文章编号:1671-4229(2022)04-0021-16

零知识证明在区块链中的应用综述

宋英齐[1], 冯荣权[2*]
(1. 中国科学技术大学 数学科学学院, 安徽 合肥 230026; 2. 北京大学 数学科学学院, 北京 100871)

摘 要: 零知识证明是现代密码学中的基本理论之一,在现代密码学前沿具有广泛的应用场景。随着零知识证明的工业化——简洁非交互零知识的知识论证(zero-knowledge Succinct Non-interactive Argument of Knowledge, zk-SNARK)的诞生,零知识证明在区块链领域得到了强有力的发挥,改进了早期区块链在隐私、效率和存储等多方面的问题,使得区块链能够在现代经济和社会中具有越来越显著的技术先进性。文章综述了零知识证明在区块链中,尤其是在公链项目中具有代表性的应用方式,旨在强调零知识证明等隐私计算技术是未来区块链与实体经济融合,并且发展良好区块链网络生态的必经途径。

关键词: 零知识证明; 隐私计算; 区块链
中图分类号: TP 309.2 **文献标志码**: A

A survey on applications of zero-knowledge proof in blockchain

SONG Ying-qi[1], *FENG Rong-quan*[2*]
(1. School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, China;
2. School of Mathematical Sciences, Peking University, Beijing 100871, China)

Abstract: Zero-Knowledge Proof is one of the fundamental theories in modern cyptography which can be widely used in application scenarios at the frontiers. Along with the industrialization of zero-knowledge proof, the naissance of zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK), zero-knowledge proof has brought its superiority into full play in the domain of blockchain, ameliorating the existed issues of earlier blockchains in various aspects such as privacy, efficiency and storage, and enabling blockchains to be increasingly technologically advanced in modern economies and societies. This paper makes a survey on some representative applications of zero-knowledge proof in blockchain, especially in public chain projects, and aims to emphasize that privacy-preserving computation technologies such as zero-knowledge proof are a necessary way to integrate blockchains with the real economy and develop a good blockchain network ecology in the future.

Key words: zero-knowledge proof; privacy-preserving computation; blockchain

零知识证明(Zero-Knowledge Proof, ZKP)是现代密码学中一个极其重要的工具,它使得证明方在不泄露额外信息的前提下实现对某一事实的证明。零知识证明自1985年诞生以来,从交互式模型走向非交互式模型,为密码学和理论计算机科学带来了丰富的成果。由于其与现实情境十分贴合,具有丰富的应用场景,零知识证明迅速地进入了工程化实施阶段,并且不断在效率和安全性上得到改进。在30多年的发展过程中,零知识

收稿日期: 2022-10-08; **修回日期**: 2022-10-15
基金项目: 国家自然科学基金资助项目(12126301,12231015,61972032);国家重点研发计划资助项目(2018YFA0704703)
作者简介: 宋英齐(1996—),男,博士研究生. E-mail:vinsanity-S@pku.edu.cn
* 通信作者. E-mail:fengrq@math.pku.edu.cn
引文格式: 宋英齐, 冯荣权. 零知识证明在区块链中的应用综述[J]. 广州大学学报(自然科学版),2022,21(4):21-36.

证明已经具备了比较成熟的体系,并且能够与大量密码学工具配合使用。基于零知识证明本身的背景设定,它在以区块链(Blockchain)为代表的去中心化网络中的应用取得了丰富的成果。区块链是一个集合了密码学、分布式计算和博弈论等多学科交叉技术的领域,早期区块链所使用的密码学工具在真正意义上实现了区块链的本质特征,即去中心化。但是区块链距离设想中的能够得到广泛应用的去中心化信任机器(Decentralized Trust Machine)的目标还很远,需要有各项技术上的突破;其中隐私问题、效率问题和存储问题等尤为突出。在比特币、以太坊等传统的工作量证明(Proof of Work, PoW)共识机制的公链上暴露出一系列问题后,十几年来,科学家普遍借助零知识证明对区块链进行了一系列改进,在一定程度上成功解决了这些问题。如果能将零知识证明以及其它隐私计算技术发挥得当,那么理论上讲区块链可以突破各种问题的限制,合理地创建或消除信息不对称,从而搭载各种人们想要实现的功能,构建一个完善的去中心化生态。对于已有的零知识证明在区块链中的应用研究是必要的,因为这些具体应用以及相应问题的解决能够为后续技术突破提供借鉴,也能够为具体应用场景的设计提供现成的方案。当前,公链尚未在国内得到足够的认可,国际已有的公链也并没有在很大程度上影响到传统的中心化行业,究其原因是信任机制尚未成熟,与实体经济和大数据的结合还不够强,因此,区块链需要不断结合新的技术以寻求突破。

1 零知识证明及其工业化

1.1 零知识证明

零知识证明这一概念最早见于1985年的文献[1],其目标是在一个双方收发信息的情景下,信息发送方为证明某个断言的真实性,以某种不暴露额外知识的方式与接收方进行信息交互。该文提出的收发双方利用交互式系统进行重复检验,以达成证明的方式实现了这一目标。由于此处交互性的存在,这样的零知识证明也称作交互式零知识证明。这种证明的模式一方面构建了一类证明系统,它能够让某项断言通过交互信息达成证明或证否,保护了信息接收方实现判断的基本权利,也能满足信息发送方实现证明过程的基本需求;另一方面,其不暴露信息本身内容的特点防止了信息发送方提供的消息被对方利用,保护了信息发送方隐私。

交互式零知识证明是证明者 P 向验证者 V 试图证明某一断言 x 属于语言 L 的过程:P 具有足够的计算能力,使得当 $x\in L$ 为真时,P 能够实现关于 $x\in L$ 这一判断的证明;随后 P 与具有概率多项式时间计算能力的验证者 V 进行信息交互(自然也是多项式次数的),让 V 在交互结束后判断 P 是否通过了证明。此处涉及到零知识证明定义中的3个重要性质:

(1)完备性:如果 $x\in L$ 为真,那么 V 判断 P 通过证明的概率可忽略地①接近1(完备性);

(2)稳健性:如果 $x\notin L$ 为真,那么 V 判断 P 通过证明的概率是可忽略的,不论 P 具有多强大的计算能力;

(3)零知识性:当 $x\in L$ 时,任意的 P 和 V 可能的交互文本分布 τ(在2者的随机性下,交互文本构成一个分布),与任何概率多项式时间的模拟机 M 在模拟2者的交互式证明时给出的模拟分布τ是计算不可区分的。

交互式零知识证明的实现原理描述如下:假设 $x\in L$,则 P 具有能够证明 $x\in L$ 的知识,此时 V 发出一个随机挑战,只有当 P 能够证明 $x\in L$ 时,才能够给出应对该挑战的交互信息;换句话说,只有具有给出通过随机挑战的交互信息的计算能力才能够归约到 P 具有关于断言 $x\in L$ 的证明能力。如果 $x\notin L$,则 P 通常只能通过随机猜测给出回应,于是 P 能通过证明的概率随着挑战次数指数级的减小而减小。而根据随机挑战的正确回应,V 仅仅能够获得"P 具有证明 $x\in L$ 这一能力"的判断,但无法从 P 的证明本身获得任何额外知识。

一个重要的事实是,当 $L\in PSPACE$ 时,总存在关于 L 的交互式零知识证明系统[3-4],换句话说,对于任何一个多项式空间范围内可识别的语言 L,都可以用零知识的方式证明其中的任何一个断言 x 确实属于该语言。实际应用中,我们通常考虑 L 是一个 NP 类语言,因此,$x\in L$ 等价于存在一个简短证据 w,使得 w 可以转化为识

① 设 f: $N\to R_+$ 是一个函数,满足 $\forall p\in R[x]$, $\exists N_p\in N$, 使得 $f(n)\cdot|p(n)|<1$ 对一切 $n>N_p$ 成立,则称 f 是一个可忽略函数。交互式零知识证明的完备性定义中可以将"通过证明的概率可忽略地接近1"加强为"通过证明的概率为1",即具有完全完备性。对于完备性定义的加强并没有改变交互式零知识证明系统可识别语言类的范围,事实上 Furer 等[2] 于1989年证明了对于任何一个可由交互式零知识证明系统识别的语言 L,存在一个具有完全完备性的交互式零知识证明系统(Arthur-Merlin 证明系统)也识别 L。

别语言 L 的一个多项式时间非确定性图灵机 N,在输入 x 时的停机于 N 接受格局的计算序列①。换句话说,此时 P 拥有证明 $x \in L$ 的知识等价于 P 可以计算出一个对应 x 的简短证据 w。NP 类语言的零知识证明可以借助一个密码学承诺(Cryptographic Commitment)来实现,证明者 P 将秘密信息 w 通过密码学承诺绑定下来,以保证后续的挑战始终以 w 作为额外知识参与计算,同时也通过承诺隐藏了 w 的信息。

1.2　非交互的实现

交互式零知识证明的原理十分依赖交互性和验证者随机性。交互性体现在交互式零知识证明的基本逻辑是验证者发出挑战 - 证明者应对挑战,这一流程的不断重复才能推动证明的实现,仅依靠单次挑战应对很难排除证明者的偶然判断对结果带来的影响。而验证者随机性在于验证者借助自身生成的随机性对证明者发起挑战,这种随机性不能被具有更强计算能力和拥有更多额外知识的证明者提前知道,否则他可能有机会提前准备好相应的应对措施,从而提高自己通过验证的概率。

交互性和验证者随机性这 2 个特点却并非适用于所有的应用场景。在一个具有一定用户规模的网络中,他们之间的证明需求可能是非常频繁的,区块链、集体之间的身份认证等就存在这样的情形。更有甚者,证明双方的关系可能会对调,并且每个人可能会和多人产生证明需求关系,如果再要考虑证明双方在空间和时间上的参与差异性,那么会给交互式的证明带来很多麻烦。此外,对一个知识的证明可能需要经过不同的验证者的检验,要求证明者始终配合验证需求随时进行交互也是完全不现实的。总而言之,交互会使得效率降低,同时也带来了双方配合的不便利。同样地,在一个人群庞大的系统里,保持验证者在高频的证明过程中自主生成的随机性,会带来效率的严重降低,因此,借助某种公共而不可预知的随机性也是必要的。以下有 2 种实现非交互式零知识证明的常用方法:

