计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2021.20200751
58(2):281-290,2021

一种基于区块链的泛用型数据隐私保护的安全多方计算协议

刘 峰[1,3] 杨 杰[2] 李志斌[3] 齐佳音[2]
[1](华东师范大学计算机科学与技术学院 上海 200062)
[2](上海对外经贸大学人工智能与变革管理研究院 上海 200336)
[3](华东师范大学数据科学与工程学院 上海 200062)
(lsttoy@163.com)

A Secure Multi-Party Computation Protocol for Universal Data Privacy Protection Based on Blockchain

Liu Feng[1,3], Yang Jie[2], Li Zhibin[3], and Qi Jiayin[2]
[1](*School of Computer Science and Technology, East China Normal University, Shanghai* 200062)
[2](*Institute of Artificial Intelligence and Change Management, Shanghai University of International Business and Economics, Shanghai* 200336)
[3](*School of Data Science and Engineering, East China Normal University, Shanghai* 200062)

Abstract Recent years, how to protect user privacy data on the blockchain reasonably and efficiently is a key issue in the current blockchain technology field. Based on this, in this paper, a secure multi-party computation protocol is designed based on the Pedersen commitment and Schnorr protocol (protocol of blockchain based on Pedersen commitment linked schnorr protocol for multi-party computation, BPLSM). Through constructing the structure of the protocol and carrying out formal proof calculations, it is confirmed that the protocol can be integrated into the blockchain network to merge different private messages for efficient signing under anonymity. In addition, by analyzing the nature and security of the protocol, it can be proved that the overhead about computation of the general-purpose privacy computing scheme using the BPLSM protocol on the blockchain is low, and it also has strong information imperceptibility. In the end, experimental simulation results show that the time cost of BPLSM protocol verification in a small-scale multi-party transaction with a fixed number of people is about 83.5% lower than that of the current mainstream BLS signature.

Key words blockchain; privacy computing; secure multi-party computation; Pedersen commitment; Schnorr signature; BLS signature

摘 要 近年来,如何合理有效地在区块链上实现用户隐私数据保护是区块链技术领域的一个关键性问题.针对此问题,设计出一种基于 Pedersen 承诺与 Schnorr 协议的安全多方计算协议(protocol of blockchain based on Pedersen commitment linked Schnorr protocol for multi-party computation, BPLSM).通过构筑该协议架构并进行形式化证明演算,表明了该协议能够融入区块链网络、能够在匿名情况下合并不同隐私消息并进行高效签署的特点.此外分析了协议的性质与安全性,证明了在区块链

收稿日期:2020-09-16;**修回日期**:2020-10-26
基金项目:国家重点研发计划项目(2017YFB0803304);国家自然科学基金项目(72042004)
This work was supported by the National Key Research and Development Program of China (2017YFB0803304) and the National Natural Science Foundation of China (72042004).
通信作者:李志斌(lizb@cs.ecnu.edu.cn);齐佳音(ai@suibe.edu.cn)

中应用 BPLSM 协议的泛用型隐私计算方案计算上的低算力开销,并具备良好的信息隐蔽性.最后对协议进行实验仿真,结果表明:在小范围人数固定的多方计算中,BPLSM 协议验签的时间成本比当前主流的 BLS 签名节省约 83.5%.

关键词 区块链;隐私计算;安全多方计算;Pedersen 承诺;Schnorr 签名;BLS 签名

中图法分类号 TP301.4

自中本聪 2008 年提出比特币以来,区块链作为一种跨行业应用的突破性底层技术,得到了飞速的发展.与大数据、云计算、人工智能等当前流行的信息技术相比,区块链去中心化、难篡改、可追溯、公开透明等特性更能满足人们日益增长的需求[1].然而,随着学界研究的不断深入,越来越多学者发现,在基于区块链技术的去中心化账本中用户的信息是很容易被追踪的.陈伟利等人[2]就通过研究发现,用户隐私信息的窃取与其用户地址信息的泄露多少是有很大关联的.基于环签名的数字货币门罗币,也曾被 Möser 等人[3]通过追踪用户签名私钥的方式攻破了隐藏交易,从而对交易发起人的隐私信息造成了很大的威胁.除此之外,大多数区块链交易所也都集成了中心化身份验证机制,用户的个人信息和区块链的地址之间映射关系会理所当然地被交易所记录下来.在大数据技术运用下,用户的交易信息和行为被破解的风险也就会激增.2017 年 Ermilov 等人就利用自动聚类法分析出用户与其比特币地址间的关联关系,提出了用户不安全的比特币使用模式[4].

因此,区块链技术的安全问题也越来越受到重视.如何合理高效地做到用户身份信息及交易数据的隐私保护是当前区块链技术领域在安全方面的一个关键问题.如在医疗健康服务行业,如何安全高效地解决个人医疗健康数据共享问题同时保证个人数据的隐私;再如在货运物流行业,如何与多方货物供应商快速建立信任关系,保证优质长尾资源同时确保交易信息不被窃取篡改等.随着技术的不断更新与迭代,安全多方计算的出现为解决此类问题提供了可能途径.

