,L,\mathrm{TK},\mathrm{I},\sigma)$。

c) Verify：

1) 验证$\frac{c_i}{c_i'}=h_2^{2^i}$是否成立；

2) 验证$g^{\beta}\cdot\prod c_i=c$是否正确；

3) 验证者计算$e_k=H(\mathrm{L};\mathrm{TK};\mathrm{I};k)$，$k=1,\cdots,m+1$；

4) 对于所有$i=0,\cdots,n-1$，验证者计算$L_{PK_i}=(c_i\cdot I_i^{e_{m+1}}\cdot\prod_{j=1}^{m}TK_{i,j}^{e_j},c_i'\cdot I_i^{e_{m+1}}\cdot\prod_{j=1}^{m}TK_{i,j}^{e_j})$，得到$L_{PK}=\{L_{PK_1},\cdots,L_{PK_{n-1}}\}$；

5) 验证Borromean多环签名σ的正确性，使用生成元为$g_1=gh^{e_{m+1}}\prod_{j=1}^{m}h_j^{e_j}$。

d) Audit：对于$i=1,\cdots,m$，监管者R_j进行如下运算：

1) 对于每一位$i=0,\cdots,n-1$，监管者R_j计算$TK_{i,j}^{y_j^{-1}}$；

2) 如果$c_i=TK_{i,j}^{y_j^{-1}}$，则R_j输出$a_i=0$；

3) 如果$c_i'=TK_{i,j}^{y_j^{-1}}$，则R_j输出$a_i=1$；

4) 监管者R_j计算$a=a_0+\cdots+2^i a_i+\cdots+2^{n-1}a_{n-1}$。

5.7.2.2 安全性

多重监管可审计范围证明（MARP），在随机预言机模型下，具备完备性、特殊合理性、零知识性以及可审计性，其中零知识性仅对不掌握审计私钥的多项式时间攻击者成立，而完备性、特殊合理性和可审计性对于掌握审计私钥的攻击者仍保持成立，并且能够抵抗全部监管者的合谋攻击。算法的安全性基于椭圆曲线离散对数和（扩展）DDH 等困难问题，以及哈希函数的伪随机性，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.7.2.3 性能

多重监管可审计范围证明（MARP）的证明生成、验证时间以及证明尺寸，关于隐私数据长度n都是呈线性关系，关于监管者数量m同样呈线性关系。

5.7.3 MARS 多重监管可审计环签名算法

5.7.3.1 算法描述

多重监管可审计环签名 MARS（Multi-Auditable Ring signature）描述如下。

a) Gen：

1) 系统选定椭圆曲线群$(\mathbb{G},q)$，随机选择生成元$g \in \mathbb{G}$；

2) 假设系统中有m个监管者，各自都能监管链上的隐私数据，我们记每个监管者为$R_i, i = 1,\cdots,m$，每个监管者R_i随机选择$y_i \in \mathbb{Z}_q^*$作为监管者R_i的监管陷门，计算$h_i = g^{y_i}$，保留y_i作为陷门；

3) 系统再随机选取$h \in \mathbb{G}$，使得任何人无法获得g与h的离散对数关系，可选的使用随机哈希生成h，即使用 Hash-to-Point 计算$h = H_p(g, h_1, gh_m)$，系统公开公共参数$(\mathbb{G},q,g,h,h_1,\cdots,h_m)$。 其中$H_p$是 Hash to point，用于随机生成椭圆曲线点。

4) 每一个用户P随机选取$x \in \mathbb{Z}_q^*$，计算公钥$PK = g^x$，私钥为x。

b) Sign：

1) 用户P_i 随机选取$n-1$ 个其它用户，将包含自己在内的n 个公钥，组成多重监管可追踪可链接环签名公钥集合$L_{PK} = \{PK_1,\cdots,PK_n\} = \{g^{x_1},\cdots,g^{x_n}\}$，此时假定用户$P_i$的公私钥为$(g^{x_i}, x_i) = (g^x, x)$；

2) 用户P_i 分别计算追踪密钥$TK_j = h_j^{x_i}, j = 1,\cdots,m$和签名标签$I = h^{x_i}$，记追踪密钥集为 $\mathrm{TK} = \{TK_j\}_{j=1,\cdots,m}$，然后计算 $m+1$ 个随机数 $e_j = H(L_{PK}, I, \mathrm{TK}, j), j = 0,\cdots,m$；