(1)1988 年,Blum 等[5]基于公共参考串模型,依赖 Blum 整数的二次剩余计算困难性假设,给出了 NP 类语言的有界非交互式零知识证明系统;这一系统又在 1991 年被 Blum 等[6]改进。通过非交互的形式,证明者能够单向地产生消息供验证者使用;此外,随机性来源于先于证明生成的一个均匀分布的公共的随机字符参考串。因为公共参考串的随机性是不可被双方所预知的,且不会受到双方的操作影响,所以公共参考串一般认为是由某个可信第三方生成的,或是通过一个安全多方计算来生成。当考虑零知识性时应当注意,由于没有了交互,模拟机难以像交互式中的那样不断重新设置有利于通过证明的挑战以"瞒过"验证者。但在公共参考串模型下,模拟机可以自行生成满足同一分布的公共参考串,甚至可以在没有简短证据 w 的情况下在参考串中嵌入陷门信息。模拟机始终保有的超出验证者的"额外能力",使得 NP 类语言在公共参考串模型下具有非交互式的零知识证明系统。

(2)1986 年,Fiat 等[7]提出了一种将公共随机性的交互式身份证明变换为非交互式的签名方案的方式,称作 Fiat-Shamir 变换。一般地,这个变换能够将一个交互式零知识证明系统转化为一个非交互式的零知识证明系统。这种变换的思想在于,用一个随机谕示机替代交互式证明中验证者的随机挑战,这样就免除了交互的步骤。由于在实际中不存在真正的理想随机源,因此,在应用中通常用哈希函数作为伪随机源。具体而言,证明者自身进行一个交互式证明的模拟,不断将历史交互信息的哈希像作为下一次随机挑战的值。由于认为哈希函数是一个不可预知的随机源,因此,它能模拟验证者在交互式证明中发出的随机挑战。值得一提的是,Fiat-Shamir 变换仅仅在随机谕示机模型下是安全的,对于实际采用的哈希函数,这种方式是不具备可证明安全性的,更有甚者,后来 Goldwasser 等[8]发现了在 Fiat-Shamir 变换下并不安全的 3 轮公共随机性的交互式身份证明,这意味着 Fiat-Shamir 变换本身可能具有使用上的风险。这些原因都表明 Fiat-Shamir 变换实质上是一个启发式的方法,但尽管如此,由于 Fiat-Shamir 变换的易操作性和不需要第三方的特点,它仍然在零知识证明的工程化中受到广泛应用。

1.3　zk-SNARK

零知识证明的工程化要求我们考虑非交互式零知识证明在实用层面上的具体实现,其中最大的变化是应用场景的复杂化。

一方面,稳健性和零知识性具有一种微妙的平衡关系。在非交互式零知识证明的定义中,稳健性是统计意义上的,而零知识性是计算意义上的,这种设置更强调

① NP 类语言可以如下定义:设 L 是一个语言,若存在一个多项式时间界限的非确定性图灵机 N,使得 N 识别 L,则 $L \in NP$。设 $L \in NP$ 且 N 是如上定义中的一个图灵机,当 $x \in L$ 作为 N 的输入时,在 N 的计算树中一定存在某个计算序列的路径停机于 N 的一个接受格局,且该路径的长度是关于 $|x|$ 的多项式规模的。该路径的实质对应着 $x \in L$ 存在一个多项式长度的判据,即为 NP 等价定义中的 $x \in L$ 的简短证据。

于限制强大计算力的证明者在多项式时间内用假证明欺骗验证者。但是在实际情况下,如果倾向于利用零知识证明隐藏简短证据时,就必须考虑到非诚实的验证者利用强大的算力窃取证明中知识的情况。然而统计意义上的稳健性和零知识性不能同时满足,事实上 Fortnow[9] 证明了在满足统计意义上的稳健性的前提下,不存在满足 *NP* 问题的统计零知识交互式证明系统,这同时意味着统计零知识证明语言类是计算零知识证明语言类的真子集。基于这样的原因,不再认为 P 具有无限的计算能力,而是在合理的计算能力下给定了一些额外的秘密知识(如 *NP* 类语言中的简短证据)。事实上,如果证明者拥有过于强大的计算能力,那么许多密码学体制(如公钥加密等)本身就已经不安全了,因此,这样的条件弱化是可以理解的。如果将一个零知识证明系统中的稳健性弱化为计算意义上的级别,而将零知识性提升到统计意义上的级别,则此时的证明系统就称作是一个零知识论证(zero-knowledge Argument)。面对计算意义上的稳健性,对于一般的证明者仍然无法生成假证明,而对于拥有较高计算能力的证明者,调整系统的安全参数能够在一定程度上遏制他们的恶意行为。零知识论证保证了非诚实的验证者无法从证明者的证明中获得额外知识。此外,希望证明者的确是在拥有知识的前提下才能给出证明,而非通过其它的方法绕开知识本身,仅仅证明一个断言属于某个语言,即证明者提供一个正确的证明的难度不会低于证明者拥有知识本身的难度。在 *NP* 类语言 L 中,这就是说希望证明者 P 能够给出 $x \in L$ 的证明等价于 P 拥有秘密的简短证据 w。这样的证明系统称作是一个知识论证(Argument of Knowledge),保证了非诚实的(实际上并不具备知识的)证明者无法在 $x \in L$ 时能给出让验证者通过的证明。当考虑双方都可能存在的非诚实性,就需要一个零知识的知识论证(zero-knowledge Argument of Knowledge)①。

当非交互式零知识证明被广泛应用在大型网络中时,验证计算的速度成为了决定系统运行速度的关键考量,同时大量生成的单向证明会被当作历史数据存储下来,以供多方验证和存证。当证明已经产生并存储在数据库中,零知识性的重要性就更加凸显出来,这是因为计算机的计算能力是在不断提升中的,但是证明本身一旦保存就不会有变化,这进一步印证了采取零知识论证而非零知识证明的必要性。此外,大量存储的需求也限制了生成证明的大小。对零知识证明而言,由于要求至少统计意义上的稳健性,这导致证明长度至少会大于简短证据的长度,然而零知识论证放宽了稳健性的要求,所以给证明大小的压缩提供了更多的空间。证明长度在不断变小的同时,也使得验证效率不断提升,可以看到某些简短证明在零知识论证中被验证的速度快于简短证据在标准的 *NP* 类语言证明中的速度②。事实上,Groth[10] 在 2010 年就设计出了常数大小的零知识论证方案,又在 2016 年给出了证明长度最优、效率最高的零知识的知识论证[11],这更加表明了零知识论证在工程应用中具有重要意义。

2012 年,Bitansky 等[12] 第一次提出了称作零知识的简洁非交互式知识论证(zero-knowledge Succinct Non-Interactive Arguments of Knowledge, zk-SNARK)的概念。在零知识的非交互式知识论证的基础上,zk-SNARK 具有简洁性,即对 *NP* 类语言而言,与该系统的输入 x(及其简短证据 w)相比,其生成证明的长度足够小,甚至是常数规模的大小,这表明 zk-SNARK 是一类高效的零知识协议。zk-SNARK 在早期主要还是延续了依赖可信设置③(trusted setup)生成公共参考串的模型。2013 年由 Par-

① 本文后续的叙述中,通常用零知识证明代表技术本身,而不作为具体的与零知识论证、知识论证等相区分的概念。

② 具体地说,如果将 V_p 和 V 分别视作零知识论证系统的验证算法和 *NP* 语言识别中检验简短证据的验证算法(对应同一个 *NP* 类语言 L),设 π 和 w 分别是某个 $x \in L$ 对应的零知识论证和简短证据,则可能 $V_p(x,\pi)=1$ 的计算速度要快于 $V(x,w)=1$ 的计算速度。也就是说,如果将零知识论证系统作为断言 x 属于某个 *NP* 类语言的概率性意义上的识别器,其接受该断言的速度(即在零知识论证系统中被证明的速度)可能会快于同一语言依赖简短证据实现断言识别的严格意义上的证明。

③ 可信设置通常是一个安全多方计算的程序,由系统各个参与成员或少数代表成员共同参与计算。可信设置是为了确定系统中进行非交互式零知识证明的基本算法、公共参数、公共参考串等公共数据,这些信息将为各成员执行计算而服务。参与可信设置的各个成员都将各自持有一部分秘密信息,全体的秘密信息将在安全多方计算中生成这些公共数据,然后在公共数据产生之后遗忘这些秘密值。可信设置通常只需生成一次,即可长期、反复地用于证明生成等计算过程。可信设置中公共数据的安全性和随机性是整个系统安全的保障,因此,传统的可信设置可能是秘密进行的,且成员选择和计算过程不透明,这反而会造成可信设置的信任问题。另外,作为安全多方计算的可信设置可能遭受秘密泄露、合谋攻击等风险,这进一步加大了系统成员不认可可信设置的可能性。