在当前区块链安全多方计算的研究中,区块链科技服务商 Defi 利用区块链以及可信计算搭建了帮助企业能够实现联合风控的系统架构,但效率、透明度上还存在一些问题;宋俊典等人[5]提出了一种基于区块链的数据治理协同方法,并给出多方协作的构建标准以实现区块链协同治理,但是隐私安全上却并没有给出较为具体的分析策略.结合上述文献并从当前技术发展趋势来看,要提升效率的同时做到有效的隐私保护,设计一种匿名条件下对多方消息进行合并签署并高效验签的安全多方计算协议势在必行.其实早在 2001 年就出现了可对不同消息进行合并签署并验证的 BLS 方案[6],并且近些年该方案主要提出者 Boneh 教授仍在更新这种签名方案[7]以适配当前安全多方计算的研究发展.不过由于 BLS 签名自身独有的运算逻辑导致其过于依赖有限域中椭圆曲线上的双线性对(bilinear pairing)运算且运算次数较多[8-9],这导致在相同签名条件下,BLS 签名甚至要比当前以太坊(Ethereum)广泛使用的 ECDSA 签名验签花费的时间成本多上 3 倍.

本文在已有的研究基础上,为了对不同消息进行合并签署并验证,在现有聚合签名基础上提升签名验签的效率,同时加强敏感数据的隐私性和匿名性,参照了 Yu 的 Pedersen 承诺加 Schnorr 签名方案[10],设计出了 BPLSM 协议,一种区块链上融合 Pedersen 承诺与 Schnorr 协议的安全多方计算协议.通过对该协议进行形式化验证,证明了 BPLSM 协议能确保信息在不被泄露情况下证实签名的合法有效.在此之后,利用此协议设计相关实验构建了一个半诚实者模型下的签名验签程序.实验主体逻辑是通过对验签不通过的签名进行舍弃,验签通过的抽离出相关签名信息交由链上合约使用承诺时的盲因子和宣告进行解析,得出统和意见后处理事务并公布结果,保证签署消息的合理性、匿名验签的高效可行性.实验结果证明,在相同验签条件下基于 BPLSM 协议构建的签名比 BLS 签名在验签上能节省约 83.5%的时间成本.

1 基础知识

本节主要介绍协议所涉及的基础知识,包括 Pedersen 承诺、Schnorr 协议以及安全多方计算.

1.1 Pedersen 承诺

目前隐私保护方案中使用较为广泛的密码学承诺之一便是 Pedersen 承诺[11].与常见的 Hash 承诺

相比,Pedersen 承诺具备同态特性,可直接对密文消息进行操作,即无需破坏敏感数据源就能实现隐匿计算.

在区块链中,椭圆曲线加密(Elliptic Curve Cryptography, ECC)是基础技术之一.而基于椭圆曲线的 Pedersen 承诺[12]则是一种构造范围证明的重要手段.对于一次 Pedersen 承诺,示证者 A 向验证者 B 发出关于隐私消息 m 的承诺后,再由 B 验证承诺合法性的过程如图 1 所示.从图 1 中可以看出,该承诺实现方式为:承诺的准备阶段在椭圆曲线上选取 2 个以某个大素数 p 为阶的基点 G,H,作为承诺生成时的公共参数.然后在承诺的产生阶段,由 A 产生一个随机数 $seed$,通过 $seed$ 按照如下公式计算承诺 $Commitment$ 并将其发送给 B:

$$Commitment = m * G + seed * H, \tag{1}$$

其中,$*$ 表示有限域上离散对数运算的二元关系.

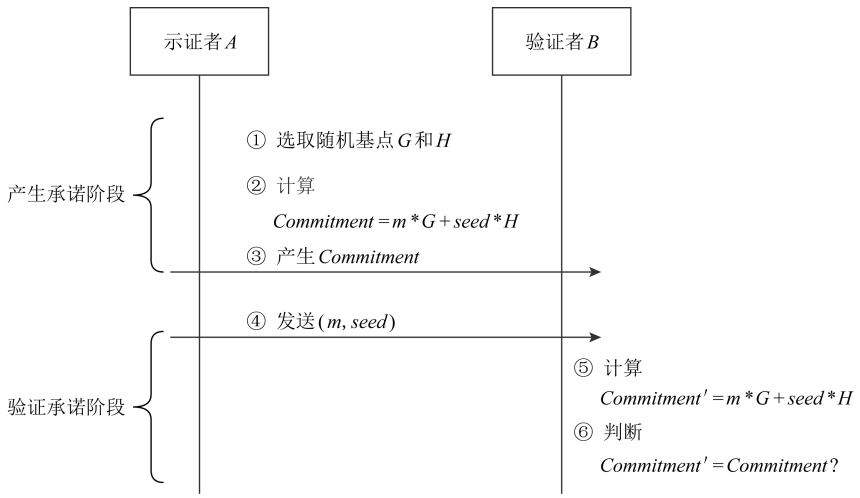

Fig. 1　Pedersen commitment interaction process based on elliptic curve

图 1　基于椭圆曲线的 Pedersen 承诺交互过程

在验证阶段,B 将会再次接受 A 从匿名信道发送过来的验证参数对(m,$seed$)并按照式(1)计算承诺 $Commitment'$.通过比较 $Commitment$ 与 $Commitment'$ 是否相等即可判断承诺是否正确合法.在此过程中,对于验证者 B,表征在区块链上就是一个验证承诺的智能合约,也就是由合约获取验证参数进行承诺验证,在一定程度上保证公示结果正确不被篡改.

至于应用方面,Pedersen 承诺在当前的基于 Mimblewimble 协议的交易输出中包含了约 33 B 的 Pedersen 承诺内容以实现保密交易,并结合 Bullet-Proofs 的零知识证明体系消除对地址和私钥依赖以实现轻量级的隐私保护[13-14].Pedersen 承诺已在数值隐藏、恒等关系验证以及审计验证等方面发挥着不可或缺的作用.