3) 用户P_i 计算普通环签名公钥集

$$L_{RPK}=\left\{PK_1\cdot I^{e_0}\cdot\prod_{j=1}^{m}TK_j^{e_j},\cdots,PK_n\cdot I^{e_0}\cdot\prod_{j=1}^{m}TK_j^{e_j}\right\}$$

$$=\left\{g^{x_1}\cdot h^{e_0x_i}\cdot\prod_{j=1}^{m}h_j^{e_jx_i},\cdots,g^{x_n}\cdot h^{e_0x_i}\cdot\prod_{j=1}^{m}h_j^{e_jx_i}\right\};$$

4) 对于消息μ，用户P_i计算普通环签名$\tau=RSIG(x_i,L_{RPK},\mu)$（普通环签名以$gh^{e_0}\prod_{j=1}^{m}h_j^{e_j}$为生成元）；

5) 签名人输出最终的多重监管可审计环签名为$\sigma=(L_{PK},\mu,\tau,I,\mathrm{TK})$。

c) Verify：

1) 验证者计算$e_j=H(L_{PK},I,\mathrm{TK},j),j=0,\cdots,m$，然后计算

2) $L_{RPK}=\left\{PK_1\cdot I^{e_0}\cdot\prod_{j=1}^{m}TK_j^{e_j},\cdots,PK_n\cdot I^{e_0}\cdot\prod_{j=1}^{m}TK_j^{e_j}\right\}$；

3) 验证者检验普通环签名$\tau=RSIG(x_i,L_{RPK},\mu)$的正确性，以$gh^{e_0}\prod_{j=1}^{m}h_j^{e_j}$为生成元；

4) 验证者通过多重监管可审计环签名$\sigma=(L_{PK},\mu,\tau,I,\mathrm{TK})$的正确性（合法性）验证。

d) Link：

1) 对于多重监管可审计环签名$\sigma=(L_{PK},\mu,\tau,I,\mathrm{TK})$以及$\sigma'=(L'_{PK},\mu',\tau',I',\mathrm{TK}')$，如果$I=I'$则输出 Linked，反之输出 unlinked。

e) Audit：该步骤由各个监管者执行，无需监管者之间额外通信或联合计算

1) 对于$j=1,\cdots,m$，每一位监管者R_j计算$TK_j^{y_j^{-1}}$，对于所有$L_{PK}=(PK_1,\cdots,PK_n)=(g^{x_1},\cdots,g^{x_n})$，监管者$R_j$计算并搜索第一个满足$PK_i=TK_j^{y_j^{-1}}$条件的$i$，输出追踪结果$i$。

5.7.3.2 安全性

多重监管可审计环签名（MARS），在随机预言机模型下，具备不可伪造性、匿名性、可链接性、不可诽谤性以及可审计性，除匿名性外的其他安全性质对于掌握审计私钥的攻击者仍保持成立，并且能够抵抗全部监管者的合谋攻击。基于椭圆曲线离散对数和 DDH 等困难问题，不依赖配对（pairing）或可信初始化（trusted setup）等额外假设，并且能够适配当前的国密曲线。

5.7.3.3 性能

多重监管可审计环签名（MARS）的签名生成、验证时间以及签名尺寸，关于公钥集元素数量n都是呈线性关系，关于监管者数量m同样呈线性关系，并且满足监管者数量上升对签名生成和验证时间的影响极小。

对于生成与验证时间，我们重点考察其包含的椭圆曲线点乘运算（倍点运算）数量；

对于证明尺寸，我们分别考察其包含的椭圆曲线群$\mathbb{G}$元素数量以及模q整数环$\mathbb{Z}_q^*$的元素数量。在下表中我们采用 AOS’算法（多环版本）作为普通多环签名模块。

表 10　MARP & MARS 多重监管可审计算法

算法(m个监管者)	金额长度 or 环元素数量	生成	验证	尺寸$(\mathbb{G},\mathbb{Z}_q^*)$	支持并行
MARP	n	$2mn+m+6n+1$	$mn+m+5n+2$	$(mn+2n, 2n+2)$	是
MARS	n	$2m+n+3$	$2m+n+3$	$(m+1, n+1)$	否