no 等[13]提出的基于二次算术电路证明的 Pinocchio 协议,是第一个达到工程化可实现的 zk-SNARK 方案。此后一段时间的 zk-SNARK 在运算性能上不断突破,特别是 2016 年的 Groth16 协议[11]给出证明长度只包含 3 个群元素且验证时只要进行 3 个配对计算的 zk-SNARK,此时证明大小已经接近理论最优,是目前 zk-SNARK 协议性能的标杆。然而这些方案非常依赖可信设置的步骤,难以在非信任的条件下进行,同时单次可信设置所生成的参考串只能适用于特定电路的证明,不利于证明需求的扩展。Groth 等[14]定义了通用可更新(universal and updatable)公共参考串模型,该模型能允许成员在不认可传统可信设置的情况下,将公共参考串进行可验证的更新,从而解决了可信设置的信任问题;进一步地,生成的公共参考串也能够适用于一般的电路,不需要针对不同电路进行重新设计。目前应用前景较好的通用可更新 zk-SNARK 模型是改进了 Sonic 协议[15]的 Plonk 协议①,在其基础上的实现方案 Halo2[16]将会成为 Zcash 项目的下一代解决方案。这说明解决可信设置问题的还有将公共参考串生成透明化的设计,如 2018 年诞生的无需可信设置、准线性时间的后量子安全的 zk-STARK 方案[17]。而 Bünz 等[18]基于离散对数假设和 Fiat-Shamir 变换提出了 Bulletproofs,它通过提供在一段时间内秘密承诺值的简短证明,降低了分布式系统中的证明成本,并且 Bulletproofs 的良好性质还使得其经常用于范围证明(但由于 Bulletproofs 与其它 zk-SNARK 算法的证明时间相较而言不够短,这导致其在很多情况下不被归于 zk-SNARK 之类)。Bowe 等②利用递归思想设计了 Halo 协议,该协议通过把计算量较大的验证过程进行分摊,然后借助zk-SNARK 的高效性以递归方式完成证明,其升级版本 Halo2 也包含了大量的递归零知识证明思想。

zk-SNARK 作为目前具有实用价值的零知识证明工具,在具体的工程应用中也有不同的侧重,这取决于不同 zk-SNARK 方案的特点以及实际应用场景,例如,实现零知识的方式是基于何种公共参考串模型,还是使用了其它方法;生成证明和验证证明的时间复杂度,证明大小等方面是否具有效率优势;是否具有通用性或可更新性;是否具有抗量子性;密码学假设是怎样的,等等。尤其是零知识证明在区块链这一应用场景下,不同的区块链根据设计和需求对零知识证明方案也具有不同的考量,具体可参照文献[19],这里不再赘述。本文将侧重于综述零知识证明是如何具体应用到区块链中的,它们究竟以何种方式解决了区块链的技术困境,并提升了区块链的相关性能。

2　区块链背景下的零知识证明

区块链最早是从 2008 年中本聪发表的比特币(Bitcoin)白皮书[20]中抽象出来的数据结构。区块链本身是一个分布式(distributed)的数据库,它的核心属性是去中心化(decentralization),数据在分布式网络的各个节点上存储或处理。当数据发生更新时,网络的某些特殊节点会以某种方式收集更新数据并打包成区块,通过网络将这些区块广播出去,这样各个节点之间可以同步数据更新。

传统的中心化系统往往存在结构低效、无隐私、不透明、数据可用性差、信息鲁棒性差,以及运营成本高等内在缺点,这归根到底是普通节点与中心化机构之间存在数据交互和数据使用的信任问题。区块链正是为了解决这种信任问题而诞生的:①区块链的分布式网络中并不存在永久意义上的中心化节点,各个节点具有完全平等的身份,将需要集中处理的数据分散到整个网络的参与者中,每个节点都能够成为数据处理和更新的一环;②用户的身份信息以公钥的形式存在于链上,数据处理以公开的方式打包成区块,并通过哈希函数将这些区块连接在一起,既能在一定程度上隐藏用户,又做到了透明性。哈希函数将数据绑定,避免了历史数据的篡改,同时这个网络通过广播的形式实现各个节点的同步,数据不会因为局部节点的丢失而被抹除。此外,哈希函数和签名则保证了信息对接的稳定性和基本的信息安全,只要不是基于系统的密码学上漏洞的攻击(如用户自身密钥泄露等),那么损失仅会在局部出现,而不会波及到整个网络,由此,系统的安全性只需要通过巩固其相应的密码学协议和各个密码学工具即可得到保证。区块链基于分布式系统的信息传递和博弈论,以低分化的群体实现共识,并通过一定的激励机制保证用户

① Gabizon A, Williamson Z, Ciobotaru O. PLONK: Permutations over Lagrange-bases for oecumenical noninteractive arguments of knowledge. IACR Cryptology ePrint Archive, 2019: No. 953.

② Bowe S, Grigg J, Hopwood D. Recursive proof composition without a trusted setup. IACR Cryptology ePrint Archive, 2019: No. 1021.

以积极的、诚实的姿态参与进来,这使得整个网络中的节点都能参与到共识当中,提高了信息处理的自主性。区块链是信任机器(Trust Machine)的模型,它能够在不借助任何第三方力量的基础上,实现多方之间安全、透明的信息处理和交互,在诸多现实社会中的场景都有很大的应用价值。

2.1 早期区块链的发展和若干问题

区块链在早期的发展中已经形成了依照成员许可模式的划分。比特币、以太坊等以去中心化为核心特征的区块链系统不需要进行任何形式的成员审核,任何人通过在本地生成一对公私钥,就可以成为系统中的用户,这种系统称为无许可系统(Permissionless System),无许可的区块链系统也称作公链(Public Chain)。与之相对的,如果成员需要关联自身的一些信息,通过审核的方式才能加入的区块链系统称作许可系统(Permissioned System),通常分为私有链(Private Chain)、联盟链(Consortium Chain)和混合链(Hybrid Chain)。许可系统通常会保有一部分子系统充当第三方的职能,用于协助实现成员审核、执行协议与监管等目的。许可系统通常不具备完整的去中心化性质,它的一些性能的完成一般需要借助第三方的力量。

公链通常最能够体现区块链作为新兴技术的革命性意义,因为公链最大程度地将数据处理的权利转移到了一个集体当中,并且集体中的成员具有相同的权利。公链最早带来了去中心化的共识机制,并在此基础上将区块链网络与传统的中心化网络区别开来。尽管区块链已经具备了解决诸多中心化系统弊病的能力,但是区块链本身也产生了一些新的问题,如果将早期的公链和中心化机构作对比,其仍在共识机制、可扩展性、安全性和隐私性等诸多方面存在明显的短板。

(1)共识机制

作为一个去中心化的数据结构,区块链的数据处理权分布在整个网络的各个节点中。为了保证高效处理,必须通过某种方法选出少量的节点获得临时的数据集中处理权,而其它节点则负责监督和验证这一过程的正确性,这种节点成员之间的数据处理权竞争方式称作区块链的共识机制(Consensus Mechanism)。比特币、以太坊1.0等早期区块链的共识机制是工作量证明PoW,系统借助随机性给出的困难问题只能通过暴力求解,让成员以自身算力作为竞争筹码以获得数据处理权和奖励。共识机制是一个区块链的核心属性,它决定了区块以怎样的方式延长下去,因此,还会对区块链的其它方面产生影响。工作量证明PoW的一个显著弊端在于,算力消耗巨大但能量转化率低,进而导致交易吞吐量低下、环境破坏,以及挖矿垄断等问题。

(2)可扩展性及其它性能指标

区块链的可扩展性是一项相当重要的指标,在网络规模扩大、数据量和历史不断累积、应用挑战不断增强的情况下,它是衡量区块链保持良好运行能力的指标。具有可扩展性的区块链通常具有更灵活的设计,能够基于新成员、新功能和新模块有更强大的容纳性,同时自身要保证一定的性能要求。广义来说,可扩展性与衡量区块链的许多性能指标有关:①网络传输协议:是否支持高用户量进行稳定的网络传输,其中要考虑网络运行成本和维护成本;②吞吐量:衡量区块链进行信息处理的效率(在基于加密货币的区块链中,吞吐量相当于区块链单位时间处理交易的数量),考虑到区块链未来会成为Web3.0互联网时代[①]的底层架构,区块链的吞吐量必须要提升到比肩于Web2.0时代电商和银行等大型交易机构的水平,甚至更高的吞吐量;③计算能力:包括智能合约(Smart Contract)支持、协议复杂度、通信复杂度、出块时间、验证时间和计算成本等多方面在内的考量,计算能力既与信息处理效率相关,又与区块链功能的实现密切相关;④存储能力:区块链本身是一个大的数据库,随着区块的产生历史信息会不断积累,同时由于节点需要依据历史信息进行验证,不能将这些历史信息直接抹除,因此,区块链必须应对去中心化存储的问题;⑤最终性(finality):数据处理信息从被广播到实现共识并真正执行(不可逆转)的效率往往更能反映出区块链处理信息的能力,特别地,最终性也衡量了区块链

① 2003年以前的互联网模式称为Web1.0,特点是网页提供信息内容,人们对其上的内容是可读但不可交互的。过去十几年引领网络交互、自由创造、电子商务、娱乐信息、产业互联的Web2.0时代则是在Web1.0可读的基础上,使得信息灵活可用。各个Web2.0时代的龙头企业通过网络效应提供了丰富的网络服务,将基础设施与网络创建互联,人们可以在网络上享受服务,创造更多的信息。而Web3.0则是近两年才火热起来的新概念,它能够基于区块链创建连通全球的去中心化生态,最显著的特点是数字资产和数据的可拥有性。在Web2.0中,信息由互联网中心巨头统一调配和管理,用户没有数据的可支配权,而Web3.0互联网中的个人权限进一步提高,在数据互联的基础上,以各种新兴技术创建更完善的互联网体系。

分叉产生的难易程度,这侧面反映了共识算法的有效性。早期区块链的可扩展性问题一直使得它作为新兴的互联网技术却只能以较慢的速度融入实体经济,更难以期待它能在很短的时间内撼动传统的中心化基础网络设施。