1.2　Schnorr 协议

作为 Sigma 协议(Σ-protocol)的一种,非交互式的 Schnorr 协议[15]是一个非常优秀、简洁且具备零知识性的安全协议.虽然 Schnorr 协议于 1989 年就被提出,但数十年来学界对其探索、研究的热情丝毫未减,近年来在区块链领域更是大展身手.如零知识数据交换协议 zkPoD 中,为了另辟蹊径实现公平交互(fair exchange),就利用了一个扩展的 Schnorr 协议结合 Pedersen 承诺去实现高效性和可扩展性[16].在 Schnorr 协议研究发展中,各种 Schnorr 聚合签名扩展了密码学椭圆曲线数字签名的方案,推动了数字签名的发展.Maxwell 等人在 2019 年的文献中,也提出基于 Schnorr 协议的多重签名方式,并应用到比特币网络中[17].

Schnorr 聚合签名可以分为密钥生成、密钥集聚和、交互随机数、生成单一签名、聚合签名以及验证签名 6 个步骤.交互过程如图 2 所示.

以 3 个签名者聚合签名为例,首先需要彼此之间交换各自的随机数,然后利用各自的签名私钥分别对同一消息进行单一签名,接着就是对单一签名进行聚合,生成一个新签名,最后交由验证者利用验证密钥进行验证.在此过程中,对于验证者,表征在区块链上就是一个验证签名的智能合约,由合约利用预定义的验证密钥对组合签名进行验证,保证签名的合法性.

根据先行研究结合上图签名逻辑,不难看出对单一消息的单一签名或者聚合签名,Schnorr 协议在签名方案上已经非常成熟.

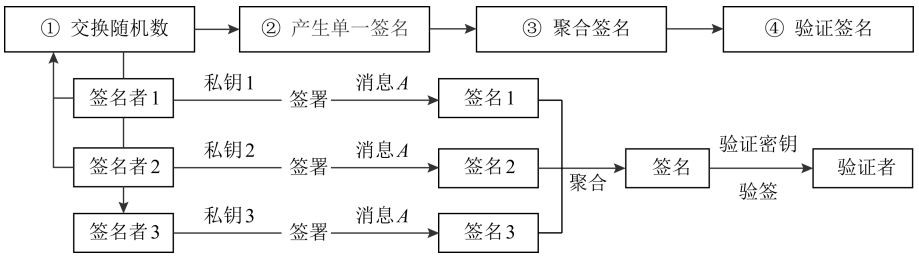

Fig. 2 Schnorr aggregate signature interaction process

图 2 Schnorr 聚合签名的交互过程

1.3 安全多方计算

安全多方计算主要目的是解决互不信任的参与方在保护隐私的前提下协同计算的难题[18].相比传统的数据保密,安全多方计算的优势如表 1 所示:

Table 1 Comparison of Traditional Data Confidentiality and Secure Multi-Party Computation

表 1 传统数据保密与安全多方计算的比较

对照信息	传统数据保密	多方安全计算
计算	数据无法在加密前计算也不能在解密之前被使用	可以在数据加密下被计算并能够获得与明文直接计算相同的结果
安全	需要可信系统管理员	由数学理论保证
存储	需要可信硬件环境和稳定操作系统	无需可信硬件环境和稳定操作系统
使用	静态读取	动态使用

在信息飞速发展的今天,协作情境的出现已经屡见不鲜.安全多方计算输入隐私性、计算正确性以及去中心化性的技术特点正好满足了在这样的场景下参与方希望得到合作利益却又不希望泄露自己数据的客观需求.所以,近年来安全多方计算的研究也在不断深入.Zhu 等人[19]在 2018 年采用动态规划的方法改进了常数轮的多方计算协议,提升了运算效率.周俊等人[20]也曾在边缘计算中分析过电子医疗系统中运用安全多方计算时进行隐私保护,并权衡计算开销.Hastings 等人[21]更是在安全顶级会议 Security and Privacy 2019 上详尽分析了多个安全多方计算通用框架,并在 docker 中将构建环境进行了打包,从各个维度评估了这些框架的优劣.

当然,随着数据隐私问题日益明显,区块链与安全多方计算的结合也开始被纳入加强隐私保护的范畴内.区块链因区块数据难被篡改的特征往往更加强调计算的可验证性而不考虑输入信息的保密性.而安全多方计算则强调的是多方计算过程中对消息的保密性但不能确保数据的可验证性.故二者可进行优势互补,一方面区块链利用安全多方计算提升隐私能力,以便于实施到更多的应用场景中;另一方面,安全多方计算可借助区块链技术进行公开透明不被篡改的交易验证.如智慧医疗、舆情存证、拍卖清算等应用场景上的隐私保护与高效事务处理问题,2 种技术的正交为加快分布式网络中的数据隐私保护提供了可能.

2 安全多方计算协议 BPLSM

本节先提出安全多方计算签名的现状引出 BPLSM 协议,然后从 BPLSM 的架构设计入手,阐释该协议主要的 3 个性质并分析其安全性.