5.7.4 多重监管算法应用推广要求

根据算法调研与课题组充分讨论，按照下面三方面要求给出多重监管可审计算法的应用推广要求。

a) 功能要求
   1) 多重监管可审计范围证明算法应支持对任意长度的隐私数据，都能给出对于任意区间范围的零知识证明；
   2) 多重监管可审计环签名算法应支持对于任意数量的公钥环，都能给出可审计范围证明，支持多监管者；
   3) 多重监管可审计范围证明算法和多重监管可审计环签名算法应兼容国密椭圆曲线，可复用相关的随机数生成算法、倍点和点乘运算，以及相应的安全参数；
   4) 多重监管可审计范围证明算法和多重监管可审计环签名算法不应依赖各监管方实时通信和计算，各监管方可以独立进行监管审计，并能够抵抗全部恶意监管者的合谋攻击。

b) 安全性要求
   1) 多重监管可审计范围证明算法应满足基于标准模型或随机预言机模型下的完备性、合理性、零知识性和可审计性等安全性要求；
   2) 多重监管可审计环签名算法应满足基于标准模型或随机预言机模型下的匿名性、不可伪造性、可链接性、不可诽谤性和可审计性等安全性要求；
   3) 多重监管可审计范围证明算法和多重监管可审计环签名算法应抵抗恶意监管者合谋攻击；
   4) 多重监管可审计算法应基于主流的安全性假设和数学困难问题，不宜基于可信初始化等额外假设。

c) 性能要求
   1) 多重监管可审计范围证明算法和多重监管可审计环签名算法应满足证明（签名）尺寸关于隐私数据长度（公钥环元素数量）呈线性或对数渐进关系；
   2) 多重监管可审计范围证明算法和多重监管可审计环签名算法应满足证明（签名）生成和验证时间关于隐私数据长度（公钥环元素数量）呈线性或对数渐进关系。

根据上述多重监管算法应用推广要求，课题组建议将本文的多重监管可审计范围证明算法 MARP 和 MARS（国密曲线版本）分别作为多重监管可审计范围证明和多重监管可审计环签名应用推广算法。

6 总结与预期结果

本研究报告分析总结了现有区块链隐私保护机制中的范围证明算法和环签名算法，提出了可审计范围证明、可审计环签名、多重监管可审计密码算法等新概念，能够在区块链实现隐私保护的同时兼容穿透式监管审计，在链上资产交易、跨境支付结算、供应链金融等场景中具备落地的前景。

本研究报告提出了新的模块化密码算法构造方法，给出了新的模块化可审计范围证明、可链接环签名、可审计环签名、可审计多环签名、多重监管可审计范围证明和环签名算法构造方法，使用标签和追踪密钥生成并随机化嵌入的技术，通过调用普通范围证明、普通环签名等算法组件即可实现可审计范围证明、可链接环签名、可审计环签名等技术功能，并且能够抵抗恶意监管者的潜在攻击，实现了更高的效率。

本研究报告算法的模块化设计方法和可审计功能添加，提供了自主设计的算法与实现策略,能够有效适配各类金融应用场景，尤其满足相关节点需要签名隐私保护的需求,通过零知识性、可链接性、可审计性等安全性特征，更好地服务于金融隐私保护应用。

参考文献

[1] Nakamoto S. Bitcoin: A peer-to-peer electionic cash system[J]. Unpublished, 2008.

[2] Buterin V. A next-generation smart contract and decentralized application platform (2014), https://cryptorating.eu/whitepapers/Ethereum/Ethereumwhitepaper.pdf.

[3] Sasson E B, Chiesa A, Garman C, Green M, Miers I, Tromer E, and Virza M. Zerocash: Decentralized anonymous payments from Bitcoin[C]. In IEEE Symposium on Security and Privacy. IEEE, 2014. IACR Cryptology ePrint Archive 2014:349

[4] Saberhagen N V: Cryptonote v 2.0 (2013), https://cryptonote.org/whitepaper.pdf.

[5] DERO Community. Dero White Paper[J]. URl: https://dero.io/attachment/Whitepaper.pdf, 2019.

[6] Bünz B, Agrawal S, Zamani M, et al. Zether: Towards Privacy in a Smart Contract World[J]. IACR Cryptology ePrint Archive, 2019: 191.

[7] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing[C]. Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1991: 129-140.