(3)安全性、隐私与监管

作为新兴的互联网技术,区块链的安全性自然是一个必不可少的重要标准。由于去中心化使得信任从第三方转移到了密码学和算法协议中,这对区块链的安全性提出了越来越高的要求。区块链的共识协议、网络传输协议和随机性模型等通常都有一定的安全性假设,并且要考虑诸多潜在的攻击模型及其应对措施。一个安全的区块链协议需要考虑来自于密码学、分布式系统和博弈论等多个层面的挑战,并且需要随着技术的进步不断提升安全性。此外,早期公链的过于透明使得用户在链上数据的隐私性难以保证,由于完全没有提供区块链层上的隐私机制,用户很可能因为隐私泄露遭受外部攻击、甚至蒙受财产损失。隐私的缺乏使得公链的各个成员之间处于一种非信任状态,多方互信局面的缺乏进一步阻止了区块链搭载新的功能,造成了区块链长期在实体经济领域难以得到应用的尴尬局面。除了内外可能遭受的风险之外,早期区块链对于这些可能存在的风险同样缺乏合理的监管措施。对于成熟的互联网体系而言,监管系统一定是必不可少的,它能够协助区块链更好地实现正常运作,敦促用户的良好行为和防控风险,监管系统还有助于系统维护和发起议案,是未来去中心化生态的必要建设。需要强调的是,去中心化监管一定是指区块链内部以线上的方式进行的监管,而非借助任何第三方机构力量参与的区块链外部的监管。

事实上,区块链的各种特征和技术指标也都是相互关联的,其具体性能会受到应用场景和协议特点的影响。比如许可链具有比公链更强的计算力、技术支持与监管,而公链是去中心化的,具有更大的潜力和未来市场;共识机制往往决定了区块链生态的上限,PoW 的区块链在性能上的综合表现明显劣于后来采用股权证明(Proof of Stake, PoS)的区块链;吞吐量高的区块链不一定意味着更快实现最终性,可扩展性强的区块链通常也会具有更强的性能表现和更完善的区块链生态,等等。需要认识到的是,大多数区块链还在随着技术的突破不断取得新的进展,当前的性能表现也并非是停滞不前的(表 1)。

表 1　部分区块链项目的特征和技术指标对比

Table 1　Comparison of features and technical indicators of some blockchain projects

特征和技术指标	比特币	以太坊 1.0	Algorand	HLF	Mina	Dfinity	Polkadot	Avalanche
许可性	公链	公链	公链	联盟链	公链	混合链	公链	公链
共识机制	PoW	PoW	PoS	Raft①	PoS	PoS	PoS	PoS
智能合约	不支持	支持	支持	支持	支持	支持	不支持	支持
监管	不支持	不支持	不支持	支持	半支持	支持	支持	半支持
可扩展性	不支持	不支持	支持	支持	支持	支持	半支持	支持
出块时间	10 min	14 s	4.5 s	2 s	5 min	2 s	6 s	2 s
吞吐量	4.6 笔/s	15 笔/s	1 000 笔/s	320 笔/s	22 笔/s	11 500 笔/s	1 000 笔/s	4 500 笔/s
最终性	1 h	5 min	5 s	15 s	33 min	2 s	1 min	3 s

表 1 比较了部分区块链项目的各项特征和技术指标。区块链经过十几年的发展,从最早期的去中心化电子账本的区块链 1.0 时代,进入到搭载了智能合约的区块链 2.0 时代。随着区块链和实体经济的进一步结合,应用场景进一步扩大,生态建设进一步完善,还将迎来区块链 3.0 时代,与此同时,区块链的性能和安全性也面临着更多的挑战。

传统区块链(尤其是公链)在实际执行中存在一系列问题:①针对共识机制等区块链构件的恶意节点的内部攻击和外部攻击:由于区块链代表着新兴互联网技术的广袤市场,加密货币存在着巨大的经济潜力,因此,攻击者会想尽各种方法从系统漏洞中获利;②区块链自身的性能无法满足实际需求,尤其是早期区块链的可扩展性无法适应庞大的用户市场,又缺乏隐私保护机制,无

① 基于联盟链的性质,Hyperledger Fabric 1.0 并未采用任何常见的共识算法,而是依赖其自有的排序算法 Raft 实现共识;Hyperledger 2.0 又添加了新的 Kafka 算法。

法与实体经济和各种应用场景紧密结合;③区块链能够搭载的功能仍有待完善,在很多方面还不能达到作为新的互联网生态体系的底层构建要求。

一系列问题催生着区块链技术的推进,尤其是隐私计算在其中发挥了巨大的作用。隐私计算(Privacy-Preserving Computation)是实现在交互网络中以保护各方隐私和系统安全为目标的计算综合技术体,它集合了密码学与计算理论领域的若干方面:安全多方计算(Secure Multi-party Computation)、零知识证明、同态加密(Holomorphic Encryption)和可信执行环境(Trusted Execution Environment),等等。区块链作为一个多方交互的去中心化网络,其能够真正融入社会体系的关键就在于安全的可计算性,因而区块链自然与隐私计算应用具有极高的匹配度。

2.2 公链项目中的零知识证明

本节将综述几个公链项目如何通过引进零知识证明,使其在各项特征和技术指标上取得突破的。

2.2.1 Zcash——隐私货币

几乎所有的公链都有这样的一个特点,即任何用户只需在本地生成一对公私钥即可成为公链的一员,当用户利用自己的公私钥进行交易时,公钥和签名就会作为公开信息放在区块中以供各个节点验证交易的合法性。在这样的过程中,用户的社会身份与区块链中的数字身份产生了分离,在区块链中用户仅以公钥标记,因此,一方面社会身份得到了一定的隐藏;另一方面,其它节点验证的需求使得交易细节必须在区块链上公开,也正因此区块链是透明的。但追溯到本质上讲,这样的公链并没有深挖区块链要采用公钥隐藏身份和透明化数据的根本原因:①隐藏身份的目的归根到底是要将用户的社会身份和私人信息同数字身份完全隔离,但是即使是在区块链上显示的是公钥信息,随着用户使用这一数字身份在区块链中的交互历史不断累积,攻击者结合外界信息依靠统计学的方法想要通过数字身份关联到社会身份并非不可能;②之所以要求区块链的透明性,本质上是为了各个节点能够参与到合法性的验证当中,但是透明性导致了交易金额被披露,数字身份也公之于众。当然数字私人信息的完全公开并不是实现节点验证的唯一方法。

2014年Zerocash(Zcash)白皮书发布[21],它的目标就是在保护数字身份和数字私人信息的基础上,维持区块链的可验证性。换句话说,Zcash实现了一个可匿名交易的区块链,这个匿名甚至是数字身份上的意义。Zcash实现匿名交易的大体思路是将用户要消费UTXO①中的货币转化为关于这部分货币的承诺,用唯一的唯一的序列号标记转移的货币,再用零知识证明保证交易的合法性。

为了实现匿名交易,首先用户需要将自己在链上的持有货币隐藏起来,具体而言就是自己的公私钥和货币金额,只有用户自己知道链上的哪些货币是属于自己的。当匿名货币被交易时,这些隐藏的信息仍应该被保留,但同时要公开一些关于交易的信息使得网络上的节点可以验证这些信息,因而这些信息和交易中的匿名货币具有绑定关系(这是防止货币借助匿名性被双花)。公开的信息还需要满足零知识性,也就是任何人无法通过公开信息反推出隐藏币的信息。值得注意的是,匿名性使得交易双方的计算在一定程度上产生了分离,因为链上的信息并不直接反映交易双方的身份,所以公开信息还要使得收款方能够且唯一能够知道链上进行了一笔转到自己的交易。需要强调的是,匿名交易并不代表用户的公钥地址是不为系统中其它成员所知的,否则交易也就失去了目标。事实上,只要能够在交易中隐藏交易双方的身份,即使交易实际发生了,其它节点也无从得知是哪些用户实际参与了交易,这样匿名性的效果就达到了。因此,在匿名交易系统中,用户将匿名货币放在区块链上,当匿名货币的归属发生变化时,则交易双方交易的实质是创造新的匿名货币作为接收者的归属,而公开信息则指示之前具有归属的匿名货币已经被转移了。

Zcash一开始是建立在比特币系统之上的,于是Zcash需要将一定数量的比特币(非匿名货币)转化为匿名货币,发起这样的交易过程称为铸币(Mint Process),而在Zcash系统中发起匿名货币的直接交易过程称为注入(Pour Process),对于这样2个交易过程,收款方接受匿名货币的过程称为接收(Receive Process)。

Mint相当于匿名货币系统中的铸币交易,用户选取一些秘密的种子后,分别计算匿名币的承诺m、匿名币c,并产生一个广播在链上的交易$\mathrm{tx}_{\mathrm{mint}}$。当系统中产生一个

① 即Unspent Transaction Output的缩写,是比特币等以余额模型记账的区块链中重要的数据结构。整个比特币系统的交易链实际上维护了一个潜在的称作UTXO的数据结构,这个数据结构相当于是一个列表,记录了对应各个账户上的余额信息,具体指示了系统中当前所有未花费比特币的所属权,但是UTXO并不是显式地记录在区块链上的,而是由全节点通过追踪所有链上的交易信息不断更新得到的。

被 Mint 的匿名币 c 时,c 中的承诺 m 会加入到链上一个承诺列表中。实际情况下,这些不直接具备数据可用性的承诺值一般会以 Hash Merkle Tree[①] 的形式显示出来。

Pour 将在匿名系统中归属于支付方的匿名货币转化成为归属于接收方的匿名货币,相当于匿名货币系统中转账交易被广播的过程。具体而言,假设支付方已经拥有了放在自己地址下的匿名币 c,当他要将等值的匿名货币转移到接收方的地址时,并非直接将匿名币 c 的信息发送给对方,而是要在链上创建一个新的等值匿名货币 c',并计算对应于原匿名币 c 的一个序列号,以证明之前归属于自己的货币已经被转移。同时支付方要提供如下一些计算正确性的零知识证明:

- c 和 c'是计算合规的匿名币;
- 原匿名币的公私钥关系是匹配的;
- 匿名货币转移的序列号是计算正确的;
- 原匿名币的承诺是系统中 Hash Merkle Tree 的一个叶子顶点;
- 交易前后 c 和 c'的金额是对等的。

以上 5 条断言对于支付方而言是一个具有简短证据的 *NP* 问题,这些简短证据就是在正确执行计算时产生的所有中间过程,于是他可以给出该 *NP* 问题的一个 zk-SNARK 证明 π_{pour}。支付方把 Merkle Tree 的根 Hash、序列号、新匿名币承诺、π_{pour} 和 Pour 过程的所有随机种子用接收方的公钥加密后的密文打包到一起,作为 Pour 交易 tx_{pour} 广播到网络中。