2.1 安全多方计算单一签名现状

一个简单的基于区块链的安全多方计算的单一签名实现如图 3 所示:

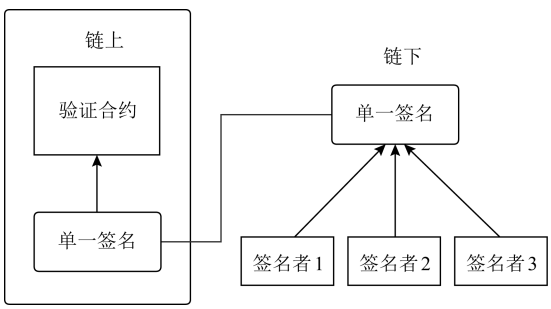

Fig. 3 Single signature based on secure multi-party computation

图 3 基于安全多方计算的单一签名

从图 3 中可以看出签名参与方在链下进行协作签名且只产生一个统一签名交由链上合约对其验

证,判断合法性.而在一些多方计算场景中,参与方的选择可能是多向的,比如投票选举中参与方投同意票或者否决票、交易仲裁中参与方同意返还资金给买家或者释放资金给卖家等.面对这样的多方参与的情形,单一消息的统一签名往往并不能满足需求.

因此在保证安全的前提下提升一定的效率是能够应用的关键.本文考虑使用具有同态加法特性的Pedersen承诺对Schnorr协议进行改进,提出了能够实现对多种不同消息进行聚合签名并具备在区块链上进行高效验签能力的BPLSM协议.

2.2　BPLSM协议架构

本协议的架构设计图如图4所示.从图4中可以看出,首先第一步各签名参与方进行隐匿消息承诺,公式如下所示:

$$MCommitment_i = m_i * G + seed_i * H, \quad (2)$$

其中:G 和 H 为有限域椭圆曲线上2个位置不同的固定点;m_i 为各方签署的互异消息;$seed_i$ 为随机数种子,$i \in \{1,2,3\}$.在作出敏感消息的承诺 $MCommitment_i$ 后,需要将参与三方的消息承诺进行聚合,以便生成一个统一的签名,交由合约验证.

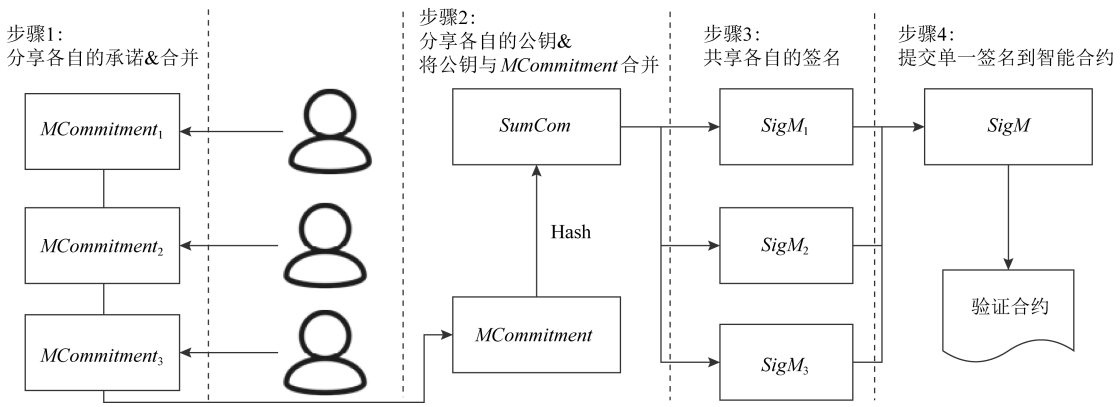

Fig. 4　BPLSM protocol architecture

图4　BPLSM协议架构

有关签名相应的步骤以及计算如下:

1)参与三方按式(2)生成各自的承诺 $MCommitment_1$,$MCommitment_2$ 和 $MCommitment_3$;

2)各方分别私下生成盲因子 r_1,r_2,r_3,并互相公开 $r_1 * H$,$r_2 * H$,$r_3 * H$,作为基于盲因子 r_i 的公钥 R_i,$i \in \{1,2,3\}$;

3)根据各自公开的公钥 R_i,定义组合的消息承诺:

$$MCommitment = \sum_{i=1}^{3} MCommitment_i,$$

$$SumCom = Hash(R_1 \| R_2 \| R_3 \| MCommitment), \quad (3)$$

各方将计算得出的组合交易承诺进行Schnorr签名,即 $SigM_i = r_i + SumCom * prkey_i$,其中 $prkey_i$ 为各方的私钥.并把各自交易宣告 $declare_i$,以及使用的 $seed_i$ 通过一个匿名信道分别传给合约,$i \in \{1,2,3\}$;

4)各方将各自交易的签名交给最后签名的一方进行聚合签名

$$SigM = SigM_1 + SigM_2 + SigM_3, \quad (4)$$

5)聚合完成之后,由最后签名的一方将单一签名对$(R, SigM, MCommitment)$提交到合约,其中 $R = R_1 + R_2 + R_3$.

上述协议签名步骤,通过利用Pedersen加法同态的特性,在不解密各方交易承诺的条件下,直接对密文形式的交易承诺进行组合运算,使得参与三方消息承诺都加入到隐私计算之中,保证了各方在匿名条件下的发言权.此外,因为最后提交到合约的是一个单一签名,所以也减少了合约签名验算的时间,避免了不必要的计算开销.

值得一提的是,具体方案设计中,相关签名参与方是在链下完成业务交互后,将对应的数值变化表达成Pedersen承诺,再对承诺数据进行Schnorr签名进行上链操作,这个过程中无需披露任何隐私数据明文.而上链之后,虽然区块链分布式网络中第三方难以通过Pedersen承诺的密文形式反推出隐私数据明文,但是可验证承诺间的约束关系、签名的有效性以及核实业务交互的合法性.

2.3　BPLSM协议的3个性质

为了证实研究方法中隐私计算的三大交互性质,本节从完备性、可靠性以及零知识性上进行分析.