[8] Noether S, Mackenzie A, et al. Ring confidential transactions[J]. Ledger, 1:1–18, 2016.

[9] Li Y, Yang G, Susilo W, et al. Traceable Monero: Anonymous Cryptocurrency with Enhanced Accountability[J]. IEEE Transactions on Dependable and Secure Computing, 2019.

[10] Facebook. Libra White Paper[J]. URl: https://libra.org/en-US/white-paper/, 2019.

[11] Bünz B, Bootle J, Boneh D, Poelstra A, Wuille P, and Maxwell G. Bulletproofs: Efficient range proofs for confidential transactions[C]. In IEEE S&P, May 2018.

[12] Wahby R S, Tzialla I, Shelat A, Thaler J, Walfish M. Doubly-efficient zk-snarks without trusted setup. In: 2018 IEEE Symposium on Security and Privacy (SP), 2018: 926–943.

[13] Xie T, Zhang J, Zhang Y, Papamanthou C, Song D: Libra: Succinct zero-knowledge proofs with optimal prover computation. IACR Cryptology ePrint Archive 2019:317.

[14] Rivest R L, Shamir A, Tauman Y. How to leak a secret[C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2001: 552-565.

[15] Abe M, Ohkubo M, Suzuki K. 1-out-of-n signatures from a variety of keys[C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2002: 415-432.

[16] Bender A, Katz J, Morselli R. Ring signatures: Stronger definitions, and constructions without random oracles[C]. Theory of Cryptography Conference. Springer, Berlin, Heidelberg, 2006: 60-79.

[17] Dodis Y, Kiayias A, Nicolosi A, et al. Anonymous identification in ad hoc groups[C]. International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2004: 609-626.

[18] Chandran N, Groth J, Sahai A. Ring signatures of sub-linear size without random oracles[C]. International Colloquium on Automata, Languages, and Programming. Springer, Berlin, Heidelberg, 2007: 423-434.

[19] Groth J, Kohlweiss M. One-out-of-many proofs: Or how to leak a secret and spend a coin[C]. Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2015: 253-280.

[20] Yuen T H, Sun S, Liu J K, et al. RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security[J]. 2019.

[21] Liu J K, Wei V K, Wong D S. Linkable spontaneous anonymous group signature for ad hoc groups[C]. Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2004: 325-335.

[22] Tsang P P, Wei V K. Short linkable ring signatures for e-voting, e-cash and attestation[C]. International Conference on Information Security Practice and Experience. Springer, Berlin, Heidelberg, 2005: 48-60.

[23] Au M H, Chow S S M, Susilo W, et al. Short linkable ring signatures revisited[C]. European Public Key Infrastructure Workshop. Springer, Berlin, Heidelberg, 2006: 101-115.

[24] Yuen T H, Liu J K, Au M H, et al. Efficient linkable and/or threshold ring signature without random oracles[J]. The Computer Journal, 2013, 56(4): 407-421.

[25] Liu J K, Au M H, Susilo W, et al. Linkable ring signature with unconditional anonymity[J]. IEEE Transactions on Knowledge and Data Engineering, 2013, 26(1): 157-165.

[26] Back A. Ring signature efficiency[J]. Bitcointalk (accessed 1 May 2015) URl: https://bitcointalk.org/index.php, 2015.

[27] Gregory M and Andrew P. Borromean ring signatures[J]. URl: http://diyhpl.us/~bryan/papers2/bitcoin/ Borromean ring signatures.pdf, 2015.

[28] Sun S F, Au M H, Liu J K, et al. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency Monero[C]. European Symposium on Research in Computer Security. Springer, Cham, 2017: 456-474.

[29] Goodell B, Noether S, Blue A. Compact linkable ring signatures and applications[J]. 2019.

[30] Fujisaki E, Suzuki K. Traceable ring signature[C]. International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2007: 181-200.

[31] Zhang F, Huang N N, Gao S. Privacy data authentication schemes based on Borromean ring signature[J]. Journal of Cryptologic Research, 2018, 5(5): 529 -

537. [DOI: 10.13868/j.cnki.jcr.000262] 张凡, 黄念念, 高胜. 基于 Borromean 环签名的隐私数据认证方案 [J]. 密码学报, 2018, 5(5): 529-537. [DOI:10.13868/j.cnki.jcr.000262].