Mint 交易和 Pour 交易被广播后,区块链上的节点将验证相应的交易合法性,特别是在 Pour 交易中,节点除了验证零知识证明 π_{pour} 之外,还要检查根 Hash 值(是否出现在历史记录)与序列号(是否尚未出现)。

Receive 从区块链上归属于自己的匿名交易中还原出新的匿名货币,相当于匿名货币系统中转账交易的执行。由于只有 Pour 交易的接收方才能打开 Pour 交易计算时的随机种子,因此,只有接收方才能还原新的匿名币。

从全局来梳理,可以看到零知识证明是如何创造 Zcash 上的隐私环境的:①Mint 交易将非匿名币进行承诺,使其转化为某个地址所持有的匿名币;②Pour 交易通过公开一个与旧匿名币承诺相关的序列号,使得旧匿名币变为已花费状态,同时产生一个新匿名币,转化为接收方地址所持有的匿名币。当一笔交易出现在链上,接收方得到一个匿名币及其承诺,同时得到了未来消费这个匿名币需要宣称计算的隐含序列号,序列号和匿名币的承诺都是由种子 ρ 生成的,因此,2 者处于一种绑定关系。一旦这个匿名币再被消费,则链上的信息可以指示原有匿名币已经发生了转移。而以交易发起方的视角来看,生成新匿名币的过程就使得他知晓对应新匿名币的承诺,然而发起方只能看到链上存在这样的一个新匿名币的承诺,却不能追踪该匿名币将来是否再次发生交易转移,因为交易发起方不具备接收方的私钥,无法识别新匿名币产生的序列号。对于链上其它用户,只能看到链上的 Mint 交易和 Pour 交易,但是不能察觉到有关交易双方地址和货币量的任何信息;而参与验证的节点并没有获得任何额外信息,只能查询到链上产生了新的承诺或新的序列号,但是无法从证明中获得交易细节。此外,序列号与被转移的匿名币在链上的承诺绑定,保证了同一个匿名币不能发起 2 次交易,避免了双花问题。

Zcash 运用零知识证明技术,使得交易在完全匿名而可验证的情况下进行,交易双方和交易的货币本身具备了隐私性,不必担心任何交易之外的第三方能够通过交易获得额外信息。

2.2.2　ZK Rollup——区块链 Layer 2 扩容

早期公链的可扩展性很差,尽管比特币和以太坊 1.0 引领的区块链交易市场开辟了一个新的时代,但作为交易处理的去中心化网络,比起技术已经相当成熟的中心化交易平台还是相去甚远。比特币 10 分钟才能产生一个区块,且吞吐量仅有至多每秒 7 笔;以太坊尽管仅需十几秒就可以产生新区块,但是搭载了智能合约的以太坊每个区块仅能包含不过几百笔的交易量,这造成它每秒能够处理的交易量也仅有数十笔。相较于 Visa、PayPal 等中心化平台每秒数千笔、淘宝等电商每秒上万笔甚至数十万笔的吞吐量,这些公链远远无法承载一个能够广泛应用的交易市场。如果仅仅依靠增加区块容量的方法,则会造成矿工和节点打包区块和验证的工作量加大,这样的结果使得对算力的依赖越来越大,而算

① 以哈希函数值组织起来的一个树状结构,对于已将某个整体数据划分后的若干子数据、或一个现成的数据列表,将各个数据作为叶子节点逐层取 Hash 值作为父节点构成的二叉树即为 Hash Merkle Tree。Hash Merkle Tree 的顶点称为根 Hash 值,当整个数据中有任何一处变动,由哈希函数的(伪)单向性可知根 Hash 值一定会产生相应的变化,因此可用于判断数据篡改。此外,通过访问相邻的 Hash 值,可以检测到 Hash Merkle Tree 的局部变化。综上几个特点,Hash Merkle Tree 可以起到数据压缩、检测数据改动、为路径证明提供依据的作用。

力分布的不均匀会使得区块链上各个节点的角色分化明显,最后将会偏离去中心化的本质。以太坊创始人Vitalik曾提出一个关于区块链属性重要的观点——三重困境(Trilemma),即常规方法无法同时实现区块链的可扩展性、去中心化和安全性。特别地,就已有的网络模型来看:

- 传统公链能够同时满足去中心化和安全性,但是可扩展性很差;
- 中心化交易平台(电商、银行、部分许可链)具有很高的吞吐量和安全性,但是没有去中心化的属性;
- 传统多链的生态系统(由若干不同区块链通过跨链协议构成的生态)是去中心化的,且能够很好地支持可扩展性,但只要其中一个链受到攻击,则整个生态将被打破。

因此,在不改变公链去中心化和安全性实质的基础上,提升公链的可扩展性的方案成为了区块链研究的一个重要课题,这样的方案称作区块链的扩容(scaling)方案。

造成比特币、以太坊1.0等早期公链数据处理过慢的原因主要有2点:①过度依赖算力的PoW共识机制使得链上节点产生了一定程度的分化,为了让包含若干条交易记录的区块产生,矿工反而需要为了达成Hash目标消耗更多的时间和算力(尤其是比特币)。②当区块打包到链上时,所有的矿工和节点(尤其是全节点)都需要对区块的内容进行验证,并且随着交易数据量的大量积累和新节点的加入,这些新节点(尤其是全节点)需要追踪整个系统庞大的交易记录量,考虑到各种具体因素,这一验证环节也将造成区块达成最终性的效率变得很低。

基于以太坊1.0,有众多的区块链扩容方案被提出[22-25],这些方案主要分为2种:①链上的扩容,称作Layer 1扩容方案,主要采用分片(sharding)技术,能够在一定程度上提升整个链的处理速度;②链下的扩容,称作Layer 2扩容方案,相对于直接改造区块链本身协议的Layer 1扩容,Layer 2扩容由于是在链下进行,不必直接受制于区块链本身的限制,也正因此有一系列不同的方案被提出,如状态通道、侧链、子链等。但这些方案或造成货币流通问题,或存在安全性问题,可验证性或数据可用性差。

Rollup方案则同时兼顾了数据可用性和计算效率,通过主链上的一个Rollup智能合约,实现状态记录、信息转移和存取款操作。此外,Rollup能够将Layer 2中的交易数据压缩到主链中,保证链上的节点能够通过这些压缩数据实现验证和状态更新。当交易产生时,Layer 2的矿工会在链下打包计算压缩数据和新的状态,并将其提交到Rollup合约中,合约验证这些数据后再放到主链上,进而能够实现主链的状态更新。当Rollup合约或链上节点指出了Layer 2中的问题,那么提交该数据的矿工将会受到扣除货币的惩罚。Rollup方案中被广泛使用的是Optimistic Rollup(如Loopering[26])和ZK Rollup(如zkSync[27]),2者分别能够将以太坊的吞吐量上限提升到500笔/s和2 000笔/s,这已经是足以和Visa、PayPal这样的支付渠道相提并论的结果了。

在Optimistic Rollup方案中,尽管Layer 2会给主链提供能够让其它节点交易的状态更新信息和部分可验证的交易信息,保证了数据的可用性,但是它存在一个挑战期,在此期间链上节点利用链上可用数据验证交易合法性,如果存在非法交易,则可以生成一个欺诈证明。不过一旦挑战期结束,交易就会被最终确定并无法修改。而Optimistic的安全性是基于节点行为的,如果主链上的节点没有做足够的验证计算(如果真如Optimistic其名,大家乐观地相信可用性数据即可保证交易准确无误),那么至少得保证系统中的每个账户要时刻注意与自己相关的交易细节,但这一点基于理性行为的并不是一个现实的考虑。利用Optimistic Rollup的这一机制漏洞,攻击者可能从潜在的非法交易中获得收益。

在ZK Rollup方案中,Layer 2的矿工在提交压缩的交易数据的同时,还会附上一个合法性证明(零知识证明)。注意到,Layer 2扩容方案的逻辑是保持交易数据更新的内容不变,同时将交易数据的验证和打包的工作转移到链下进行。因此,这个合法性证明是为了证明"矿工确实对被打包的交易进行了交易数据合法性的验证计算,验证计算的结果是这些交易都是合法的,同时计算的状态更新也是正确的"这一论断。零知识证明的安全性保证了矿工只能在进行正规计算的前提下提交合法证明,并且只能提交正确的状态更新计算。此外,矿工不再需要将交易数据中的可验证性部分压缩提交到Rollup智能合约中,只需要提交新状态和合法证明即可。由于没有挑战期的限制,链上用户可以随时提取Layer 2中的资金,并且不需要像在Optimistic中一样时刻跟进主链上的验证工作,因为Rollup智能合约会将所有的合法证明放在链上,杜绝了可能存在的矿工作弊行为。Layer 2上的矿工通常由具备较强算力的节点担任,也正因为如此,它们能够将先前复杂耗时的链上打包验证工作转移到链下进行,完成正确计算和打包的矿工也会受到相应的奖励。

具体而言,设想A要将自己链上v个ETH(以太坊

主链的加密货币单位)余额中的 v_0 个 ETH 存入 Layer 2,并将其中的 u 个 ETH 转移给 B,那么 ZK Rollup 的执行会涉及到以下 4 个步骤:

(1)A 将 v_0 个 ETH 转移到 Rollup 合约账户,Rollup 合约相应记录这一转移交易并提交给矿工,此时这 v_0 个 ETH 已经放在了 Layer 2 上。相反,如果有某个用户要从 Layer 2 中取出一定的货币,则同样由 Rollup 合约记录这一转移请求并提交给矿工,最终由 Rollup 合约账户转移相同的货币给该用户。

(2)A 在 Layer 2 上提交一个向 B 转移 u 个 ETH 的交易,矿工接收到该交易,被其连同其它交易(转账、存入、转出)打包起来计算状态更新和证明;