2.3.1 完备性

诚实的参与方会按照链下签名逻辑,先生成自己的消息承诺;然后接受其他参与方的消息承诺,计算组合的承诺并进行签名;最后向合约宣告自己的消息,整个过程并不会向其他参与方泄露自己签名的承诺,确保了方案流程的完备性.

2.3.2 可靠性

从整个协议隐私加密过程中链上承诺验证和签名验证这 2 个方面的安全性进行分析.关于承诺的验证是否可靠,可以根据式(1)推导验证:

$$MCommitment = MCommitment_1 + MCommitment_2 + MCommitment_3 = (m_1 + m_2 + m_3) * G + (seed_1 + seed_2 + seed_3) * H = \sum_{i=1}^{3} declare_i * G + \sum_{i=1}^{3} seed_i * H. \quad (5)$$

如果式(5)能够成立,则说明最后提交承诺合法,承诺是可靠的.而关于签名的验证是否可靠,可根据式(6)进行推导验证:

$$Sig * H = (Sig_1 + Sig_2 + Sig_3) * H = \sum_{i=1}^{3} (r_i + SumCom * prkey_i) * H = R + SumCom * pbkey. \quad (6)$$

如果式(6)能够成立,则表明签名可靠.各参与方在各方互相公开公钥 R_i 前,对需要进行 Hash 的组合承诺 *MCommitment* 是没有办法预测的,即使这个组合承诺最终是参与方自己计算的,但参与方并没有能力通过挑选 *SumCom* 实现作弊,因为只要 R_i 公示之后组合承诺就被固定下来了.

最后,为防止参与方的真实地址会在验签结束后被使用的匿名地址所追溯,在每次验签结束之后,匿名地址就会被销毁.

2.3.3 零知识性

链上基于 Pedersen 承诺的隐藏性,在合约未验证前任何人都无法从承诺中获取任何有关的敏感数据信息.此外,合约进行验证时,除了最终的聚合签名和参与方的共同公钥,并没有传输大量的数据,验证简洁.

对于多方计算进行组合承诺时,各参与方并未暴露自己用于承诺的随机数 $seed_i$,且一方无法通过作弊手段获取另一方私钥,即无法学习到有关单个 Pedersen 承诺内的任何知识,该协议对每个参与方来说是零知识的.

2.4 安全性分析

链下实现多重签名相比链上实现多重签名而言安全性会更高,主要是前者更依赖于密码学算法,后者更依赖于智能合约,而智能合约在设计上可能会存在漏洞,也就容易存在安全隐患.另外,由于不是每个参与方都需要与合约进行频繁交互,且用户地址均已匿名,因而也就降低了用户被攻击的风险.安全多方计算中还需要考虑以下 3 种攻击者模型.

1) 在诚实者模型中,诚实的参与方总是能够按照设计准则提供正确交易承诺和签名,且不窃取其他参与方输入.

2) 在半诚实者模型中,半诚实参与方按规则提供正确交易承诺,但试图窃取其他参与方交易承诺的内容.因为单个交易的消息是按 Pedersen 承诺进行处理的,所以半诚实的参与方要想窃取其他参与方的交易承诺,需要拿到其他参与方生成的随机数种子 *seed*.但是 *seed* 是由各参与方秘密保管的,且每次签名验签时只使用一次,所以只要参与方不串谋就不可能被轻易获取.另外,对于各参与方的交易签名,半诚实参与方因为无法获取交易签名中的盲因子 r_i 以及用户私钥 $prkey_i$,所以交易签名的具体内容也是很难被窃取的.

3) 对于恶意攻击模型,恶意参与方可能会提供虚假隐私消息承诺的签名,并试图窃取、更改其他参与方隐私交易承诺和结果.比如在存在一个恶意参与方的多方计算中,恶意参与方会提交虚假的承诺,然后使得参与三方组合承诺的计算值不一致,从而破坏签名验签的合法流程.本研究方法并不适用于恶意攻击模型.

此外,借助 Pedersen 承诺对签名的承诺内容进行加密保证,使得任何参与方在合约宣布验签结果前都不会知晓组合承诺内容,以应对攻击者可能的提前终止行为.所以总的来说,对于需要在数据密文形式上直接进行运算和交叉验证的业务,只要不涉及互不透露数据明文的多方协同计算,相比现有的同态加密算法,以 Pedersen 承诺为代表的密码学承诺往往可以提供更好的性能.

3 验证与模拟实验

为证实 BPLSM 协议的可行性,本研究以 Go 语言和 Solidity 语言为主,设计了 BPLSM 协议实验.为了在签名验签流程上体现出 BPLSM 协议高效的客观性,本研究在同等硬件环境下对当前可签署不同消息的主流聚合签名之一的 BLS 签名进行计算

开销对比实验.相关实现代码以及测试代码参照本页注脚①.实验运行的硬件环境如下:

内存:8 核 8 GB;CPU:intel i7 2.60 GHz;硬盘:448 GB;系统运行环境:windows10 家庭版.

补充说明一点,之所以选择 Go 语言进行仿真实验编码的首选语言是因为 Go 语言开发效率高、执行性能好且支持并发,所以利用 Go 语言来编写链下测试代码模拟多方协同签名消息是非常方便的.此外 Go 语言自带的单体测试和压力测试功能,也省去了编写测试签名的繁琐代码的时间.实验中以 3 个参与方为基准,构建的 BPLSM 协议实验的聚合签名效率与基于 BLS12-381 曲线构建的 BLS 签名效率的比较信息如图 5 所示:

```
goos: windows
goarch: amd64
pkg: BPLSM-protocol
BenchmarkBPLSMSumCom-8          1692274              682 ns/op
PASS
goos: windows
goarch: amd64
pkg: BPLSM-protocol
BenchmarkBPLSMSignature-8       1837884              597 ns/op
PASS
goos: windows
goarch: amd64
pkg: BPLSM-protocol/bls
BenchmarkBLSAggregateSignatureN-8        197246             5714 ns/op
PASS
```

Fig. 5　BPLSM protocol and BLS signature stress test information

图 5　BPLSM 协议签名与 BLS 签名压力测试信息

从图 5 中不难看出,相同硬件环境下,利用 Go 语言对 BPLSM 协议签名进压力测试平均消耗的时间比(组合承诺生成时间加上聚合签名时间)BLS 签名的平均消耗时间要少.

为了更加清晰地查看结果,将图 5 上终端测试信息归总为如表 2 所示:

Table 2　BPLSM Protocol and BLS Signature Stress Test

表 2　BPLSM 协议签名与 BLS 签名压力测试

测试内容	测试频次	平均每次测试开销/ns
BenchmarkBPLSMSumCom-8	1 673 246	700
BenchmarkBPLSMSignature-8	2 058 015	543
BenchmarkBLSAggregateSignatureN-8	203 926	5 789

因为 BPLSM 协议的签名过程分为组合承诺的生成和聚合签名的生成,所以需要在压力测试过程中分成 2 块进行分析,而 BLS 签名需要多方参与聚合过程只有一次,所以压力测试的时候只需要对聚合签名进行测压.从表 2 中可以看出同等测压情况下,BPLSM 协议签名时间开销成本比 BLS 签名低,但是如果撇开效率问题,BPLSM 协议牵扯到了 2 次多方交互的过程,所以签名的比较还存在一些局限性.因为存在不可控的环境因素,详细分析请参照 4.1 节.

验签过程需要通过智能合约在区块链上执行,所以本实验中用 Solidity 语言在 Ganache 私有链上实现与 Go 语言中签名内容相适配的验签逻辑.并通过使用 truffle 框架结合 Javascript 脚本语言进行验签效率的分析.仍然以 3 个参与方为基准,给出利用 truffle 框架结合 Javascript 脚本语言构建 BPLSM 协议验签部分与基于 BLS12-381 曲线的 BLS 验签部分.单次实验验签测试的信息比较如图 6 所示:

```
> Artifacts written to C:\Users\Lenovo\AppData\Local\Temp\test--3288-s7Q3uhszTbqy
> Compiled successfully using:
   - solc: 0.4.19+commit.c4cbbb05.Emscripten.clang

  Contract: BPLSM
    √ should verify a valid SumCom (113ms)
    √ should verify a valid BPLSM_SchnorrAggregateSign (73ms)


  2 passing (269ms)

> Artifacts written to C:\Users\Lenovo\AppData\Local\Temp\test--9892-Qz63QqEcQc6i
> Compiled successfully using:
   - solc: 0.4.23+commit.124ca40d.Emscripten.clang

  Contract: BLSMultiSig
    √ should verify a valid BLSAggregateSign (2880ms)


  1 passing (4s)
```

Fig. 6　BPLSM protocol verification and BLS signature verification test information

图 6　BPLSM 协议验签与 BLS 签名验签测试信息

① 实验测试代码地址:https://github.com/york-yang-me/BPLSM-Protocol

从图 6 中不难看出,在测试过程中 BPLSM 协议验签时间为 186 ms,而 BLS 协议验签时间却为 2 880 ms.二者验签测试结果差异很大.

为了能更加准确地对两者在以太坊智能合约验签上进行效率比对,从而证实设计的 BPLSM 协议的优势.我们再次对 BPLSM 协议验签部分和基于 BLS12-381 曲线的 BLS 验签部分分别进行 20 次单体测试,以便加强实验的说服力.如表 3 所示:

Table 3 BPLSM Protocol and BLS Signature Verification Test

表 3 BPLSM 协议与 BLS 验签测试 ms

测试次数	BPLSM 组合承诺验证开销	BPLSM 聚合签名验证开销	BLS 聚合签名验证开销
1	113	73	2 986
2	165	87	3 385
3	120	79	3 055
4	127	79	2 986
5	267	89	3 073
6	188	112	2 816
7	234	74	2 742
8	124	96	2 854
9	123	84	2 815
10	158	85	2 870
11	144	198	2 989
12	151	160	2 817
13	401	128	2 959
14	125	69	2 813
15	166	114	2 820
16	128	68	2 782
17	134	114	2 797
18	149	69	2 973
19	161	73	2 764
20	113	73	2 880

从表 3 中可以看出,本研究设计的 BPLSM 协议在参与三方中的平均验签时间约为几百毫秒,而 BLS 签名的平均验签时间约几千毫秒,也就是说同等实验环境中 BPLSM 协议在验签效率上要比 BLS 签名花费的时间成本少,且无需进行多次单公钥签名消息比较即可验算结果.

为客观详实展现实验现象,图 7 以两者的平均验签时间损耗相比较.可以看出,对于该实验中参与三方的多方计算中,BPLSM 的平均验签时间为 260.8 ms,而 BLS 的平均验签时间为 2 908.8 ms. BPLSM 协议验签时间成本损耗比 BLS 签名验签节省了约 83.5%.