(3)矿工将所有数据(交易更新的状态和合法证明)通过 Rollup 合约广播到 Layer 1 上的验证节点;

(4)验证节点通过 Rollup 合约验证矿工提供的信息,并随后更新链上的状态。

由于 zk-SNARK 的高效简洁,证明长度不会随着 Layer 2 上打包的交易数量而增加,这使得 Layer 1 上的矿工在检查证明时不会花费额外时间,且单次证明的效率本来就相较于直接检查更高,这就大大加快了 Layer 2 处理交易的速度。

需要提到的是,ZK Rollup 的零知识证明方案也会依照具体使用方案而定。例如,Loopring 采用的是非通用可更新的(针对应用专门化的)SNARK 方案(Groth16),zkSync1.0 采用的是通用可更新的(或通用应用的)SNORK 方案(Sonic, Plonk),未来将在 zkSync2.0 采用无需可信设置的 STARK 方案(zk-STARK)。这些零知识工程化方案的性能决定了 ZK Rollup 生成的合法证明的性质。

ZK Rollup 借助 zk-SNARK 的高效性,极大地缩短了 Layer 1 上的验证时间,使区块链的吞吐量得到增加,链上存储负担变小,验证速度加快,区块链的可扩展性大幅提升。

2.2.3　Filecoin——去中心化存储

随着中心化网络向去中心化网络的过渡,中心化可信计算被去中心化可验证计算所替代,数据存储模式也由中心化存储服务器过渡到去中心化存储模式。早期公链无法处理海量数据,只能处理存储压力相对较小的交易信息,如比特币这样的电子账本。而越来越复杂的架构所带来的存储成本也随之增加,如果区块链不能提供优化的去中心化存储方案,将来就不可能应对更为复杂的数据类型。此外,链上留存大量的历史数据都需要用于去中心化节点的验证计算,但很难要求每个验证节点将不断扩张的历史数据全数存储下来,最终也只有存储能力很强的少数节点(如原有的中心化存储服务器)能够胜任,这将威胁到去中心化的本质。2014 年,星际文件系统(InterPlanetary File System, IPFS)诞生,它是一个全球范围的分布式互联网底层协议,通过点对点的文件和应用的分配、传输和共享,极大地提升了互联网数据传输的效率①。IPFS 将替代已在互联网时代产生了巨大影响的 HTTP 协议,它解决了 HTTP 中心化服务器带来的效率低、使用成本高和带宽浪费严重等若干问题。随着 IPFS 协议的推动,区块链存储能力的提升需求也愈发地迫切。

Filecoin 项目是作为 IPFS 的激励层出现的[28-29],负责货币激励、数据存储和检索服务、调整存储市场和资源分配。与传统的电子账本式区块链不同,在 Filecoin 中除了验证节点以外,还有用于提供存储服务的节点和负责信息检索的矿工节点。存储节点与用户在链上签订存储合同,在一定时间内存储用户指定的数据,存储合同作为数据被纳入到区块中,而检索节点则负责从整个存储系统的数据集中检索用户所需的信息,以更好地促进信息传输和降低延迟。这 2 类节点通过在存储市场和检索市场提供的相应服务而获得代币奖励。

一个网络存储服务协议将要求存储节点按照用户订单中所指定的存储时限,在一定的存储空间中固定地存储一定的文件。中心化的存储方案(云存储)中,用户基于中心化服务器的信任完成整个存储和取回的流程,但在去中心化网络中,由于存储空间与记账权和存储收益直接挂钩,恶意节点可能利用去中心化的特点进行一系列攻击(甚至是在要求存储节点提供存储证明时,也存在潜在的攻击):

- 女巫攻击:恶意节点创建多个身份,用户提供文件的若干副本,被恶意节点诱导到多个身份宣称的独立存储空间中,但实际上恶意节点只存储了其中一份,在用户取回时将同一份文件复制多份再发回用户。恶意节点利用女巫攻击获得更多的报酬(因为实际存储空间远小于订单存储空间)。女巫攻击的实质是利用文件与存储空间缺乏实质对应性,导致文件与实际存储空间的不对等。

① Benet J. IPFS-content addressed, versioned, P2P file system. arXiv preprint, 2014: arXiv:1407.3561[cs. NI].

- 外包攻击:恶意节点承诺比实际存储容纳量更多的数据,通过中心化的数据库协助存储自身无法容纳的多余数据(因为中心化的存储处理更快,数据更集中,服务更便宜),这相当于将这部分的数据外包出去,而节点获得与自己实际存储量不符合的收益。外包攻击的实质是没有将宣称的存储文件存放在应有的存储空间中,文件在该过程中可能产生时空上的流动。
- 生成攻击:恶意节点并未实际存储数据,当其被要求提供存储数据的证明时,恶意节点以某种方式快速计算生成证明,从而在未被挑战期间数据仍可以不存储于相应的空间中。生成攻击的实质是文件时空流动的成本和时间过低,使得存储证明被要求时文件能够快速被转移到应当存储的地方。

Filecoin 为了应对这样的攻击,要求存储节点必须提供相应的存储证明才能让用户相信存储服务是正确进行的。这包括 2 个方面:①存储节点确实接收到了用户指定的完整文件,且与对应的存储空间是匹配的;②存储节点在服务运行期间的任何时刻都将同一文件完整地留存在自己封存的存储空间中。为了证明以上 2 点,存储节点要提供 2 个非交互式的零知识证明——复制证明(Proof of Replication, PoRep)和时空证明(Proof of Spacetime, PoSt),这 2 个证明使得被存储文件与特定分配的空间和确定的一段时间绑定在一起。由于复制证明的存在,任何被存储的文件副本必须分别存储在不同的空间中,因此,女巫攻击是不可行的。另外,由于时空证明的存在,节点无法在规定时间之内将数据的一部分挪出存储区,也不能在给定的空间中存储与给定数据不同的其它数据,因此,外包攻击和生成攻击也是不可行的。

本文首先引入一个可验证延迟函数(Verifiable Delay Function, VDF)的概念,这个概念是由 Boneh 等[30]提出的。VDF 除了具有计算可验证性之外,最显著的特点是它不是一个快速算法,即当安全参数为 k 时,VDF 的计算过程至少是 k 的次指数时间级别的,这意味着即使 VDF 的算法是公开的,其输出也具有明显的时延性质。

复制证明的第一步要求存储节点对数据进行封存操作。为了保证数据的每个副本被储存在独立的存储空间,每个副本在存储到某个特定空间时,需要保证都要通过一个特定的变换打乱副本,这样的操作称为封存,如果需要存储同一个文件的若干副本,则这些副本在封存之后也是不同的,因此,没法直接复制。Filecoin 设计了一个 VDF 作为封存算法,它通过存储节点的私钥将数据 D 在一定的时延后封存为 R。一方面,存储节点要计算 D 的 Hash 值和封存数据 R(视作长数组)的 Hash Merkle Root 这 2 个公开信息以供验证;而由于存储节点具有私钥和 D 作为秘密的简短证据,于是该节点可以计算一个零知识证明 π_{SEAL},来证明自己封存阶段的正确性;另一方面,封存的数据 R 只有固定存放在存储空间时才能被读取,因而存储阶段还要在随机挑战下找到 R 中被随机挑战的叶子节点在 Hash Merkle Tree 中的一条路径,这进一步要求该节点提供另一个零知识证明 π_{POS},这就证明了存储节点确实固定封存了相应的文件。(π_{SEAL},π_{POS})构成了一个完整的复制证明。

时空证明是在订单执行期间(设时间为 T)的各个时间节点中不断产生的,相当于不断地根据挑战者的随机挑战链提供一套复制证明,存储节点必须保证每个副本的封存信息在 T 时间内始终被存储在先前指定的特定空间中,且全过程封存信息都是完整的。如果存储节点没有在被挑战的时间点保存完整的封存数据,则为了通过证明,他必须将完整数据 D 重新封存回原有的空间以支持复制证明的计算,但鉴于封存操作的延时性,恶意节点无法以很快的速度生成这样的证明,这就保证了封存信息在指定存储时间和空间中的完整性。时空证明意味着到任意的被挑战时间 $t\in[0,T]$ 为止,存储节点一直将封存数据存储在对应的空间中。

Filecoin 通过可验证延迟函数实现了复制证明和时空证明,解决了中心化存储向去中心化存储过渡的一系列潜在安全问题,提供了一个去中心化存储方案。有了去中心化存储,区块链的存储能力能够得到大幅提升,也具备了应对未来去中心化需要面临的海量数据处理挑战的能力。可验证延迟函数本身也为未来公链融合更多技术提供了重要的工具。

2.2.4 Mina——递归证明

传统公链中验证时间长、验证成本巨大、链上存储的信息积累严重的问题一直拖累着区块链的可扩展性。先前提到的 2 个公链中,ZK Rollup 尝试将验证转移到链下进行,将验证时间和成本转移到 Layer 2 上具有较强计算能力的矿工;Filecoin 则是通过去中心化的存储方案,使得原先链上存储的积累验证信息转移到存储节点的空间中,但是 2 者都没有解决验证信息占用空间与实际效用之比过大的问题。为了确保区块链的安全性,此前的公链必须在链上存有足够多的验证信息,以供每个节点进行重复验证,而这些验证信息所占用的空间过于庞大,反而给区块链的可扩展性带来麻烦;又由于验证工作过于消耗时间和计算资源,新加入的节点没有足

够的动力去运营全节点,网络中全节点的比例下降将进一步影响到去中心化和安全性。

2017 年诞生的 Mina 给出了一个全新的去中心化支付系统的设计,实现了第一个超轻量级的区块链[31]。所谓 Mina 的超轻量级,其含义是任何 Mina 上的节点仅通过一个大小为 22 KB 的零知识证明,就可以验证从创始区块以来整条区块链上的所有区块的合法性,使得链上验证所需的时间和资源成本压缩到了极致。Mina 的网络中包含 3 类节点,分别是区块生产者、证明生产者和存档节点,区块生产者是在链上产生区块的矿工节点,证明生产者则利用 zk-SNARK 对区块进行压缩并生成证明,而存档节点要负责链下存储所有历史上的详细区块数据,是 Mina 的持久性可检索数据源。Mina 大致的运行流程如下:

(1)由 Ouroboros Samasika 共识协议[32](Mina 采用的 PoS 共识协议)确定的区块生产者负责打包交易并生成共识区块;

(2)证明生产者依照区块信息更新链上的状态,同时产生更新的零知识证明 snark ;

(3)第 i 个区块链快照 $S_i=(\sigma_i,\text{snark}_i)$,由区块 B_i 生成后的系统状态 σ_i 和对应的状态转移零知识证明 snark_i 构成,任何节点可以验证新生成的区块 B_{i+1} 是否与 S_i 匹配(检查区块交易生效前的系统状态是否满足 S_i 所指示的状态匹配)。如果 2 者匹配,则节点可以依照 σ_i 和 B_{i+1} 更新系统状态为 σ_{i+1},而 σ_{i+1} 和对应区块 B_{i+1} 的零知识证明 snark_{i+1} 组成新的区块链快照 S_{i+1}。

从创世区块 B_0 开始,区块链的初始状态 σ_0 和一个缺省的 snark_0 产生,每当有新的区块 B_{i+1} 生成时,系统状态将从 σ_i 更新为 σ_{i+1},同时整个合法更新过程的信息连同上一个证明都被压缩到一个简洁的零知识证明 snark_{i+1} 中,继而新的区块链快照产生,它包括 2 个方面的信息:①σ_{i+1} 是 B_{i+1} 产生后整个系统的新状态信息;②snark_{i+1} 则表示从创世区块 B_0 以来的所有区块链状态更新都是合法的。区块链快照是这一过程中的一个个切片信息,如果把区块链延长比作是一个摄影师对着一幅画从近到远的拍摄过程,那么拍摄过程中每个瞬间的照片都指示了当前区块链的系统状态,而零知识证明则保证了摄影师拉远过程的连续性:摄影师始终对着同一幅图拍照,不同状态是依照同一幅图从近到远的顺序拍照得到的;照片的尺寸永远是固定的,先前拍摄的照片在之后的照片中占据了一部分图像,但是将这部分图像放大后就可以得到先前的照片。

Mina 之所以能够做到让 snark_i 蕴含先前所有状态更新合法性的事实,是借助了被称为递增可计算证明(Incrementally Computable SNARKs)这一由 Valiant[33] 提出的递归零知识证明技术。递归零知识证明可以实现对递归过程中所有证明的有效性证明。在区块链中实施递归零知识证明不仅能够保证当前的共识区块的合法性,同时也能保证此前所有历史已共识区块的合法性。形式化地说:

(1)假设 σ_i 和 σ_{i+1} 是 2 个先后状态,则 σ_{i+1} 作为 σ_i 的一个可能的后继状态,可以由状态转换过程 τ_i 作为简短证据,于是存在一个 zk-SNARK π_i 可以证明:存在某个状态转换过程 τ_i 使得 σ_{i+1} 是 σ_i 在 τ_i 作用下的后继状态;

(2)假设 σ_i 和 σ_{i+1} 是 2 个先后状态,则如(1)所述,可以由 π_i 作为简短证据,于是存在一个 zk-SNARK π_i' 可以证明:存在某个 zk-SNARK π_i 使得 π_i 是“σ_{i+1} 是 σ_i 的后继状态”的证明;

(3)假设 σ_i 和 σ_{i+2} 是 2 个相差阶段为 2 的状态,将 π_i' 和 π_{i+1}' 作为简短证据,于是存在一个 zk-SNARK $\pi_{(i)}$ 可以证明:存在 2 个先后的状态转换过程 τ_i 和 τ_{i+1},使得 σ_{i+2} 是 σ_i 在 2 个阶段后的状态。

从(1)~(3)的过程使得零知识证明完成了一次递归,从而一次状态转化的正确性证明被延长了 2 次。随着区块的不断延长,递增可计算证明可以通过先前区块的证明递归生成,于是最终只需验证一个 zk-SNARK,即可证明之前所有区块状态转化的正确性,不再需要每个节点进行重复而复杂的验证工作。

在区块链中,将区块视作系统状态的转换过程,于是递增可计算证明可以通过先前区块的证明递归生成,同时由于 zk-SNARK 可以给出非常简短的证明,留存在链上的存储数据可以非常地小。递归零知识证明的优势可以推广到更多的应用场景中。Mina 搭载了通用可支持的 SNARK 智能合约——Snapps,这使得 Mina 可以整合许多互联网中的数据库,并只需要执行一次初始化的业务逻辑,就可以使得互联网服务在 Mina 上进行,且不再需要每个节点进行重复而复杂的验证工作。传统的中心化机构的公信力使得人们不必经过无意义的重复验证就能够相信数据更新的正确性,现在 Mina 将这种信任放在了去中心化网络中,并且相对于中心化而言,依赖零知识证明的安全性也更加可靠。

Mina 通过递归零知识证明技术,将区块链的验证效率和存储效率提升到了一个新的高度,真正意义上将传统基于公信力的信任转化为基于对算法和密码学的信任。Mina 在提升了区块链可扩展性的同时,也为区块链

和实体经济、网络服务等落地化的应用场景提供了更多的结合空间。

2.2.5 Aztec——Layer 2 的隐私实现

zk-SNARK 虽然已经在 Zcash 上实现了隐私性,但是在本来已经低效的 PoW 公链中加上隐私属性,会使其运行效率进一步下降。此外,尽管以太坊在 Layer 2 扩容的加持下能够具备较为理想的扩展性,但是其隐私性问题一直没有得到解决。尤其是当智能合约进入公链后,公链的功能性得到了极大的拓展,这对公链的可扩展性带来了更大的挑战。

2018 年,创造 Plonk 协议的研究团队 Aztec 提出了 Aztec Protocol[34],旨在在以太坊上建立一个 Layer 2 上的高速隐私网络。其大致想法是,对于以太坊上发起的转账交易和合约交易,构造关于交易数额、交易双方信息、运行代码和被交易资产等信息的隐私保护机制,在之前 ZK Rollup 的基础上建立隐私层,搭建 Plonk 证明的 ZK-ZK Rollup,从而缩减主网上隐私交易的 gas 费①,提升吞吐量并保证隐私性。之前的 ZK Rollup 虽然冠以 zero-knowledge 的词头,但是本质上 ZK Rollup 并不提供隐私保护,而是借助 zk-SNARK 的证明简洁性使得 Layer 1 上各个节点的验证效率提高,因此,真正添加了零知识性,使得 Layer 2 具有隐私性的 Rollup 合约被称作 ZK-ZK Rollup。

在 Aztec 的隐私层架构中,相对于以太坊的基于账户的记账模型,Aztec 采取了更利于构建隐私性的 UTXO 模型。UTXO 模型的本质是货币的归属权的转移,当构建隐私层时,只需考虑被加密的货币在交易中发生的单向归属权转移的过程,因而只需将货币原持有者的匿名信息替换为新持有者的匿名信息。然而在基于账户的记账模型下,随着交易的发生,双方的账户余额都有直接的改变,在验证相应的隐私交易时,需要牵涉到双方的计算过程。Aztec Protocol 将 Layer 2 上产生的交易通过 Aztec 承诺函数进行加密,称为 Aztec Note。一个Aztec Note 由一组椭圆曲线承诺值、浏览密钥、支出密钥和加密交易信息构成,浏览密钥的持有者能够解密 Note 并浏览其中的内容,并且要负责创建交易的 Join-Split 零知识证明,而支出密钥的持有者能够批准交易的进行,即提供一个由支出密钥签署的 Join-Split 证明的签名。

Join-Split 零知识证明是 Aztec 中隐私交易的关键。交易发起方在一笔交易中将提供自身持有的若干 Note,并将其中的一部分 Note 的所属权转移给交易接收方,这部分 Note 中包含的金额就是交易额,而剩余的 Note 仍归属自己。于是 Join-Split 要求证明交易前的若干 Note 包含的金额与分配给双方后的若干 Note 包含的金额相等。类似于 Zcash 的处理,Note 并不是直接转移给对方,而是使得消费过的 Note 变得无效,同时新产生若干相当价值的 Note。交易发起方需要用自己 Note 中的浏览密钥创建 Join-Split 零知识证明,以证明自己在交易中所创建的所有新 Note 与先前的 Note 等值,此外,交易发起方还将证明自己拥有对先前 Note 的所属权。Aztec 特别维护了 2 个数据结构(Hash Merkle Tree):Note 树和 Nullifier 树,其中,Note 树包含了所有在系统历史上创建过的 Note,而一旦 Note 经过交易被无效化,则会被 Nullifier 树记录下来。当交易发起方需要证明自己拥有先前的 Note 时,需要提供另一个零知识证明来说明几个条件:①这个 Note 存在于 Note 树中;②在交易前该 Note 并不记录在 Nullifier 树中;③在获得该 Note 的交易中,Note 的现持有者具有支出密钥,因而这个 Note 在当时的交易中将转移给现持有者。

为了 ZK-ZK Rollup 的高效执行,Aztec 还将多个 zk-SNARK 的逻辑编译成若干电路。除了计算隐私 Note 和执行交易的 Join-Split 证明电路和隐藏用户身份的 Account 电路之外,还有负责处理隐私用户从 Layer 2 中提取货币的 Escape-Hatch 电路,以及继承了 ZK Rollup 的 Rollup 电路,它能够将所有 Join-Split 证明聚合为单个证明放在链上。而 Root Rollup 电路能够将 Rollup 电路进行进一步的聚合,这极大地提高了 Layer 1 上的验证效率。此外,Aztec Protocol 后续还将用于零知识证明的 Plonk 改进为 TurboPlonk[35]。TurboPlonk 增加了一些自定义门(如逻辑门、范围检查和 Pedersen Hash 等),解决了 Plonk 因只有算术电路门带来的效率低下问题。此外,高效的范围检查使得递归证明在以太坊上的实现更简单。特别地,TurboPlonk 更是引入了最新的椭圆曲线多标度乘法算法,极大地提升了零知识证明的效率。