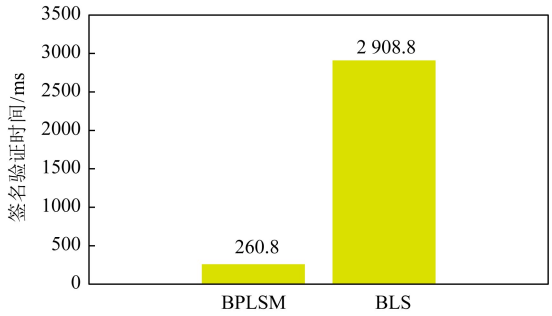

Fig. 7 The average verification efficiency between BPLSM protocol and BLS signature

图 7 BPLSM 协议与 BLS 签名平均验签效率

4 讨 论

4.1 算力开销

多方协作的签名组合放在链下执行相比在以太坊合约中设计多重签名的逻辑而言,利用安全多方计算来提交单一的签名会节省算力开销.因为链上验签是需要消耗节点算力的,而多重签名的验证机制就等同于一次签多个单签名,也就增加了算力成本的开销.

另外从第 3 节的实验中可以看出,BPLSM 协议中多方签名、验签的时间普遍比 BLS 签名、验签时间快.但实验中只进行了验签部分开销的分析,并没有对链下签名部分展开比较,这是因为实验代码模拟的是参与三方同时在线的状态进行的测试.由于 BPLSM 协议中包含 Pedersen 承诺与 Schnorr 签名 2 部分,在签名同时往往需要进行 2 次交互,这就给签名时间的损耗带来了不可控因素,因此并没有在签名部分与 BLS 签名进行详细的比较.

4.2 存储开销

通过 Schnorr 协议进行签名,使用到了公共密钥的 merkle 树来进行存储.虽然在签名计算的参与方少且固定的情形里存储开销并不是很大,但是如果推广到参与方多且不固定人数的情形里,存储空间的开销就会随着参与人数增多而不断地增大.

4.3 隐私承诺

对隐私数据机密性要求高的场景中,Hash 承诺不具备随机性,从而导致提供的数据隐匿性有限.对于单一隐私数据 d,$H(d)$ 值是恒定的,所以利用

Hash 彩虹表即可推出 $H(d)$ 中实际承诺的 d.而 Pedersen 承诺具有便于业务系统在密文形式下对其处理的附加特性,在多个关联的承诺值间实行加密运算和交叉验证便会发挥很大的作用.但 Pedersen 承诺由于不直接提供解密功能,在互不透露的数据明文的多方协同计算中还需要结合其他功能性的密码学算法进行扩展.

5　总结与展望

本研究从多方计算的场景出发,结合区块链技术提出了一种泛用型数据隐私保护的安全多方计算协议 BPLSM.该协议实现了在链上验签并做到了承诺的保证、加密值的正确性和地址的隐藏,并在链下利用 Pedersen 加法同态的特性实现了组合的交易承诺,结合 Schnorr 协议构造了可以签署不同消息的安全多方计算的方案,保证了该链下计算方案中参与方身份鉴别的零知识性和交易签名的正确性.同时,对设计的协议进行了实验仿真.证实了该隐私计算的解决方案在小范围人数固定的多方交易中,验签的时间成本会比当前的 BLS 签名降低约 83.5%.

考虑到协议仅在实验仿真阶段,并没有将设计的 BPLSM 协议放入更贴近实际的多方应用场景中进行分析.在现实场景中链下签名存在诸多不可控因素,如签名方不同时在线、漏签等,因此本研究的下一步工作将会是把协议应用到具体的场景中,如跨境贸易场景,通过 BPLSM 协议来融合当前主流的零知识证明算法来进一步优化.

此外,BPLSM 协议虽然引入随机数实现了信息论安全的最强隐蔽性,但是随着 Shor 算法、Grover 算法等量子密码学算法的出现,在多项式时间内求解离散对数的困难问题开始变得容易[22],协议中基于椭圆曲线的生成元 G 和 H 便不能有效抵抗量子计算的攻击.如何融入量子同态加密[23]、引入量子比特承诺[24]来设计一个后量子安全的新型多方安全计算协议应用至网络舆情治理、区块链舆情存证等方面也是接下来的研究发展方向.

贡献说明:刘峰在创意提出、区块链技术实现及论文撰写上做出了贡献,杨杰在形式化推导,核心协议的辅助实现及论文撰写上做出了贡献,李志斌(通信作者)在本文的数学推导符号及框架设计上进行了建议和指导,齐佳音(共同通信作者)从文章场景设计角度上进行了建议和指导.

参　考　文　献

[1] Liu Feng. Blockchain heat and enterprise opportunities [J]. Enterprise Management, 2018, 442(6): 19-21 (in Chinese)
(刘峰. 区块链热与企业机遇[J]. 企业管理, 2018, 442(6): 19-21)

[2] Chen Weili, Zheng Zibin. Blockchain data analysis: A review of status, trends and challenges [J]. Journal of Computer Research and Development, 2018, 55(9): 1853-1870 (in Chinese)
(陈伟利, 郑子彬. 区块链数据分析: 现状、趋势与挑战[J]. 计算机研究与发展, 2018, 55(9): 1853-1870)

[3] Möser M, Soska K, Heilman E, et al. An empirical analysis of traceability in the monero blockchain [J]. Proceedings on Privacy Enhancing Technologies, 2018, 2018(3): 143-163

[4] Ermilov D, Panov M, Yanovich Y. Automatic bitcoin address clustering [C] //Proc of the IEEE Int Conf on Machine Learning & Applications. Piscataway, NJ: IEEE, 2017: 461-466