① 在以太坊网络中,合约执行需要依赖算力的推动,合约代码的具体执行将以算力的方式换算为计算产生的 gas 费,支付时以 ETH 进行换算。通过对网络上每一笔计算的执行收费,以太坊能够保证系统免于遭受循环攻击的风险,同时也能避免过大的代码或循环代码造成不必要的算力损耗。每笔合约交易中将指定一个最大的 gas 费以指示运行该合约代码所需要支付的算力费用,一旦实际计算产生的 gas 费超过最大 gas 费则计算终止,如果计算结束后产生的 gas 费没有超过最大值,则会在扣除小费后返还剩余的 ETH。

Aztec 从零知识证明算法起家,不断改善高效的零知识证明算法,最终实现了 Layer 2 上的高效隐私网络。Aztec Protocol 不仅大幅缩减了区块链的验证成本,同时还解决了 Layer 2 上困难的隐私问题,是兼具了隐私保护和可扩展性的方案。它将零知识证明在区块链中发挥的作用又提升到了一个新的高度,为扩大区块链网络业务,创造更完善的区块链生态提供了技术支持。

3　总　结

本文从多个方面探讨了零知识证明技术在改善区块链的隐私性、可扩展性、存储能力和验证能力等方面上发挥的巨大效用:①Zcash 借助零知识证明首次实现了隐私交易;②ZK Rollup 利用 zk-SNARK 的简洁性极大地提升了区块链的可扩展性;③Filecoin 通过 2 个关键的零知识证明实现了去中心化存储,同时解决了区块链验证数据累积的问题;④Mina 将整个区块链的验证极小化地浓缩在了简短的零知识证明中,为区块链的可验证性带来了前所未有的效率;⑤Aztec Protocol 将各类零知识证明作为实现兼具隐私性和可扩展性的网络层的重要工具,使得区块链具有越来越多的良好性质。

可以清楚地认识到零知识证明已经实现了,或将能够实现越来越多的功能,包括但不局限于秘密计算、隐藏种子、递归证明、创造黑箱、秘密投票、身份识别、范围证明、延迟证明和秘密共享,等等。**零知识证明本质上起到了制造隐私和信息差的作用,同时能够保持证明信息的简短,这将使得区块链能够在需要抹除不平等信息差的时候提供透明性和不可篡改性,在需要保护隐私的时候利用零知识证明建立起足够的信息差,在需要提供可验证性时能保持足够的高效性。**零知识证明在 30 多年的发展中已经形成了一个较成熟的体系,**在区块链系统设计中,将系统需要改善的问题或需要实现的特性转化为某个特定的可证明问题,再由证明问题转向工程化算法实现的途径,最后将零知识证明转化为区块链可实现的某个性质或功能,**每一步都需要综合考虑零知识证明是否能够做到在不牺牲去中心化、安全性和可扩展性的基础上仍然能提供这些特性。特别地,区块链作为一个新兴的网络生态底层架构,其所需的广泛的隐私安全、互通互联的高运算能力和全局可验证、可监管的基础保障手段都需要隐私计算,尤其是零知识证明的参与,这也是零知识证明在未来区块链发展中的趋势和优势。若干已有的零知识证明在区块链中应用的案例已经表明,只要将零知识证明发挥恰当,就能够极大地改善区块链在应用场景中的性能,这种新兴的互联网技术也就能体现出越来越重要的价值。

致谢　北京大数据研究院区块链与隐私计算中心的莫晓康教授对本文写作提供了宝贵的指导意见,在此表示感谢!

参考文献:

[1] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof-systems[C]//STOC '85: Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing. New York: ACM, 1985: 291-304.

[2] Furer M, Goldreich O, Mansour Y, et al. On completeness and soundness in interactive proof systems[J]. Advances in Computing Research: A Research Annual, 1989,5: 429-442.

[3] Ben-Or M, Goldreich O, Goldwasser S, et al. Everything provable is provable in zero-knowledge[C]//Advances in Cryptology-CRYPTO '88. Berlin: Springer, 1988: 37-56.

[4] Shamir A. IP = PSPACE[J]. Journal of the ACM, 1992,39(4): 869-877.

[5] Blum M, Feldman P, Micali S. Non-interactive zero-knowledge and its applications[C]//STOC '88: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing. New York: ACM, 1988: 103-112.

[6] Blum M, De Santis A, Micali S, et al. Noninteractive zero-knowledge[J]. SIAM Journal on Computing, 1991,20(6): 1084-1118.

[7] Fiat A, Shamir A. How to prove yourself: Practical solutions to identification and signature problems[C]//Advances in Cryptology-CRYPTO '86. Berlin: Springer, 1987: 186-194.

[8] Goldwasser S, Tauman Y. On the (in)security of the Fiat-Shamir paradigm[C]//FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE, 2003: 102-115.

[9] Fortnow L. The complexity of perfect zero-knowledge[C]//STOC '87: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing. New York: ACM, 1987: 204-209.

[10] Groth J. Short pairing-based non-interactive zero-knowledge arguments[C]//Advances in Cryptology-ASIACRYPT '10. Berlin: Springer, 2010: 321-340.

[11] Groth J. On the size of pairing-based non-interactive arguments[C]//Advances in Cryptology-EUROCRYPT '16. Berlin:

Springer, 2016: 305-326.
[12] Bitansky N, Canetti R, Chiesa A, et al. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again[C]//ITCS '12: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. New York: ACM, 2012: 326-349.
[13] Parno B, Howell J, Gentry C, et al. Pinocchio: Nearly practical verifiable computation[C]//2013 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2013: 238-252.
[14] Groth J, Kohlweiss M, Maller M, et al. Updatable and universal common reference strings with applications to zk-SNARKs [C]// Advances in Cryptology-CRYPTO '18. Berlin: Springer, 2018: 698-728.
[15] Maller M, Bowe S, Kohlweiss M, et al. Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings[C]//CCS '19: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2019: 2111-2128.
[16] The halo2 book[EB/OL]. [2022-10-01]. https://zcash. github. io/halo2/.
[17] Ben-Sasson E, Bentov I, Horesh Y, et al. Scalable zero knowledge with no trusted setup[C]// Advances in Cryptology-CRYPTO '19. Berlin: Springer, 2019: 701-732.
[18] Bünz B, Bootle J, Boneh D, et al. Bulletproofs: short proofs for confidential transactions and more[C]//2018 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2018: 315-334.
[19] 朱旭东,邓燚. 零知识证明在区块链中的应用[R]. 中国密码学发展报告2021,2021:181-195.
[20] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL]. (2008-10-31)[2022-10-01]. https://bitcoin. org/bitcoin. pdf.
[21] Ben-Sasson E, Chiesa A, Garmanz C, et al. Zerocash: Decentralized anonymous payments from Bitcoin[C]//2014 IEEE Symposium on Security and Privacy. Piscataway: IEEE, 2014: 459-474.
[22] The ZILLIQA Team. The ZILLIQA technical whitepaper (version 0.1)[EB/OL]. (2017-08-10)[2022-10-01]. https://docs. zilliqa. com/whitepaper. pdf.
[23] Kendzicky J. Raiden Network: Ethereum network payment channels[EB/OL]. (2018-05-28)[2022-10-01]. https://messari. github. io/research/profiles/RaidenNetwork(RDN). pdf.
[24] Arasev V. POA network whitepaper[EB/OL]. (2018-09-28)[2022-10-01]. https://github. com/poanetwork/wiki/wiki/POA-Network-Whitepaper.
[25] Poon J, Buterin V. Plasma: Scalable autonomous smart contracts[EB/OL]. (2017-08-11)[2022-10-01]. https://www. plasma. io/plasma. pdf.
[26] Wang D, Zhou J, Wang A. Loopring: A decentralized token exchange protocol[EB/OL]. (2018-09-08)[2022-10-01]. https://loopring. org/resources/en_whitepaper. pdf.
[27] Matter Labs. zkSync rollup protocol[EB/OL]. (2021-10-07)[2022-10-01]. https://github. com/matter-labs/zksync/blob/f59e154865374bdc0f5 ded2e2604dac599cb75ee/docs/protocol. md.
[28] Protocol Labs. Filecoin: A cryptocurrency operated file storage network[EB/OL]. (2014-07-15)[2022-10-01]. https://filecoin. io/filecoin-jul-2014. pdf.
[29] Protocol Labs. Filecoin: A decentralized storage network[EB/OL]. (2017-07-19)[2022-10-01]. https://filecoin. io/filecoin. pdf.
[30] Boneh D, Bonneau J, Bünz B, et al. Verifiable delay functions[C]// Advances in Cryptology-CRYPTO '18, Berlin: Springer, 2018: 757-788.
[31] Bonneau J, Meckler I, Rao V, et al. Mina: Decentralized cryptocurrency at scale[EB/OL]. (2020-10-17)[2022-10-01]. https://docs. minaprotocol. com/static/pdf/technical\%20Whitepaper. pdf.
[32] David B, Gaži P, Kiayias A, et al. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake protocol[C]// Advances in Cryptology-EUROCRYPT'18(2). Berlin: Springer, 2018:66-98.
[33] Valiant P. Incrementally verifiable computation or proofs of knowledge imply time/space efficiency[C]//TCC '08: Theory of Cryptography. Berlin: Springer, 2008: 1-18.
[34] Williamson Z. The AZTEC protocol (version 1.0.1)[EB/OL]. (2018-12-04)[2022-10-01]. https://github. com/AztecProtocol/AZTEC/blob/master/AZTEC. pdf.
[35] Gabizon A, Williamson Z. Proposal: The Turbo-PLONK program syntax for specifying SNARK programs[EB/OL]. (2019-12-18)[2022-10-01]. https://docs. zkproof. org/pages/standards/accepted-workshop3/proposal-turboplonk. pdf.

【责任编辑:陈 钢】