[5] Song Jundian, Dai Bingrong, Jiang Liwen, et al. Data governance collaborative method based on blockchain [J]. Journal Computer Application, 2018, 38(9): 2500-2506 (in Chinese)
(宋俊典, 戴炳荣, 蒋丽雯, 等. 基于区块链的数据治理协同方法[J]. 计算机应用, 2018(9): 2500-2506)

[6] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing [C] //Proc of the Int Conf on the Theory and Application of cryptology and information security. Berlin: Springer, 2001: 514-532

[7] Boneh D, Drijvers M, Neven G. Compact multi-signatures for smaller blockchains [C] //Proc of the Int Conf on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2018: 435-464

[8] Qian Qifeng, Cheng Chunling. Pairing-free certificateless group key agreement protocol for wireless sensor network [J].Computer Science, 2015, 42(7): 186-190 (in Chinese)
(钱琦锋, 程春玲. WSN 中基于非双线性对的无证书群组密钥协商协议[J]. 计算机科学, 2015, 42(7): 186-190)

[9] Su Jingfeng, Liu Juxia. Efficient certificateless aggregate signcryption scheme without bilinear pairings [J]. Journal of Computer Applications, 2018, 38(2): 374-378, 385 (in Chinese)
(苏靖枫, 柳菊霞. 不含双线性对的高效无证书聚合签密方案[J]. 计算机应用, 2018, 38(2): 374-378, 385)

[10] Yu G. Simple schnorr signature with pedersen commitment as key [OL]. [2020-10-11]. https://eprint.iacr.org/2020/061.pdf

[11] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing [C] //Proc of Annual Int Cryptology Conf. Berlin: Springer, 1991: 129-140

[12] Maxwell G. Confidential transactions [EB/OL]. 2015 (2016-09-05) [2020-09-15]. https://people.xiph.org/~greg/confidential_values.txt

[13] Fuchsbauer G, Orrù M, Seurin Y. Aggregate cash systems: A cryptographic investigation of mimblewimble [C] //Proc of Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2019: 657-689

[14] Bünz B, Bootle J, Boneh D, et al. Bulletproofs: Short proofs for confidential transactions and more [C] //Proc of 2018 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2018: 315-334

[15] Schnorr C P. Efficient identification and signatures for smart cards [C] //Proc of the Conf on the Theory and Application of Cryptology. Berlin: Springer, 1989: 239-252

[16] Hu Yugang, Irving M, Guo Yu, et al. zkPoD: A practical decentralized system for data exchange [EB/OL]. (2020-09-10) https://secbit.io/zkPoD-node/paper.pdf

[17] Maxwell G, Poelstra A, Seurin Y, et al. Simple schnorr multi-signatures with applications to bitcoin [J]. Designs, Codes and Cryptography, 2019, 87(9): 2139-2164

[18] Smart N P. Cryptography Made Simple [M]. Berlin: Springe, 2016: 439-450

[19] Zhu R, Cassel D, Sabry A, et al. NANOPI: extreme-scale actively-secure multi-party computation [C] //Proc of the 2018 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2018: 862-879

[20] Zhou Jun, Shen Huajie, Lin Zhongyun, et al. Research advances on privacy preserving in edge computing [J]. Journal of Computer Research and Development, 2020, 57(10): 2027-2051 (in Chinese)
(周俊, 沈华杰, 林中允, 等. 边缘计算隐私保护研究进展[J]. 计算机研究与发展, 2020, 57(10): 2027-2051)

[21] Hastings M, Hemenway B, Noble D, et al. Sok: General purpose compilers for secure multi-party computation [C] //Proc of 2019 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2019: 1220-1237

[22] Wang Yongli, Xu Qiuliang. Principle and research progress of quantum computation and quantum cryptography [J]. Journal of Computer Research and Development, 2020, 57(10): 2015-2026 (in Chinese)
(王永利, 徐秋亮. 量子计算与量子密码的原理及研究进展综述[J]. 计算机研究与发展, 2020, 57(10): 2015-2026)

[23] Mahadev U. Classical homomorphic encryption for quantum circuits [C] //Proc of the 59th Annual Symp on Foundations of Computer Science (FOCS). Piscataway, NJ: IEEE, 2018: 332-338

[24] Song Yaqi, Yang Li. Semi-counterfactual quantum bit commitment protocol [J]. Scientific Reports, 2020, 10(1): 1-12

Liu Feng, born in 1988. PhD candidate. Engineer. Senior Member of CCF. His main research interests in blockchain technology and data science.

刘 峰, 1988 年生. 博士研究生, 工程师, CCF 高级会员. 主要研究方向为区块链技术和数据科学.

Yang Jie, born in 1998. Member of CCF. His main research interests include cryptography, blockchain, information privacy and secure multi-party computation.

杨 杰, 1998 年生. CCF 会员. 主要研究方向为密码学、区块链、信息隐私和安全多方计算.

Li Zhibin, born in 1960. Professor and PhD supervisor. His main research interests include computer symbolic calculation and its application in nonlinear science.

李志斌, 1960 年生. 教授, 博士生导师. 主要研究方向为计算机符号计算及其在非线性科学中的应用.

Qi Jiayin, born in 1972. Professor and PhD supervisor. Her main research interests include blockchain, artificial intellience, information management and change management.

齐佳音, 1972 年生. 教授, 博士生导师. 主要研究方向为区块链、人工智能、信息管理与变革管理.